Independent Component Analysis
Theoretical Foundation

Marijo Simunovic

August 2023

1 Introduction

Independent Component Analysis (ICA) is an umbrella term for a set of mathematical and computational methods where the goal is to separate a multivariate signal (a mixture) into additive components (original signals). These signals are often referred to as independent components (IC). A detailed overview of different approaches can be found in [6]. ICA is often motivated as a method for solving *cocktail-party problem* [3] where we want to reconstruct the voices signals of n speakers based on the audio recordings of n microphones. Other possible uses include reconstruction of neural activity from EEG recordings [10], image sharpening [8] or face recognition [2].

We can think of ICA as an estimation of generative model [5]. The observables are n random variables x_1, x_2, ..., x_n. The assumption is that these random variables are formed as linear combination of n unknown (latent) variables s_1, s_2, ..., s_n. The $i-th$ component is formed as a mixture:

$$x_i = a_{i1} * s_1 + a_{i2} * s_2 + ... + a_{in} * s_n; \quad i = 1, 2, ..., n. \tag{1}$$

The mixture coefficients are also unknown. We can write down the same formulation in matrix notation:

$$\mathbf{x} = \mathbf{A} * \mathbf{s} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{s}$ are n-vectors and $\mathbf{A}$ is a mixture matrix where i-th row contains the mixture coefficients of the x_i-th variable.

2 Restrictions of ICA

The basic requirement for ICA methods to work is *independence.* It is assumed that the independent components are statistically independent. We say that the the random variables are independent if their joint probability distribution function (pdf) is equal to the product of their marginal distributions, i.e.

$$p(s_1, s_2, ..., s_n) = p_1(s_1) * p_2(s_2) * ... * p_3(s_n). \tag{3}$$

Two vectors s_1 and s_2 are linearly independent if there is no non-zero scalar a such that

$$a * s_1 + s_2 = 0 \tag{4}$$

Geometrically, vectors s_1 and s_2 do not lie on the same line. If two vectors are *statistically* independent their covariance is:

$$cov(s_1, s_2) = 0 \tag{5}$$

For two *linearly* dependent vectors

$$cov(s_1, s_2) = cov(\frac{1}{a} * s_1, s_2) = \frac{1}{a} var(s_2) \neq 0. \tag{6}$$

Therefore, two *linearly* dependent vectors cannot be statistically independent (linear independence of vectors, however, is not guarantee of their statistical independence). Furthermore, the distributions of independent components need to be *nongaussian.* ICA methods often rely on higher-order cumulants for the estimation of mixture coefficients. These higher-order cumulants are zero for gaussian distribution making the application impossible of ICA in this case.

We typically assume that the mixing matrix A is square, i.e. we model the problem in such way that the number of independent components is equal to the number of observed mixtures. Once the mixture matrix is known we can determine the original signals as

$$\mathbf{s} = \mathbf{A}^{-1} * \mathbf{x} \tag{7}$$

under assumption that the $\mathbf{A}$ is invertible. If independent components are time signals, i.e.

$$x(t) = A * s(t) \tag{8}$$

they not need to be nongaussian. In this case we can separate the ICs using either assumption that they have different autocovariances or that the variances of ICs are nonstationary (see chapter 18 in [6] for more comprehensive overview of methods which rely on time structure). Once these requirements are satisfied we can determine the original components up to trivial indeterminacies. Firstly, we cannot determine the variances of independent components. Both the ICs and mixture coefficients are unknown. Therefore, any scalar multiplier in sources can be canceled by dividing the corresponding entries in mixture matrix:

$$x_i = (\frac{1}{k_1} * a_{i1}) * (s_1 * k_1) + ... + (\frac{1}{k_n} * a_{in}) * (s_n * k_n)s; \quad i = 1, 2, ..., n. \tag{9}$$

Due to this fact, we typically fix the magnitudes of mixture components to have unit variance: $E\left[s^2\right] = 1$. This still leaves us with a sign ambiguity. In addition, we cannot determine order of independent components as we can arbitrarily permute the independent components:

$$\mathbf{x} = \mathbf{A} * \mathbf{P}^{-1} * \mathbf{P} * \mathbf{s} \tag{10}$$

Here $\mathbf{P}$ is permutation matrix (entries are 1s and 0s) and $\mathbf{P} * \mathbf{s}$ is the original vector $\mathbf{s}$ with changed order of entries s_i. Matrix $\mathbf{A} * \mathbf{P}^{-1}$ is a new mixture matrix $\mathbf{M}$ to be determined by ICA methods.

