Survey report

Independent component analysis

Doan Ngoc Thinh, Phan Minh Vuong
Doan Thi Thao Quynh, Le Thi Thuy Trang

May 22rd 2020

1 Introduction

A central problem in neutral network research, as well as in statistics and signal processing, is finding a suitable representation of the data. It is crucial for subsequent analysis of the data, whether it be pattern recognition, data compression, de-nosing, visualization or anything else, that the data is represented in a manner that facilitates the analysis.

As a trivial example, 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 denote by $x_1(t)$ and $x_2(t)$, with x_1 and x_2 the amplitudes, and t the time index. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, which we denote by $s_1(t)$ and $s_2(t)$. We could express this as a linear equation:

$$x_1(t) = a_{11}s_1 + a_{12}s_2 \tag{1}$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2 \tag{2}$$

where $a_{11}, a_{12}, a_{21}, a_{22}$ are some parameters that depend on the distances of the microphones from the speakers. It would be very useful if you could now estimate the two original speech signals $s_1(t)$ and $s_2(t)$, using only the recorded signals $x_1(t)$ and $x_2(t)$. This is called the cocktail-party problem. For the time being, we omit any time delays or other extra factors from our simplified mixing model.

Actually, if we knew the parameters a_{ij}, we could solve the linear equation in (1) by classical methods. The point is, however, that if you don't know the a_{ij}, the problem is considerably more difficult.

One approach to solving this problem would be to use some information on the statistical properties of the signals $s_i(t)$ to estimate the a_{ii}. Actually, and

perhaps surprisingly, it turns out that it is enough to assume that $s_1(t)$ and $s_2(t)$, at each time instant t, are statistically independent. This is not an unrealistic assumption in many cases,and it need not be exactly true in practice. The recently developed technique of Independent Component Analysis, or ICA, can be used to estimate the a_{ij} based on the information of their independence, which allows us to separate the two original source signals $s_1(t)$ and $s_2(t)$ from their mixtures $x_1(t)$ and $x_2(t)$.

Independent component analysis was originally developed to deal with problems that are closely related to the cocktail-party problem. Since the recent increase of interest in ICA, it has become clear that this principle has a lot of other interesting applications as well.

It would be most useful to estimate the linear transformation from the data itself, in which case the transform could be ideally adapted to the kind of data that is being processed.Feature extraction by ICA will be explained in more detail later on.

2 Independent Component Analysis

2.1 Definition of ICA

Assume that we observe n linear mixtures x_1,...,x_n of n independent components

$$x_j = a_{j1}s_1 + a_{j2}s_2 + ... + a_{jn}s_n, \text{ for all j} \tag{3}$$

We have now dropped the time index t; in the ICA model, we assume that each mixture x_j as well as each independent component s_k is a random variable, instead of a proper time signal. The observed values $x_j(t)$, e.g., the microphone signals in the cocktail party problem, are then a sample of this random variable. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean: If this is not true, then the observable variables x_i can always be centered by subtracting the sample mean, which makes the model zero-mean.

It is convenient to use vector-matrix notation instead of the sums like in the previous equation. Let us denote by x the random vector whose elements are the mixtures $x_1, ..., x_n$, and likewise by s the random vector with elements $s_1, ..., s_n$. Let us denote by A the matrix with elements a_{ij}. Generally, bold lower case letters indicate vectors and bold upper-case letters denote matrices. All vectors are understood as column vectors; thus x^T, or the transpose of x, is a row vector. Using this vector-matrix notation, the above mixing model is written as

$$\boldsymbol{x = As} \tag{4}$$

Sometimes we need the columns of matrix A; denoting them by a_j the model can also be written as

$$\boldsymbol{x} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{a_i s_i} \tag{5}$$

The statistical model in Eq. 4 is called independent component analysis, or ICA model. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_i. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector x, and we must estimate both A and s using it. This must be done under as general assumptions as possible.

The starting point for ICA is the very simple assumption that the components s_i are statistically independent. It will be seen below that we must also assume that the independent component must have nongaussian distributions. However, in the basic model we do not assume these distributions known (if they are known, the problem is considerably simplified.) For simplicity, we are also assuming that the unknown mixing matrix is square. Then, after estimating the matrix A, we can compute its inverse, say W, and obtain the independent component simply by:

$$\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x} \tag{6}$$

In many applications, it would be more realistic to add a noise term in the model. For simplicity, we omit any noise terms, since the estimation of the noise-free model is difficult enough in itself, and seems to be sufficient for many applications.

2.2 Ambiguities of ICA

Noted that both **s** and **A** being unknown, we can neither determine the variances nor the order of the independence components. Firstly, to determine the variances of those, we assumes that each has unit variance: E(s_i^2)= 1. Then the matrix **A** will be adapted in the ICA solution methods to take into account this restriction. Note that this still leaves the ambiguity of the sign: we could multiply the an independent component by 1 without affecting the model. This ambiguity is, fortunately, insignificant in most applications. Secondly, we can freely change the order of the terms in the sum and call any of the independent components the first one. Formally, a permutation matrix **P** and its inverse can be substituted in the model to give $x = AP^{-1}Ps$ .The elements of **Ps** are the original independent variables s_j, but in another order. The matrix $x = AP^{-1}$ is just a new unknown mixing matrix, to be solved by the ICA algorithms.

3 Statistical independence

3.1 Definition and fundamental properties

Let us denote by $p(y_1, y_2)$ the joint probability density function (pdf) of y_1 and y_2. Let us further denote by $p_1(y_1)$ the marginal pdf of y_1, i.e. the pdf of y_1 when it is considered alone:

$$p_1(y_1) = \int p(y_1, y_2)\, dy_2 \tag{7}$$

and similarly for y_2. Then we define that y_1 and y_2 are independent if and only if the joint pdf is factorizable in the following way:

$$p(y_1, y_2) = p_1(y_1)\, p_2(y_2) \tag{8}$$

This definition extends naturally for any number n of random variables, in which case the joint density must be a product of n terms.
The definition can be used to derive a most important property of independent random variables. Given two functions, h_1 and h_2, we always have

$$E\{h_1(y_1)\, h_2(y_2)\} = E\{h_1(y_1)\}\, E\{h_2(y_2)\} \tag{9}$$

3.2 Uncorrelated variables are only partly independent

A weaker form of independence is uncorrelatedness. Two random variables y_1 and y_2 are said to be uncorrelated, if their covariance is zero:

$$E\{y_1 y_2\} - E\{y_1\}\, E\{y_2\} = 0 \tag{10}$$

On the other hand, uncorrelatedness does not imply independence. Many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem.

