Independent Component Analysis

Kannan Lu[1] and Kittithat Krongchon[1]

[1]*Department of Physics, University of Illinois*
at Urbana-Champaign, Urbana, IL 61801, USA

(Dated: May 3, 2022)

Abstract

Indenpendent component analysis (ICA) is an unsupervised learning technique that is widely used in extracting independent factors in image, sound, and medical signals. In this document, we review the basic notions of the ICA including mathematical formulations and detailed algorithms. We also implement different algorithms based on higher-order statistics and information theoretic approaches. These algorithms are applied to sound signal disentanglement and electroencephalogram blinking signal removal with success.

INTRODUCTION

Independent component analysis (ICA) is one of the classical unsupervised learning techniques that is used to separate out latent variables or reduce dimensions. The motivation of the development of this technique is based on very commom problems. For instance, in reality, source signals are often corrupted with noise. In other words, data are composed of mutiple independent source signals. To isolate the signals from different sources, ICA can be a good choice. The famous example is the cocktail party problem, where the sound detectors record the superposition of various sound signals from different sources in the party. The goal is to separate out the observed signal into sounds from each different source. Since we do not know what the sources are and how these signals are mixed, this type of problem is called blind source separation (BSS) [1]. ICA is widely used in images, sounds, stock market, and medical signals, where latent independent variables are believed to exist. ICA can also be viewed as an extension of principal component analysis (PCA), where the latter maximizes the second order statistics (covariance matrix of data). ICA maximizes higher-order statistics or simply tries to look for independent components, not just uncorrelated components. In Fig. 1, we show the ICA components IC_i and PCA components PC_i learned from the observed data x_i with the original sources s_i sampled from uniform distribution independently. Clearly, the ICA learned the independent components correctly, but the PCA did not in this case.

In this paper, we review mathematical background of the ICA technique, the detailed algorithm implementation and several realizations in typical examples.

MATHEMATICAL BACKGROUND

Definition of the problem and notations

Before we delve into the detailed mathematical formalism, we first define the notations that we use throughout the whole document. We use boldface capital letters $\mathbf{A}$ for matrices, boldface lowercase letters $\mathbf{a}$ for vectors and ordinary lower case letters a for scalars. We use subscripts to denote the components of matrices (i.e. A_{ij}) and vectors (i.e. a_i). Notice that these letters are not in boldface as they mean the specific component, which is just a scalar in $\mathbb{R}$. Superscripts are reserved for indexing different samples. For instance, m samples of vector $\mathbf{a}$ can be denoted as $\mathbf{a}^{(k)}$ where $k = 1, 2, \ldots, m$. Following these conventions, we denote