3 ICA and Gaussian Distributions

In case of Gaussian distribution first two moments are enough to sufficiently characterize the random variable. First moment is mean:

$$\mu_x = E\left[x\right] = \int_{-\infty}^{\infty} x * p_x(x)dx \tag{11}$$

and second is variance, defined as:

$$\sigma_x^2 = E\left[(x-\mu_x)^2\right] = \int_{-\infty}^{\infty} (x-\mu_x)^2 * p_x(x)dx \tag{12}$$

. By extension we define n-th order moment as:

$$v^n = E\left[(x-\mu_x)^n\right] = \int_{-\infty}^{\infty} (x-\mu_x)^n * p_x(x)dx. \tag{13}$$

From the Central Limit Theorem it follows that the distribution of the of the sum of centralized independent and identically distributed random variables converges against the standard normal distribution. The task of ICA is to reconstruct the original signals from a mixture. Therefore, when we are trying to reconstruct the mixing matrix we would like to check that the resulting signals are less Gaussian than the observed mixture. Another aspect worth considering is that the linear mixture of two Gaussian random variables results in new Gaussian distribution. This makes it impossible to infer if the original signals contained a single or multiple Gaussian distributed sources. Therefore, ICA methods are useful if only at most one source signal was Gaussian distributed (with already mentioned exception to the methods that use time structure of signals).

One measure of gaussianity is *kurtosis*. Kurtosis is fourth order momenth statistics:

$$\kappa = v_4 = \frac{E\left[(x-\mu_x)^4\right]}{\sigma^4} \tag{14}$$

The kurtosis of a Gaussian is equal to 3. A distribution with a positive kurtosis (> 3) is called super-Gaussian (or leptokurtic) and distribution with negative kurtosis (< 3) is sub-Gaussian (or platykurtic).

In practice, we use approximations of the above equations, with mean as

$$\mu_x = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{15}$$

variance as

$$\sigma^2(x) = \frac{1}{M}\sum_{i=1}^{M}(x_i-\mu_x)^2 \tag{16}$$

and finally kurtosis as:

$$\kappa(x) = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{x_i-\mu_x}{\sigma}\right)^4 \tag{17}$$

It is visible from the equation that the kurtosis is quite sensitive to the outliers. Entries far away from mean will dominate the summation due to the exponent. Instead of using kurtosis, it is common to measure the non-gaussianity using *negentropy*. Negentropy J is defined as:

$$J = H(y_{gauss}) - H(y) \tag{18}$$

where y_{gauss} is a Gaussian random variable of same covariance matrix as y. H denotes entropy of random variable. In discrete case it is defined as:

$$H(Y) = -\sum_i P(Y = a_i) log P(Y = a_i) \tag{19}$$

where a_i are all possible realizations of Y. Among all distributions of the same variance, Gaussian has the highest entropy [4]. Therefore, negentropy is always greater or equal to zero (in case y is also Gaussian). The more the distribution is concentrated on a specific set of values the lower is the entropy. This motivates the usage of negentropy as a measure of distance from the Gaussian distribution. There are problems with using negentropy directly in practice. Estimating negentropy would required an estimate of *pdf* which is often computationaly difficult problem. Classicaly, negentropy can be approximated as:

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

This approximation still suffers from the sensitivity of kurtosis to outliers. The researchers have therefore come up with more robust approximations based on the maximization of entropy [9], of the form:

$$J(y) \approx \left(E\left[G(y)\right] - E\left[G(\nu)\right]\right)^2 \tag{21}$$