3.3 Why Gaussian variables are forbidden

The fundamental restriction in ICA is that the independent components must be nongaussian for ICA to be possible. To see why gaussian variables make ICA impossible, assume that the mixing matrix is orthogonal and the s_i are gaussian. Then x_1 and x_2 are gaussian, uncorrelated, and of unit variance. Their joint density is given by

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{x_1^2 + x_2^2}{2}\right) \tag{11}$$

The density is completely symmetric. Therefore, it does not contain any information on the directions of the columns of the mixing matrix A. This is why A cannot be estimated.

More rigorously, one can prove that the distribution of any orthogonal transformation of the gaussian (x_1, x_2) has exactly the same distribution as (x_1, x_2), and that x_1 and x_2 are independent. Thus, in the case of gaussian variables, we can only estimate the ICA model up to an orthogonal transformation. In other words, the matrix $\mathbf{A}$ is not identifiable for gaussian independent components. (Actually, if just one of the independent components is gaussian, the ICA model can still be estimated.)

4 Principles of ICA estimation

In ICA, the goal is to find the unmixing matrix (W) and then projecting the whitened data onto the matrix for extracting independent signals. This matrix can be estimated using three main approaches of independence, which results in slightly different unmixing matrices. The first is based on the non-Gaussianity. This can be measured by some methods such as kurtosis, and negentropy. In the second approach, the ICA goal can be obtained by minimizing the mutual information. Additionally, ICA can be also estimated by using maximum likelihood estimation (MLE).

All approaches simply search for a rotation on unmixing matrix W. Projecting the whitened data onto the rotation matrix extracts independent signals. The preprocessing steps are calculated from the data, but the rotation matrix is approximated numerically through an optimization procedure. Searching for the optimal solution is difficult due to the local minima exists in the objective function. In this section, different approaches are introduced for extracting independent components.

4.1 "Nongaussian is independent"

The key to estimating the ICA model is nongaussianity. Based on the Central limit theory, a sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables.

We consider a linear combination of the x_i; let us denote this by

$$y = \mathbf{w}^T\mathbf{x} = \sum_i w_i x_i \tag{12}$$

where w is a vector to be determined. If w were one of the rows of the inverse of A, this linear combination would actually equal one of the independent components. In practice, we cannot determine such a w exactly, because we have no knowledge of matrix A, but we can find an estimator that gives a good approximation.

To see how this leads to the basic principle of ICA estimation, let us make a change of variables, defining $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then we have $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s}$. y is thus a linear combination of s_i, with weights given by z_i. Since a sum of even two independent random variables is more gaussian than the original variables,

$\mathbf{z}^T\mathbf{s}$ is more gaussian than any of the s_i and becomes least gaussian when it in fact equals one of the s_i. In this case, obviously only one of the elements z_i of z is nonzero. (Note that the s_i were here assumed to have identical distributions.)

Therefore, we could take as w a vector that maximizes the nongaussianity of $\mathbf{w}^T\mathbf{x}$. Such a vector would necessarily correspond (in the transformed coordinate system) to a z which has only one nonzero component. This means that $\mathbf{w}^T\mathbf{x} = \mathbf{z}^T\mathbf{s}$ equals one of the independent components!.

Maximizing the nongaussianity of $\mathbf{w}^T\mathbf{x}$ thus gives us one of the independent components. In fact, the optimization landscape for nongaussianity in the n-dimensional space of vectors w has 2n local maxima, two for each independent component, corresponding to s_i and $-s_i$ (recall that the independent components can be estimated only up to a multiplicative sign). To find several independent components, we need to find all these local maxima. This is not difficult, because the different independent components are uncorrelated.

4.2 Measures of nongaussianity

To use nongaussianity in ICA estimation, we must have a quantitative measure of nongaussianity of a random variable, say y. To simplify things, let us assume that y is centered (zero-mean) and has variance equal to one. Actually, one of the functions of preprocessing in ICA algorithms, to be covered in Section 5, is to make this simplification possible.

4.2.1 Kurtosis

The classical measure of nongaussianity is kurtosis or the fourth-order cumulant. It indicates how peaky the distribution is. The source signals can be extracted by finding the orientation of the weight vectors which maximize the the kurtosis. The kurtosis of y is classically defined by

$$kurt(y) = E\left\{y^4\right\} - 3\left(E\left\{y^2\right\}\right)^2 \tag{13}$$

Actually, since we assumed that y is of unit variance, the right-hand side simplifies to $E\left\{y^4\right\} - 3$. This shows that kurtosis is simply a normalized version of the fourth moment $E\left\{y^4\right\}$. For a gaussian y, the fourth moment equals $3\left(E\left\{y^2\right\}\right)^2$. Thus, kurtosis is zero for a gaussian random variable. For most (but not quite all) nongaussian random variables, kurtosis is nonzero.