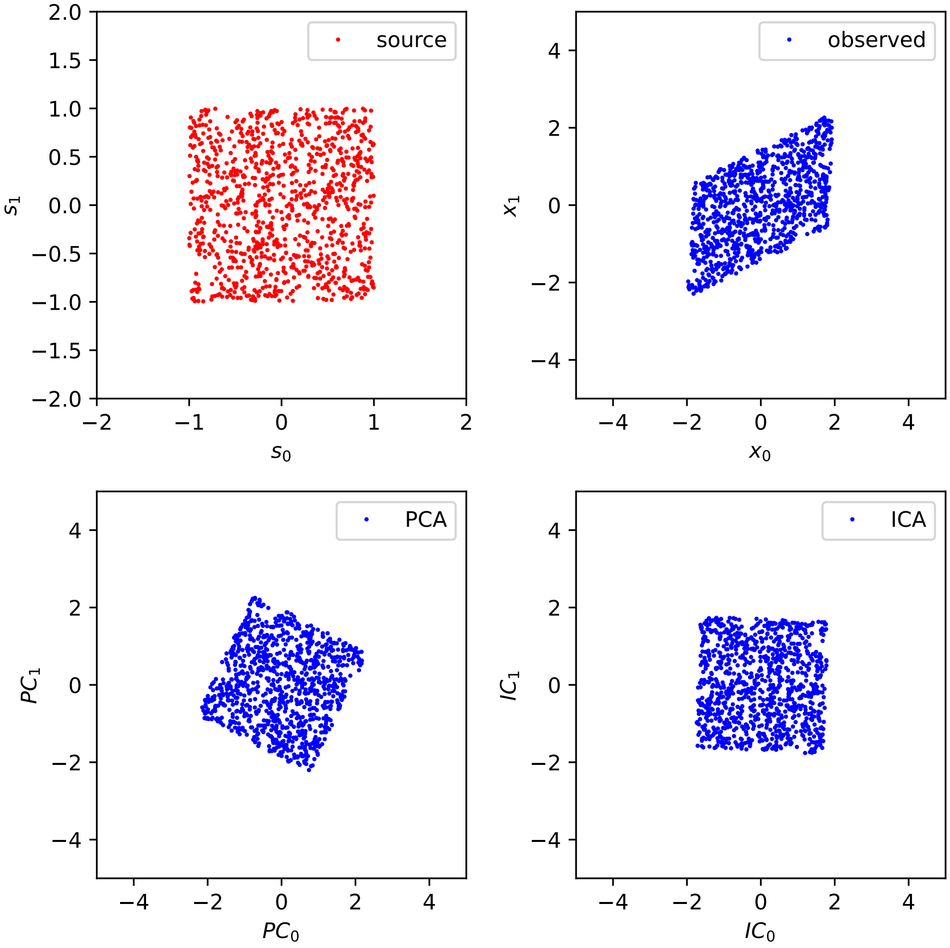

FIG. 1. shows the PCA and ICA result of unmixing the observed data x_i generated from two independent uniform variables s_i. The algorithm is based on maximizing the higher order moments.

the original signal to be $\mathbf{s}^{(i)} \in \mathrm{I\!R}^{\mathrm{d}}$. We consider the standard setting where the d dimensional signals $\mathbf{s}^{(i)} \in \mathrm{I\!R}^{\mathrm{d}}$ (d original sources) are mixed and observed by d detectors $\mathbf{x}^{(i)} \in \mathrm{I\!R}^{\mathrm{d}}$ (i.e. the cocktail party scenario). That is there is a linear map between the $\mathbf{s}^{(i)}$ and $\mathbf{x}^{(i)}$, $\mathbf{x}^{(i)} = \mathbf{A}\mathbf{s}^{(i)}$, where $\mathbf{A} \in \mathrm{I\!R}^{\mathrm{d}\times\mathrm{d}}$ is known as the mixing matrix. The whole idea of ICA is to learn the inverse of the mixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ from the observed data $\mathbf{x}^{(i)}$, known as the unmixing process. The learned independent components are denoted by $\mathbf{u} = \mathbf{W}\mathbf{x}$. This can be done in several different ways based on ideas of separating non-Gaussian signals and the original signals being independent. We will in this section discuss various mathematical formulations. All these various theoretical frameworks are constrained by the following conditions [1].

1. The number of sensors is larger than number of sources. This ensures that the mixing matrix is full rank.

2. The sources at each sample (time) are mutually independent.

3. At most one source is normally distributed.

Without these conditions, the BSS problem is ill-defined. For simplicity, we will also restrict the discussion to cases where the number of detectors and sources are the same. However, a generalization to the first condition is feasible, and the algorithms discussed in the following sections will still work.

Based on these definitions, we need to talk about some ambiguities embedded in the symmetry of the problem and justify that if the original signals are all Gaussians, they cannot be learned through the unmixing. There are two ambiguities that usually do not affect the practical application, namely the permutation ambiguity and the scaling ambiguity. It is trivial to see that the ordering of the original sources $\{s_j\}$ is ambiguous. In the case of scaling, if the original signal $\mathbf{s}$ is scaled by a non-zero constant to be $c\mathbf{s}$ where $c \neq 0$, then the mixing matrix $\mathbf{A}$ can be scaled by $1/c$ and will result in the same observed data $\mathbf{x}= \frac{1}{c}\mathbf{A}c\mathbf{s}$. This scaling ambiguity can be further extended to each component of the original signal. That is, for a particular component j, if we scale the component by c_j, the corresponding column of the mixing matrix can be scaled by $1/c_j$ to have the observed data unchanged (i.e. $x_i = \sum_j \frac{1}{c_j} A_{ij} c_j s_j$) [2].

