The many algorithms of ICA and their complex-valued variants

Yusi Chen

October 13, 2021

Contents

Notations and formulation were adapted from Jon Schlens's tutorial [1]. Please refer to the original tutorial for detailed explanation and illustration examples. This note mainly focuses on different optimization algorithms used to maximize source independence and their complex-valued variants. The algorithm scripts will also be provided in the same Github repo. For the following contents, the first session briefly summarized the key assumptions and pre-processing of ICA (i.e. data whitening). The second session listed some popular algorithms used to maximize source independence. I also summarized their approximate objective functions, approximation assumptions for finite sampling issues and whether they are convex or not. The third session described complex-valued random variables and the application of ICA to complex-valued signals.

1 ICA formulation

1.1 Problem setup

ICA was designed to solve blind source separation (BSS) problem. It assumes that the observed signals $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is a **linear combination** of **independent** source signals $\mathbf{s} \in \mathbb{R}^{N\times 1}$:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \quad \hat{\mathbf{s}} = \mathbf{W}\mathbf{x} \tag{1}$$

where $\mathbf{A}$ is the mixing matrix and $\mathbf{W}$ is the demixing matrix. Often, we assume $\mathbf{A}$ is an invertible square matrix then $\mathbf{W} = \mathbf{A}^{-1}$. This is an under-constrained problem and the goal is to find a set of solutions $(\hat{\mathbf{s}}, \mathbf{W})$ such that the elements of $\hat{\mathbf{s}}$ are as independent as possible.

1.2 Data whitening

To solve ICA, we consider the diagonalization of the covariance matrix $\langle \mathbf{x}\mathbf{x}^T \rangle$. On one hand, from the data generation assumption and assume source distribution is whitened

$$\langle \mathbf{x}\mathbf{x}^T \rangle = \langle (\mathbf{A}\mathbf{s})(\mathbf{A}\mathbf{s})^T) \rangle = \mathbf{A}\langle \mathbf{s}\mathbf{s}^T \rangle \mathbf{A}^T = \mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{\Sigma}^{\mathbf{2}}\mathbf{U}^T \tag{2}$$

The last equation was obtained after considering the SVD decomposition of $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. On the other hand, the covariance matrix could be estimated through the sample covariance matrix of observed samples. Then apply eigenvalue decomposition of the calculated sample covariance matrix:

$$\langle \mathbf{x}\mathbf{x}^T \rangle = \mathbf{E}\mathbf{D}\mathbf{E}^T \tag{3}$$

By the uniqueness of diagonalization, $\mathbf{\Sigma} = \mathbf{D}^{\frac{1}{2}}$ and $\mathbf{U} = \mathbf{E}$.

Inverting $\mathbf{A}$, we arrived at the partial solution of $\mathbf{W}$:

$$\mathbf{W} = \mathbf{V}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T \tag{4}$$

Then the estimated source can be written as:

$$\hat{\mathbf{s}} = \mathbf{V}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T\mathbf{x} = \mathbf{V}\mathbf{x}_w \tag{5}$$

One way to evaluate independence is to calculate the mutual information or multi-information for multiple sources. From information theory, multiple information could also be calculated as the difference between marginal entropy sum and the entropy of joint distribution:

$$\mathbf{I}(\hat{\mathbf{s}}) = \sum_i H[(\mathbf{V}\mathbf{x}_w)_i] - H[\mathbf{V}\mathbf{x}_w] = \sum_i H[(\mathbf{V}\mathbf{x}_w)_i] - (H[\mathbf{x}_w] + \log_2 |\mathbf{V}|) \tag{6}$$

Then it becomes an optimization problem under the constrain that $\mathbf{V}$ is a unitary matrix.

$$\mathbf{V} = \arg\min_{\mathbf{V}} \sum_i H[(\mathbf{V}\mathbf{x}_w)_i] \tag{7}$$

There are a bunch of algorithms approximating the above entropy terms from finite samples. A typical example is FastICA. Infomax however adopts a slightly different objective function and optimizes through a self-organized network. All described in the next session.

2 Optimizing for statistical independence

After whitening, the ICA problem could be formulated as finding a rotation matrix (i.e. unitary matrix) $\mathbf{V}$ such that $\hat{\mathbf{s}}$, as the matrix product of $\mathbf{V}$ and whitened observation data $\mathbf{x}_w$, is as independent as possible.