Kurtosis can be both positive or negative. Random variables that have a negative kurtosis are called subgaussian, and those with positive kurtosis 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. A typical example is the Laplace distribution, whose pdf (normalized to unit variance) is given by

$$p(y) = \frac{1}{\sqrt{2}}\exp(\sqrt{2}|y|) \tag{14}$$

Kurtosis, or rather its absolute value, has been widely used as a measure of nongaussianity in ICA and related fields. The main reason is its simplicity, both computational and theoretical. Computationally, kurtosis can be estimated simply by using the fourth moment of the sample data. Theoretical analysis is simplified because of the following linearity property: If x1 and x2 are two independent random variables, it holds

$$kurt\left(x_1 + x_2\right) = kurt\left(x_1\right) + kurt\left(x_2\right) \tag{15}$$

and

$$kurt\left(\alpha x_1\right) = \alpha^4 kurt\left(x_1\right) \tag{16}$$

where α is a scalar. These properties can be easily proven using the definition. Assume that the independent components s1, s2 have kurtosis values kurt(s1), kurt(s2), respectively, both different from zero. Let us again make the transformation $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then we have $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s} = z_1 s_1 + z_2 s_2$. Now, based on the additive property of kurtosis, we have $kurt(y) = kurt\left(z_1 s_1\right) + kurt\left(z_2 s_2\right) = z_1^4 kurt\left(s_1\right) + z_2^4 kurt\left(s_2\right)$. On the other hand, we made the constraint that the variance of y is equal to 1, based on the same assumption concerning s1,s2. This implies a constraint on z: $E\left\{y^2\right\} = z_1^2 + z_2^2 = 1$. Geometrically, this means that vector z is constrained to the unit circle on the 2-dimensional plane. The optimization problem is now: what are the maxima of the function $|kurt(y)| = \left|z_1^4 kurt\left(s_1\right) + z_2^4 kurt\left(s_2\right)\right|$ on the unit circle?

It is not hard to show (Delfosse and Loubaton, 1995)[3] that the maxima are at the points when exactly one of the elements of vector z is zero and the other nonzero; because of the unit circle constraint, the nonzero element must be equal to 1 or -1. But these points are exactly the ones when y equals one of the independent components$\pm s_i$, and the problem has been solved.

In practice, we would start from some weight vector w, compute the direction in which the kurtosis of $y = \mathbf{w}^T\mathbf{x}$ is growing most strongly (if kurtosis is positive) or decreasing most strongly (if kurtosis is negative) based on the available sample x(1),...,x(T) of mixture vector x, and use a gradient method or one of their extensions for finding a new vector w. The example can be generalized to arbitrary dimensions, showing that kurtosis can theoretically be used as an optimization criterion for the ICA problem.

In order to find the correct value of w, we can use gradient descent method. We first of all whiten the data, and transform x into a new mixture z , which has unit variance, and $\mathbf{z} = \left(z_1, z_2, \ldots, z_M\right)^T$. This process can be achieved by applying Singular value decomposition to x, $\mathbf{x} = \mathbf{U}\mathbf{D}\mathbf{V}^T$.

Rescaling each vector $U_i = U_i / \mathrm{E}\left(U_i^2\right)$, and let $\mathbf{z} = \mathbf{U}$. The signal extacted by a weighted vector $\mathbf{w}$ is $\mathbf{y} = \mathbf{w}^T\mathbf{z}$, fthe weight vector whas unit length, that is $\mathrm{E}\left[\left(\mathbf{w}^T\mathbf{z}\right)^2\right] = 1$, then the kuross can be witten as $K = \frac{E\left[y^4\right]}{(E[y^2])^2} - 3 = E\left[\left(\mathbf{w}^T z\right)^4\right] - 3$.

The updating process for $\mathbf{w}$ is:

$$\mathbf{w}_{necc} = \mathbf{w}_{old} - \eta \mathbf{E}\left[\mathbf{z}\left(\mathbf{w}_{old}^T \mathbf{z}\right)^3\right]$$

where η is a small constant to guarantee that $\mathbf{w}$ converges to the optimal solution. After each update, we normalize $\mathbf{w}_{new} = \frac{\mathbf{w}_{new}}{|\mathbf{w}_{new}|}$, and set $\mathbf{w}_{old} = \mathbf{w}_{new}$, and repeat the updating process until convergence. We can also use another algorithm to update the weight vector $\mathbf{w}$ [11].

However, kurtosis has also some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that kurtosis can be very sensitive to outliers (Huber, 1985) [4]. Its value may depend on only a few observations in the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of nongaussianity.

4.2.2 Negentropy

A second very important measure of non-gaussianity is given by negentropy. Negentropy is stand for negative entropy. Negentropy is based on the information theoretic quantity of (differential) entropy.

The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy. More rigorously, entropy is closely related to the coding length of the random variable, in fact, under some simplifying assumptions, entropy is the coding length of the random variable. For introductions on information theory, see e.g. (Cover and Thomas, 1991; Papoulis, 1991) [2]. Entropy H is defined for a discrete random variable Y as

$$H(Y) = -\sum_i P\left(Y = a_i\right) \log P\left(Y = a_i\right) \tag{17}$$

where the α are the possible values of Y. This very well-known definition can be generalized for continuous-valued random variables and vectors, in which case it is often called differential entropy. The differential entropy H of a random vector y with density f(y) is defined as (Cover and Thomas, 1991; Papoulis, 1991):

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) \mathrm{d}\mathbf{y} \tag{18}$$

A fundamental result of information theory is that a gaussian variable has the largest entropy among all random variables of equal variance. For a proof, see e.g. (Cover and Thomas, 1991; Papoulis, 1991). This means that entropy could be used as a measure of nongaussianity. In fact, this shows that the gaussian distribution is the "most random" or the least structured of all distributions. Entropy is small for distributions that are clearly concentrated on certain values, i.e., when the variable is clearly clustered, or has a pdf that is very "spiky".

To obtain a measure of nongaussianity that is zero for a gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(\mathbf{y}) = H\left(\mathbf{y}_{gauss}\right) - H(\mathbf{y}) \tag{19}$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix as y. Due to the above-mentioned properties, negentropy is always non-negative, and it is zero if and only if y has a Gaussian distribution.

The advantage of using negentropy, or, equivalently, differential entropy, as a measure of nongaussianity is that it is well justified by statistical theory. In fact, negentropy is in some sense the optimal estimator of nongaussianity, as far as statistical properties are concerned. The problem in using negentropy is, however, that it is computationally very difficult. Estimating negentropy using the definition would require an estimate (possibly nonparametric) of the pdf.