Choosing G wisely, i.e. using a slowly growing function one obtains robust estimators:

$$G(u) = \frac{1}{a_1} logcosh(a_1 * u) \tag{22}$$

where $1 \leq a_1 \leq 2$, or

$$G(u) = -exp(-\frac{u^2}{2}) \tag{23}$$

4 Whitening

Given some random variables, it is straightforward to linearly transform them into uncorrelated variables. Two random variables are uncorrelated if their covariance is zero:

$$cov(y_1, y_2) = E\left[y_1 * y_2\right] - E\left[y_1\right] * E\left[* y_2\right] = 0 \tag{24}$$

When working with ICA methods we typically centralize the data so we can safely assume that the mean of random variables are equal to 0. In that case, covariance is equal to correlation and uncorelatedness is the same thing as zero correlation. A slightly stronger property, *whiteness*, of a zero mean random vector y means that the components of the vector are uncorrelated and that their variance is equal unity. Therefore, the covariance (and correlation) matrix is identity matrix I. We can use various methods to whiten a vector. Commonly, this is done using the SVD decomposition of covariance matrix or eigenvalue decomposition of covariance matrix. Whitening operation transform the original mixing matrix into a new, orthogonal mixing matrix V':

$$y = V * z = V * A * s = V' * s \tag{25}$$

This constraints the search space of a mixing matrix to a space of orthogonal matrices. An orthogonal matrix contains $\frac{n*(n-1)}{2}$ degrees of freedom. In higher dimensional spaces, orthogonal matrices have half a number of parameters in comparison to the arbitrary mixing matrix that we would like to estimate.

5 FastICA

Fast ICA [7] uses a fixed-point iteration scheme for finding maximum of non-gaussianity of $\mathbf{w}^T * x$ [7]. The input data x is assumed to be centered and whitened. We set the random weight vector w and normalize it such that $||w|| = 1$. Then we calculate the new weight vector as

$$w_{new} = E\left[x * G(w^T * x)\right] - E\left[G_{prim}(w^T * x)\right] * w \tag{26}$$

A detailed derivation of the vector update rule is given in [5]. The new weight vector is normalized and compared to the old value. The convergence is defined as a dot-product which is equal to 1 (within a given tolerance) ignoring the sign of the dot product as we can define the ICs only up to a multiplicative constant. Geometrical interpretation is that the angle between two consecutive vector updates stays under some predefined value.

This algorithm estimates only a single independent component. If we want to extend the algorithm for multiple components we also need to ensure that the weight vectors are decorrelated in order to prevent convergence to the same vector. A simple way to achieve this is Gramm-Schmidt method. If we have estimated up to w_{p-1} weight vectors, we can orthogonalize the vector w_p as follows:

$$w_p = w_p - \sum_{i=i}^{p-1} (w_p^T * w_i) * w_i \tag{27}$$

The alternative is to perform symmetric decorrelation of all vectors at the same time using following update rule:

$$W = (W * W^T)^{-\frac{1}{2}} * W \tag{28}$$

where W is unmixing matrix we are trying to estimate.

We tested the implementation with mixture of the images from recently published synthetic dataset [1] and two randomly generated signals with Gaussian and Poisson distribution, respectively. The notebook with the example is available under https://github.com/m5imunovic/studious-engine/notebooks/ICAApplication.ipynb

References

[1] Gwangbin Bae, Martin de La Gorce, Tadas Baltrušaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. Digiface-1m: 1 million digital face images for face recognition. In *2023 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2023.

[2] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13:1450–1464, 11 2002.

[3] Glen D. Brown, Satoshi Yamada, and Terrence J. Sejnowski. Independent component analysis at the neural cocktail party. *Trends in Neurosciences*, 24:54–63, 1 2001.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 9 2005.

[5] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 6 2000.

[6] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 5 2001.

[7] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 10 1997.

[8] Ivica Kopriva, Qian Du, Harold Szu, and Wasyl Wasylkiwskyj. Independent component analysis approach to image sharpening in the presence of atmospheric turbulence. *Optics Communications*, 233:7–14, 3 2004.

[9] Dominic Langlois, Sylvain Chartier, and Dominique Gosselin. An introduction to independent component analysis: Infomax and fastica algorithms. *Tutorials in Quantitative Methods for Psychology*, 6:31–38, 3 2010.

[10] Lisha Sun, Ying Liu, and P.J. Beadle. Independent component analysis of eeg signals. pages 219–222. IEEE, 2005.