Definition. Random vector $\mathbf{s}$ is independent given its joint probability could be factored into the product of its marginal distributions:

$$\mathbb{P}(\mathbf{s}) = \prod_i \mathbb{P}(s_i) \tag{8}$$

Then various algorithms differs in how to describe independence through high-order statistics and the approximation of p.d.f. I briefly summarized them in the following table and then introduced the assumptions one by one.

Algorithm	High-order statistics	Objective function	Algorithm	Script
Infomax	Mutual information	max joint entropy of $\hat{\mathbf{s}}$	Self-organizing net	runica.m
FastICA	negentropy/non-Gaussianity	contrast functions	fixed-point iteration	fastica.m

2.1 Infomax algorithm

Infomax [2] is an algorithm to maximize the mutual information between input $\mathbf{x}$ and output $\hat{\mathbf{s}}$ of a nonlinear neural network $\hat{\mathbf{s}} = g(\mathbf{W}\mathbf{x})$. For invertible continuous deterministic mappings (i.e. invertible g and $\mathbf{W}$), maximization of input-output mutual information is equivalent to minimization of the output joint entropy alone. So given the output joint entropy as objective function, the network weights $\mathbf{W}$ could be adjusted through a self-organized learning rule derived through gradient ascent:

$$\arg\max_{\mathbf{W}} H(\hat{\mathbf{s}}) \tag{9}$$

Then to apply infomax to ICA problem. Notice the relationship between mutual information and joint entropy:

$$\mathbf{I}(\hat{\mathbf{s}}) = \sum_i H(\hat{\mathbf{s}}_i) - H(\hat{\mathbf{s}}) \tag{10}$$

Let apart the influence of marginal entropies (which is the caveat of this algorithm),

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} H(\hat{\mathbf{s}}) = \arg\min_{\mathbf{W}} \mathbf{I}(\hat{\mathbf{s}}) \tag{11}$$

Note that for an one-to-one nonlinear transformation applied to a random variable, its p.d.f. or entropy remain the same up to a constant scale. Therefore the influence of introducing the nonlinear function g is trivial.

2.2 FastICA

This algorithm [3] starts from differential entropy (different from discrete-version entropy H): $h(y) = -\int f(y)(y)dy$. Negentropy (J) is a normalized version of differential entropy which is invariant for linear transformations:

$$J(y) = h(y_{\text{gauss}}) - h(y) \tag{12}$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix as y. Through negentropy, constraining the variables to be uncorrelated, we could express mutual information as:

$$\mathbf{I}(\mathbf{y}) = J(\mathbf{y}) - \sum_i J(y_i) \tag{13}$$

Then minimizing mutual information is roughly equivalent to finding directions in which the negentropy is maximized, given the joint p.d.f. is fixed. Note that FastICA focuses on mariginal p.d.f's while infomax focuses on the joint p.d.f.

Negentropy was approximated using the following general quadratic form, it has been shown to be more accurate than the conventional, cumulant-based approximations.

$$J(y_i) \simeq c[\mathbb{E}\{G(y_i)\} - \mathbb{E}\{G(\nu)\}]^2 \tag{14}$$

where G is any quadratic function, c is a constant, ν is a standard Gaussian random variable. Then if we try to find the independent components $\mathbf{w}_i$ (i.e. rows of $\mathbf{W}$) one by one, the objective function is defined as:

$$\begin{aligned} \text{maximize} \quad & \sum_i [\mathbb{E}\{G(\mathbf{w_i}^T\mathbf{x})\} - \mathbb{E}\{G(\nu)\}]^2 \quad \text{w.r.t.}\mathbf{w}_i \\ \text{under constraint} \quad & \mathbb{E}\{(\mathbf{w_k}^T\mathbf{x})(\mathbf{w_j}^T\mathbf{x})\} = \delta_{jk} \end{aligned} \tag{15}$$

The choice of the quadratic function was discussed in the original publication [3]. The optimization was implemented through the stabilized fixed-point algorithm derived through Kuhn-Tucker conditions and the approximate Newton iteration.