4.2.3 Approximations of negentropy

The classical method of approximating negentropy is using higher-order moments, for example as follows (Jones and Sibson, 1987) [7]:

$$J(y) \approx \frac{1}{12} E\left\{y^3\right\}^2 + \frac{1}{48} kurt(y)^2 \tag{20}$$

To avoid the problems encountered with the preceding approximations of negentropy, new approximations were developed in (Hyvärinen, 1998b) [5]. These approximation were based on the maximum-entropy principle. In general we obtain the following approximation:

$$J(y) \approx \sum_{i=1}^{p} k_i \left[E\left\{G_i(y)\right\} - E\left\{G_i(\mathrm{v})\right\}\right]^2 \tag{21}$$

where k_i are some positive constants, and v is a Gaussian variable of zero mean and unit variance (i.e., standardized). The variable y is assumed to be of zero mean and unit variance, and the functions G_i are some nonquadratic functions (Hyvärinen, 1998b) [5].

In the case where we use only one nonquadratic function G, the approximation becomes

$$J(y) \propto \left[E\{G(y)\} - E\{G(v)\}\right]^2 \tag{22}$$

for practically anynon-quadratic functionG. This is clearly a generalization of the moment-based approximation in (23), if y is symmetric. Indeed, taking G(y)=y4, one then obtains exactly (23), i.e. a kurtosis-based approximation.

But the point here is that by choosing G wisely, one obtains approximations of negentropy that are much better than the one given by (23). In particular,

choosing G that does not grow too fast, one obtains more robust estimators. The following choices of G have proved very useful:

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u, \quad G_2(u) = -\exp\left(-u^2/2\right) \tag{23}$$

where $1 \leq a_1 \leq 2$ is some suitable constant.

Thus we obtain approximations of negentropy that give a very good compromise between the properties of the two classical nongaussianity measures given by kurtosis and negentropy. They are conceptually simple, fast to compute, yet have appealing statistical properties, especially robustness. Therefore, we shall use these contrast functions in our ICA methods. Since kurtosis can be expressed in this same framework, it can still be used by our ICA methods. A practical algorithm based on these contrast function will be presented in Section 6.

4.3 Minimization of Mutual Information

Another approach for ICA estimation, inspired by information theory, is minimization of mutual information. We will explain this approach here, and show that it leads to the same principle of finding most nongaussian directions as was described above. In particular, this approach gives a rigorous justification for the heuristic principles used above.

4.3.1 Mutual Information

Using the concept of differential entropy, we define the mutual information I between m (scalar) random variables, yi,i = 1...m as follows

$$I\left(y_1, y_2, \ldots, y_m\right) = \sum_{i=1}^{m} H\left(y_i\right) - H(\mathbf{y}) \tag{24}$$

Mutual information is a natural measure of the dependence between random variables. In fact, it is equivalent to the well-known Kullback-Leibler divergence between the joint density f(y) and the product of its marginal densities; a very natural measure for independence. It is always non-negative, and zero if and only if the variables are statistically independent. Thus, mutual information takes into account the whole dependence structure of the variables, and not only the covariance, like PCA and related methods.

Mutual information can be interpreted by using the interpretation of entropy as code length. The terms $H(y_i)$ give the lengths of codes for the yi when these are coded separately, and H(y) gives the code length when y is coded as a random vector, i.e. all the componentsare coded in the same code. Mutual information thus shows what code length reduction is obtained by coding the whole vector instead of the separate components. In general, better codes can be obtained by coding the whole vector. An important property of mutual

information (Papoulis, 1991; Cover and Thomas, 1991)[2] is that we have for an invertible linear transformation:

$$I\left(y_1, y_2, \ldots, y_n\right) = \sum_i H\left(y_i\right) - H(\mathbf{x}) - \log\left|det\mathbf{W}\right| \tag{25}$$

In ICA, where $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ and $H(\mathbf{y}) = H(\mathbf{x}) + \log|\mathbf{W}|$. Now, let us consider what happens if we constrain the y_i to be uncorrelated and of unit variance. This means $E\left\{\mathbf{y}\mathbf{y}^T\right\} = \mathbf{W}E\left\{\mathbf{x}\mathbf{x}^T\right\}\mathbf{W}^T = \mathbf{I}$, which implies

$$det\mathbf{I} = 1 = \left(det\mathbf{W}E\left\{\mathbf{x}\mathbf{x}^T\right\}\mathbf{W}^T\right) = \left(det\mathbf{W}\right)\left(detE\left\{\mathbf{x}\mathbf{x}^T\right\}\right)\left(det\mathbf{W}^T\right) \tag{26}$$

and this implies that detW must be constant. Moreover, for y_i of unit variance, entropy and negentropy differ only by a constant, and the sign. Thus we obtain,

$$I\left(y_1, y_2, \ldots, y_n\right) = C - \sum_i J\left(y_i\right) \tag{27}$$

where C is a constant that does not depend on W. This shows the fundamental relation between negentropy and mutual information.

4.3.2 Defining ICA by Mutual Information

It is now obvious from (27) that finding an invertible transformation W that minimizes the mutual information is roughly equivalent to finding directions in which the negentropy is maximized. More precisely, it is roughly equivalent to finding 1-D subspaces such that the projections in those subspaces have maximum negentropy. Rigorously, speaking, (27) shows that ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities of the estimates, when the estimates are constrained to be uncorrelated. The constraint of uncorrelatedness is in fact not necessary, but simplifies the computations considerably, as one can then use the simpler form in (27) instead of the more complicated form in (25)[6].

4.4 Maximum Likelihood Estimation (MLE)

4.4.1 The likelihood

A very popular approach for estimating the ICA model is maximum likelihood estimation, which is closely connected to the infomax principle. Here we discuss this approach, and show that it is essentially equivalent to minimization of mutual information. It is possible to formulate directly the likelihood in the noise-free ICA model, which was done in (Pham et al., 1992) [9], and then estimate the model by a maximum likelihood method. Denoting by $\mathbf{W} = \left(\mathbf{w}_1, \ldots, \mathbf{w}_n\right)^T$ the matrix A1, the log-likelihood takes the form (Pham et al., 1992):

$$L = \sum_{t=1}^{T}\sum_{i=1}^{n} \log f_i\left(\mathbf{w}_i^T\mathbf{x}(t)\right) + T\log\left|det\mathbf{W}\right| \tag{28}$$

where the fi are the density functions of the si (here assumed to be known), and the x(t),t = 1,...,T are the realizations of x. The term log—detW— in the likelihood comes from the classic rule for (linearly) transforming random variables and their densities (Papoulis, 1991) [10]: In general, for any random vector x with density px and for any matrix W, the density of y = Wx is given by $p_x(\mathbf{Wx})|det\mathbf{W}|$.

Using MLE ICA, the objective is to find an unmixing matrix W that yields extracted signals y = Wx with a joint pdf as similar as possible to the joint pdf fs of the unknown source signals s.

4.4.2 The Infomax Principle

Infomax ICA is essentially a multivariate, parallel version of projection pursuit. Whereas projection pursuit extracts a series of signals one at a time from a set of M signal mixtures, ICA extracts M signals in parallel. This tends to make ICA more robust than projection pursuit.

It was derived from a neural network viewpoint in (Bell and Sejnowski, 1995 [1]; Nadal and Parga, 1994 [8]). This was based on maximizing the output entropy (or information flow) of a neural network with non-linear outputs. Assume that x is the input to the neural network whose outputs are of the form $\varphi\left(\mathbf{w}_i^T\mathbf{x}\right)$, where the φ are some non-linear scalar functions, and the wi are the weight vectors of the neurons. One then wants to maximize the entropy of the outputs:

$$L_2 = H\left(\varphi_1\left(\mathbf{w}_1^T\mathbf{x}\right), \ldots, \varphi_n\left(\mathbf{w}_n^T\mathbf{x}\right)\right) \tag{29}$$

If the φ are well chosen, this framework also enables the estimation of the ICA model. Indeed, severalauthors, e.g., (Cardoso, 1997; Pearlmutter and Parra, 1997), proved the surprising result that the principle of network entropy maximization, or "infomax", is equivalent to maximum likelihood estimation. This equivalence requires that the non-linearities i used in the neural network are chosen as the cumulative distribution functions corresponding to the densities f_i, i.e., $\varphi_i(.) = f_i(.)$

4.4.3 Connection to mutual information

To see the connection between likelihood and mutual information, consider the expectation of the log-likelihood: The mean of any random variable x can be calculated as $E[x] = \frac{1}{T}\sum_{t=1}^{T} x_t \Rightarrow \sum_{t=1}^{T} x_t = TE[x]$. Hence, Eq. (28) can be simplified to

$$\frac{1}{T}E\{L\} = \sum_{i=1}^{n} E\left\{\log f_i\left(\mathbf{w}_i^T\mathbf{x}\right)\right\} + \log|det\mathbf{W}| \tag{30}$$

Actually, if the f_i were equal to the actual distributions of $\mathbf{w}_i^T\mathbf{x}$, the first term would be equal to $-\sum_i H\left(\mathbf{w}_i^T\mathbf{x}\right)$. Thus the likelihood would be equal, up to an additive constant, to the negative of mutual information as given in Eq. (28).

Actually, in practice the connectionis even stronger. This is because in practice we don't know the distributions of the independent components. A reasonable approach would be to estimate the density of $\mathbf{w}_i^T\mathbf{x}$ as part of the ML estimation method, and use this as an approximation of the density of s_i. In this case, likelihood and mutual information are, for all practical purposes, equivalent.

Nevertheless, there is a small difference that may be very important in practice. The problem with maximum likelihood estimation is that the densities f_i must be estimated correctly. They need not be estimated with any great precision: in fact it is enough to estimate whether they are sub- or supergaussian (Cardoso and Laheld, 1996; Hyvärinen and Oja, 1998; Lee et al., 1999). In many cases, in fact, we have enough prior knowledge on the independent components, and we don't need to estimate their nature from the data. In any case, if the information on the nature of the independent components is not correct, ML estimation will give completely wrong results. Some care must be taken with ML estimation, therefore. In contrast, using reasonable measures of nongaussianity, this problem does not usually arise.

4.5 ICA and Projection Pursuit

Projection pursuit (Friedman and Tukey, 1974; Friedman, 1987; Huber, 1985; Jones and Sibson, 1987) is a technique developed in statistics for finding "interesting" projections of multidimensional data. Such projections can then be used for optimal visualization of the data, and for such purposes as density estimation and regression. In basic (1-D) projection pursuit, we try to find directions such that the projections of the data in those directions have interesting distributions, i.e., display some structure. It has been argued by Huber (Huber, 1985) and by Jones and Sibson (Jones and Sibson, 1987) that the Gaussian distribution is the least interesting one, and that the most interesting directions are those that show the least Gaussian distribution. This is exactly what we do to estimate the ICA model.

Thus, in the general formulation, ICA can be considered a variant of projection pursuit. All the nongaussianity measures and the corresponding ICA algorithms presented here could also be called projection pursuit "indices" and algorithms. In particular, the projection pursuit allows us to tackle the situation where there are less independent components s_i than original variables x_i is. Assuming that those dimensions of the space that are not spanned by the independent components are filled by gaussian noise, we see that computing the nongaussian projection pursuit directions, we effectively estimate the independent components. When all the nongaussian directions have been found, all the independent components have been estimated. Such a procedure can be interpreted as a hybrid of projection pursuit and ICA.

5 Preprocessing of data

Without those two step below, data can still be processed. However, those useful preprocessing will make the problem of ICA estimation simpler and better conditioned.

5.1 Centering

The most basic and necessary preprocessing is to center x, i.e. subtract its mean vector m = E(x) so as to make x a zero-mean variable. This implies that s is zero-mean as well.

After estimating the mixing matrix A with centered data, we can complete the estimation by adding the mean vector of s back to the centered estimates of s. The mean vector of s is given by $A^{-1}m$, where m is the mean that was subtracted in the preprocessing.