2.3

3 Complex-valued ICA

3.1 Complex-valued random vectors

For a complex-valued random vector $\mathbf{z} \in \mathbb{C}^{N \times 1}$, its p.d.f is determined by the p.d.f of a real-valued $2N \times 1$ random vector $(\mathbf{z}_R, \mathbf{z}_I)^T$. The first order statistics is given by:

$$\mathbb{E}(\mathbf{z}) = \mathbb{E}(\mathbf{z}_R) + j\mathbb{E}(\mathbf{z}_I) \tag{16}$$

The second order statistics were given by $\text{cov}(\mathbf{z}_R)$, $\text{cov}(\mathbf{z}_I)$ and cross covariance between $\mathbf{z}_R$ and $\mathbf{z}_I$. Therefore, the full description of second-order statistics needs both the real-valued covariance matrix $\text{cov}(\mathbf{z})$ and the complex-valued pseudo-covariance matrix $\text{pcov}(\mathbf{z})$:

$$\begin{aligned} \text{cov}(\mathbf{z}) &= \mathbb{E}_{\mathbf{z}}[(\mathbf{z} - \mathbb{E}_{\mathbf{z}}(\mathbf{z}))(\mathbf{z} - \mathbb{E}_{\mathbf{z}}(\mathbf{z}))^H] \\ \text{pcov}(\mathbf{z}) &= \mathbb{E}_{\mathbf{z}}[(\mathbf{z} - \mathbb{E}_{\mathbf{z}}(\mathbf{z}))(\mathbf{z} - \mathbb{E}_{\mathbf{z}}(\mathbf{z}))^T] \end{aligned} \tag{17}$$

A complex random variable $\mathbf{z}$ is called **circular** if for any deterministic $\phi \in [-\pi, \pi]$, the distribution of $e^{j\phi}\mathbf{z}$ equals the distribution of $\mathbf{z}$. The expectation of a circular random variable could only be zero or undefined. Additionally, the pseudo-variance of a circular random variable could only be zero or undefined.

3.2 Complex mutual information

For complex-valued random variable $\mathbf{z}$, the its mutual information is real-valued. It simply extends the definition for real-valued random variables:

$$\mathbf{I}(\mathbf{z}) = \mathbb{E}_{\mathbf{z}}[\log \frac{f_{\mathbf{z}}(\mathbf{z})}{\prod_{k=1}^{N} f_{z_k}(z_k)}] \tag{18}$$

The difficulty lies in taking the gradient of a real-valued objective function w.r.t. complex-valued argument (i.e. $\mathbf{W}$). One way to tackle this is to define complex (partial) differential operators and define the complex matrix gradient of $\mathbf{I}(\mathbf{z})$. See [4] for all the technical details.

3.3 Strong uncorrelating transform (SUT)

As in the real-valued case, it's better to pre-whitening the dataset so that we could reduce the optimization set from invertible matrices to orthogonal rotation matrices. Since the second-order statistics of complex-valued random vectors were determined by both covariance matrix and pseudo-covariance matrix, we need some extra efforts to whiten the data. It proved in [4] that any full complex random vector $\mathbf{z}$ can be linearly transformed using a nonsingular square matrix $\mathbf{C}^{-1}$ s.t. $\mathbf{s} = \mathbf{C}^{-1}\mathbf{z}$ has an identity covariance matrix and a diagonal pseudo-covariance matrix. The diagonal values of the pseudo-covariance matrix are unique and were named as **spectral coefficients**. Although strong uncorrelating transform may not be unique, at least one could be found by the following procedure:

- Find the usual whitening matrix $\mathbf{H} = \text{cov}(\mathbf{z})^{-\frac{1}{2}}$
- Perform symmetric SVD to the transformed pseudo-covariance matrix: $\text{pcov}(\mathbf{Hz}) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
- $\mathbf{C}^{-1} = \mathbf{U}^H\mathbf{H}$

3.4 Translate into real-valued ICA

Complex ICA problems could be translated into real-valued ICA based on whether the spectral coefficients are zero or not. And then all usual optimization algorithms could be applied to solve the problem.

4 References

1. Shlens, J. *A Tutorial on Independent Component Analysis* 2014. arXiv: `1404.2986 [cs.LG]`.
2. Bell, A. J. *et al.* An information-maximization approach to blind separation and blind deconvolution. *Neural computation* **7,** 1129–1159 (1995).
3. Hyvarinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks* **10,** 626–634 (1999).
4. Eriksson, J. *et al. Complex ICA for circular and non-circular sources* in *2005 13th European Signal Processing Conference* (2005), 1–4.