5.2 Whitening

After centering, we transform the observed vector x linearly so that we obtain a new vector $\tilde{x}$ which is white, i.e. its components are uncorrelated and their variances equal unity. In other words, the covariance matrix of $\tilde{x}$ equals the identity matrix:

$$\boldsymbol{E(\tilde{x}\tilde{x}^T) = I} \tag{31}$$

The whitening transformation is always possible. One popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix $E(xx^T) = EDE^T$, where **E** is the orthogonal matrix of eigenvectors of $E(xx^T)$ and D is the diagonal matrix of its eigenvalues, D= diag$(d_1, ..., d_n)$. Note that $E(xx^T)$can be estimated in a standard way from the available sample **x(1),...,x(T)**. Whitening can now be done by

$$\boldsymbol{\tilde{x} = ED^{-1/2}E^Tx} \tag{32}$$

where the matrix $D^{-1/2}$ is computed by a simple component-wise operation as $D^{-1/2} = diag(d_1^{-1/2}, ..., d_n^{-1/2})$. It is easy to check that now $E(\tilde{x}\tilde{x}^T) = I$.

Whitening transforms the mixing matrix into a new one, $\tilde{A}$.

$$\boldsymbol{\tilde{x} = ED^{-1/2}E^TAs = \tilde{A}s} \tag{33}$$

The utility of whitening resides in the fact that the new mixing matrix ˜ A is orthogonal. This can be seen from

$$\boldsymbol{E(\tilde{x}\tilde{x}^T) = \tilde{A}E(ss^T)\tilde{A}^T = \tilde{A}\tilde{A}^T = I} \tag{34}$$

Here we see that whitening reduces the number of parameters to be estimated. Instead of having to estimate the n^2 parameters that are the elements of the original matrix **A**, we only need to estimate the new, orthogonal mixing

matrix $\tilde{\mathbf{A}}$. An orthogonal matrix contains $n(n-1)/2$ degrees of freedom. For example, in two dimensions, an orthogonal transformation is determined by a single angle parameter. In larger dimensions, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix. Thus one can say that whitening solves half of the problem of ICA. Because whitening is a very simple and standard procedure, much simpler than any ICA algorithms, it is a good idea to reduce the complexity of the problem this way.

It may also be quite useful to reduce the dimension of the data at the same time as we do the whitening. Then we look at the eigenvalues d_j of $E\left\{\mathbf{x}\mathbf{x}^T\right\}$ and discard those that are too small, as is often done in the statistical technique of principal component analysis. This has often the effect of reducing noise. Moreover, dimension reduction prevents overlearning, which can sometimes be observed in ICA (Hyvärinen et al., 1999).

In the rest of this paper, we assume that the data has been preprocessed by centering and whitening. For simplicity of notation, we denote the preprocessed data just by $\mathbf{x}$, and the transformed mixing matrix by $\mathbf{A}$, omitting the tildes.

6 The FastICA Algorithm

In the preceding sections, we introduced different measures of nongaussianity, i.e. objective functions for ICA estimation. In practice, one also needs an algorithm for maximizing the contrast function, for example the one in (25). In this section, we introduce a very efficient method of maximization suited for this task. It is here assumed that the data is preprocessed by centering and whitening as discussed in the preceding section.

6.1 FastICA for one unit

To begin with, we shall show the one-unit version of FastICA. By a "unit" we refer to a computational unit, eventually an artificial neuron, having a weight vector $\mathbf{w}$ that the neuron is able to update by a learning rule. The FastICA learning rule finds a direction, i.e. a unit vector $\mathbf{w}$such that the projection $\mathbf{w}^T\mathbf{x}$ maximizes nongaussianity. Nongaussianity is here measured by the approximation of negentropy $J(\mathbf{w}^T\mathbf{x})$ given in (22). Recall that the variance of $\mathbf{w}^T\mathbf{x}$must here be constrained to unity; for whitened data this is equivalent to constraining the norm of $\mathbf{w}$ to be unity.

The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of $\mathbf{w}^T\mathbf{x}$, as measured in (22). It can be also derived as an approximative Newton iteration [A. Hyvärinen IEEE Tran 10(3):626-634, 1999]. Denote by g the derivative of the nonquadratic function G used in (22);

for example the derivatives of the functions in (23) are:

$$g_1(u) = \tanh(a_1 u), \quad g_2(u) = u \exp(-u^2/2) \tag{35}$$

where $1 \leq a_1 \leq 2$ is some suitable constant, often taken as $a_1 = 1$. The basic form of the FastICA algorithm is as follows:

1. Choose an initial (e.g. random) weight vector $\mathbf{w}$.
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+/\|\mathbf{w}^+\|$
4. If not converged, go back to 2.

Note that convergence means that the old and new values of $\mathbf{w}$ point in the same direction, i.e. their dot-product is (almost) equal to 1. It is not necessary that the vector converges to a single point, since $\mathbf{w}$ and $-\mathbf{w}$ define the same direction. This is again because the independent components can be defined only up to a multiplicative sign. Note also that it is here assumed that the data is pre-whitened.

The derivation of FastICA is as follows. First note that the maximize of the approximation of the negentropy of $\mathbf{w}^T\mathbf{x}$ are obtained at certain optima of $E\{G(\mathbf{w}^T\mathbf{x})\}$. According to the Kuhn-Tucker conditions [D. G. Luenberger. John Wiley Sons, 1969.], the optima of $E\{G(\mathbf{w}^T\mathbf{x})\}$ under the constraint $E\{(\mathbf{w}^T\mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$ are obtained at points where

$$E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w} = 0 \tag{36}$$

Let us try to solve this equation by Newton's method. Denoting the function on the left-hand side of (36) by F, we obtain its Jacobian matrix $JF(\mathbf{w})$ as

$$JF(\mathbf{w}) = E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{I} \tag{37}$$

To simplify the inversion of this matrix, we decide to approximate the first term in (37). Since the data is sphered, a reasonable approximation seems to be $E\{\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\} \approx E\{\mathbf{x}\mathbf{x}^T\}E\{g'(\mathbf{w}^T\mathbf{x})\} = E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{I}$. Thus the Jacobian matrix becomes diagonal, and can easily be inverted. Thus we obtain the following approximative Newton iteration:

$$\mathbf{w}^+ = \mathbf{w} - [E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w}]/[E\{g'(\mathbf{w}^T\mathbf{x})\} - \beta] \tag{38}$$

This algorithm can be further simplified by multiplying both sides of (38) by $\beta - E\{g'(\mathbf{w}^T\mathbf{x})\}$. This gives, after algebraic simplication, the FastICA iteration.

In practice, the expectations in FastICA must be replaced by their estimates. The natural estimates are of course the corresponding sample means. Ideally, all the data available should be used, but this is often not a good idea because the computations may become too demanding. Then the averages can be estimated using a smaller sample, whose size may have a considerable effect on the accuracy of the final estimates. The sample points should be chosen separately at every iteration. If the convergence is not satisfactory, one may then increase the sample size.

6.2 FastICA for several units

The one-unit algorithm of the preceding subsection estimates just one of the independent components, or one projection pursuit direction. To estimate several independent components, we need to run the one-unit FastICA algorithm using several units (e.g. neurons) with weight vectors $\mathbf{w}_1, ..., \mathbf{w}_n$.

To prevent different vectors from converging to the same maxima we must decorrelate the outputs $\mathbf{w}_1^T\mathbf{x}, ..., \mathbf{w}_n^T\mathbf{x}$ after every iteration. We present here three methods for achieving this.

A simple way of achieving decorrelation is a deflation scheme based on a Gram-Schmidt-like decorrelation. This means that we estimate the independent components one by one. When we have estimated p independent components, or p vectors $\mathbf{w}_1, ..., \mathbf{w}_p$, we run the one-unit fixed-point algorithm for $\mathbf{w}_{p+1}$, and after every iteration step subtract from $\mathbf{w}_{p+1}$the "projections" $\mathbf{w}_{p+1}^T\mathbf{w}_j\mathbf{w}_j, j = 1, ..., p$of the previously estimated p vectors, and then renormalize $\mathbf{w}_{p+1}$:

$$1.\ \text{Let } \mathbf{w}_{p+1} = \mathbf{w}_{p+1}..... = \mathbf{w}_{p+1}/\sqrt{\mathbf{w}_{p+1}^T\mathbf{w}_{p+1}} \tag{39}$$

In certain applications, however, it may be desired to use a symmetric decorrelation, in which no vectors are "privileged" over others [J. Karhunen, E. Oja,... 8(3):486-504, 1997.]. This can be accomplished, e.g., by the classical method involving matrix square roots,

$$\text{Let } \mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W} \tag{40}$$

where $\mathbf{W}$ is the matrix $(\mathbf{w}_1, ..., \mathbf{w}_n)^T$ of the vectors, and the inverse square root $(\mathbf{W}\mathbf{W}^T)^{-1/2}$ is obtained from the eigenvalue decomposition of $\mathbf{W}\mathbf{W}^T = \mathbf{F}\mathbf{\Lambda}\mathbf{F}^T$as $(\mathbf{W}\mathbf{W}^T)^{-1/2} = \mathbf{F}\mathbf{\Lambda}^{-1/2}\mathbf{F}^T$. A simpler alternative is the following iterative algorithm [A. Hyvärinen IEEE Tran 10(3):626-634, 1999],

$$1.\ \text{Let } \mathbf{W} = \mathbf{W}/\sqrt{\|.....32\mathbf{W} - \tfrac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}} \tag{41}$$

The norm in step 1 can be almost any ordinary matrix norm, e.g., the 2-norm

or the largest absolute row (or column) sum (but not the Frobenius norm).

6.3 FastICA and maximum likelihood

Finally, we give a version of FastICA that shows explicitly the connection to the well-known infomax or maximum likelihood algorithm introduced in [1,3,5,6]. If we express FastICA using the intermediate formula in (38), and write it in matrix form (see [A. Hyvärinen. Neural Processing Letters, 10(1):1-5, 1999.] for details), we see that FastICA takes the following form:

$$\mathbf{W}^+ = \mathbf{W} + \mathbf{\Gamma}[diag(-\beta_i) + E\{g(\mathbf{y})\mathbf{y}^T\}]\mathbf{W}. \tag{42}$$

where $\mathbf{y} = \mathbf{W}\mathbf{x}$, $\beta_i = E\{y_i g(y_i)\}$, and $\mathbf{\Gamma} = diag(1/(\beta_i - E\{g'(y_i)\}))$. The matrix $\mathbf{W}$ needs to be orthogonalized after every step. In this matrix version, it is natural to orthogonalize $\mathbf{W}$ symmetrically.

The above version of FastICA could be compared with the stochastic gradient method for maximizing likelihood [1,3,5,6]:

$$\mathbf{W}^+ = \mathbf{W} + \mu[\mathbf{I} + g(\mathbf{y})\mathbf{y}^T]\mathbf{W}. \tag{43}$$

where μ is the learning rate, not necessarily constant in time. Comparing (42) and (43), we see that FastICA can be considered as a fixed-point algorithm for maximum likelihood estimation of the ICA data model. For details, see [A. Hyvärinen. Neural Processing Letters, 10(1):1-5, 1999.]. In FastICA, convergence speed is optimized by the choice of the matrices $\mathbf{\Gamma}$ and $diag(-\beta_i)$. Another advantage of FastICA is that it can estimate both sub- and super-gaussian independent components, which is in contrast to ordinary ML algorithms, which only work for a given class of distributions (see Sec. 4.4).

6.4 Properties of the FastICA Algorithm

The FastICA algorithm and the underlying contrast functions have a number of desirable properties when compared with existing methods for ICA.

1. The convergence is cubic (or at least quadratic), under the assumption of the ICA data model (for a proof, see [A. Hyvärinen IEEE Tran 10(3):626-634, 1999]). This is in contrast to ordinary ICA algorithms based on (stochastic) gradient descent methods, where the convergence is only linear. This means a very fast convergence, as has been confirmed by simulations and experiments on real data (see [X. Giannakopoulos, J. Karhunen, and E. Oja.(ICANN'98), pages 651-656 1998.]).

2. Contrary to gradient-based algorithms, there are no step size parameters to choose. This means that the algorithm is easy to use.

3. The algorithm finds directly independent components of (practically) any non-Gaussian distribution using any nonlinearity g. This is in contrast to many algorithms, where some estimate of the probability distribution function has to be first available, and the nonlinearity must be chosen accordingly.

4. The performance of the method can be optimized by choosing a suitable nonlinearity g. In particular, one can obtain algorithms that are robust and/or of minimum variance. In fact, the two nonlinearities in (35) have some optimal properties; for details see [A. Hyvärinen IEEE Tran 10(3):626-634, 1999].

5. The independent components can be estimated one by one, which is roughly equivalent to doing projection pursuit. This es useful in exploratory data analysis, and decreases the computational load of the method in cases where only some of the independent components need to be estimated.

6. The FastICA has most of the advantages of neural algorithms: It is parallel, distributed, computationally simple, and requires little memory space. Stochastic gradient methods seem to be preferable only if fast adaptivity in a changing environment is required.

7 Applications of ICA

With the role of separating a multivariate signal into its underlying components, ICA has been applied in different areas such as audio processing, biomedical signal processing, image processing, telecommunications, and econometrics... More specifically, seperating of Artifacts in Magnetoencephalography (MEG) data, finding Hidden Factors in Financial Data, reducing Noise in Natural Images are some well-known applications using ICA technique.

In this report, we will construct an experiment focuses on the task of separating mixtures of sound signals into their underlying independent components using the technique of independent component analysis (ICA). We construct experiment as follows:

1. Prepare suitable sound sources with sampled at the rate of 22025 samples per second and approximately 10 seconds in length. In this experiment, we use 3 audio sources which are 1 human voice and 2 instrumental music sounds.

2. Create three mixing sounds from all sound sources.

3. The mixing data was passed through the ICA analyzer with the fast ICA sound separation algorithm (including centering, whitening, and estimation). As a result, the sound sources are separated independently.

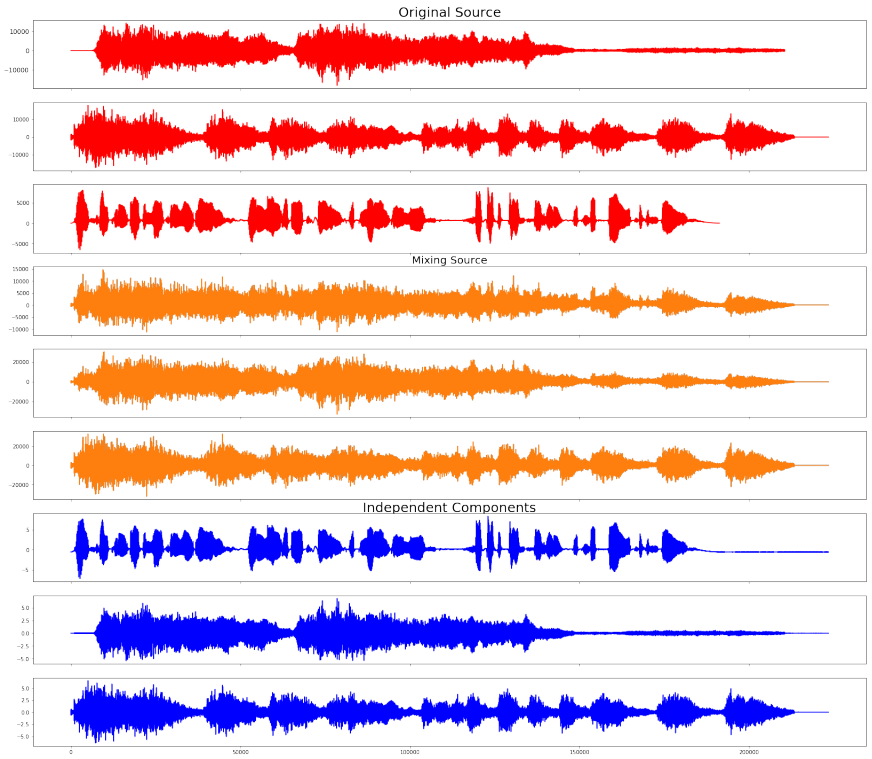

Figure 1: The Result of Experiment

We have collected good result. We got back 3 audio components with the same waveform as the original source and the output file sounds clear. For more details, please refer to our github link: `https://github.com/JVN2019/ICA_Linear_Report`

8 Conclusion

ICA is a widely-used statistical technique which is used for estimating independent components (ICA) through maximizing the non-Gaussianity of ICA, maximizing the likelihood of ICA, or minimizing mutual information between ICA. These approaches are approximately equivalent; however, each approach has its own limitations.

This paper followed the approach of not only explaining the steps for estimating ICA, but also presenting illustrative visualizations of the ICA steps to make it easy to understand. Moreover, a number of numerical examples are introduced and graphically illustrated to explain (1) how signals are mixed to form mixture signals, (2) how to estimate source signals, and (3) the preprocessing steps of ICA. Different ICA algorithms are introduced with detailed explanations. Moreover, ICA common challenges and applications are briefly highlighted.

References

[1] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[2] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of information theory*, 2:1–55, 1991.

[3] Nathalie Delfosse and Philippe Loubaton. Adaptive blind separation of independent sources: a deflation approach. *Signal processing*, 45(1):59–83, 1995.

[4] Peter J Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

[5] Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in neural information processing systems*, pages 273–279, 1998.

[6] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[7] M Chris Jones and Robin Sibson. What is projection pursuit? *Journal of the Royal Statistical Society: Series A (General)*, 150(1):1–18, 1987.

[8] Jean-Pierre Nadal and Nestor Parga. Nonlinear neurons in the low-noise limit: a factorial code maximizes information transfer. *Network: Computation in neural systems*, 5(4):565–581, 1994.

[9] DT Pahm, P Garrat, and C Jutten. Separation of a mixture of independent sources through a ml approach. In *Proc. European Signal Processing Conf*, page 771, 1992.

[10] Athanasios Papoulis. Probability, random variables, and stochastic processes. mcgraw-hill. *New York*, 19842:345–348, 1991.

[11] Alaa Tharwat. Independent component analysis: An introduction. *Applied Computing and Informatics*, 2018.