The other ambiguity that matters for the practical application is that the original sources cannot all be distributed as Gaussians [1, 2]. To be more precise, in order to separate the independent components, we require that the original signals have at most only one component to be sampled from a Gaussian distribution, i.e. $s_j \sim \mathcal{N}(\mu, \sigma^2)$ for at most one j. Let's consider the case when all the original independent signals are Gaussians. That is, $s_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$ for $j = 1, 2, \ldots, d$. Then, given the mixing matrix $\mathbf{A}$, we have

$$\mathbb{E}[\mathbf{x}] = \mathbf{A}\mathbb{E}[\mathbf{s}] = \mathbf{A}\boldsymbol{\mu} \tag{1}$$

and

$$\mathrm{Cov}[\mathbf{x}] = \mathbb{E}[\mathbf{A}\mathbf{s}\mathbf{s}^t\mathbf{A}^t] - \mathbb{E}[\mathbf{A}\mathbf{s}]\mathbb{E}[(\mathbf{A}\mathbf{s})^t] = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^t - \mathbf{A}\mathbf{M}\mathbf{A}^t, \tag{2}$$

where $\mathbf{M}$ and $\boldsymbol{\Sigma}$ are diagonal matrices of $\mathrm{I\!R}^{\mathrm{d}\times\mathrm{d}}$ with diagonal elements to be $\{\mu_j^2\}$ and $\{\sigma_j^2\}$, respectively. Due to the scaling ambiguity, this is the same as considering normally distributed signals with unity variance. Since each column j of the mixing matrix $\mathbf{A}$ can be scaled by σ_j, $\tilde{\mathbf{A}}_{:j} = \sigma_j \mathbf{A}_{:j}$, and correspondingly, the signal needs to be scaled by $1/\sigma_j$, $\tilde{s_j} = \frac{s_j - \mu_j}{\sigma_j}$. Then,

$$\mathrm{Cov}[\mathbf{x}] = \tilde{\mathbf{A}}\mathbf{I}_{d\times d}\tilde{\mathbf{A}}^t. \tag{3}$$

Now, if we consider a rotation $\mathbf{R} \in O(d)$ acting on the scaled source signals $\tilde{\mathbf{s}}$, the mixing matrix then changes to $\tilde{\mathbf{A}}\mathbf{R}$. Upon this rotation, we would observe $\mathbf{x}'$ as $\mathbf{x}' = \tilde{\mathbf{A}}\mathbf{R}\tilde{\mathbf{s}}$, and $\mathbf{x}'$ is again normally distributed and has covariance matrix,

$$\mathrm{Cov}[\mathbf{x}'] = \tilde{\mathbf{A}}\mathbf{R}\mathbf{I}_{d\times d}\mathbf{R}^t\tilde{\mathbf{A}}^t = \tilde{\mathbf{A}}\mathbf{I}_{d\times d}\tilde{\mathbf{A}}^t. \tag{4}$$

This simply means that whether the sources are rotated or not the observed data will be distributed as $\mathcal{N}(0, \tilde{\mathbf{A}}\tilde{\mathbf{A}}^t)$. Thus, because of the rotational symmetry of the multivariate Gaussian distribution, we cannot separate and obtain the original source signals. These derivations also indicate that given the observed data $\mathbf{x}$, we can whiten (apply PCA rotation and normalize each principal component) the data, and it does not affect the separated signals (up to scalings and mean translations). We will assume the observed data $\mathbf{x}$ are whitened if not explicitly stated in later discussions. That is $\tilde{\mathbf{x}} = \mathbf{K}(\mathbf{x} - \boldsymbol{\mu})$, where $\mathbf{K}$ is the whitening transformation. Then, to recover the source signals, we need to find $\mathbf{W}$ with orthonormal rows $\{\mathbf{w}_i^t\}$ such that $\mathbf{s} = \mathbf{W}\mathbf{K}\mathbf{x}$.

Higher order statistics approach

The non-Gaussianity directly provides us with the methodologies used in finding the independent component. Notice that the PCA only finds the uncorrelated components but not necessarily independent. The ICA tries to find independent components where the joint distributions can be factorized into marginal distributions by investigating higher-order moments. There are multiple moment-based objective functions that have been used along the development of ICA. We define some of the notations and discuss one of the moment-based objective functions in detail.

As we discussed previously, we whiten the observed signals and we denote the whitened observed data as $\mathbf{x}$. The kurtosis and excess kurtosis of whitened variable x are defined as

$$\beta(x) := \mathbb{E}[x^4], \ \kappa(x) := \beta(x) - 3. \tag{5}$$

The standard normal random variable has excess kurtosis to be 0. When excess kurtosis $\kappa(x) < 0$, x is said to be sub-Gaussian (flat around the center, e.g. uniform distribution). When $\kappa(x) > 0$, x is super-Gaussian (heavy tail and sharp peak around the center, e.g. Laplace distribution). The objective function to maximize can be $|\kappa(x)|$ and $\kappa^2(x)$ etc. This is further backed up by the following inequalities [3].

$$\forall i \in \{1, \ldots, d\}, \ |\kappa(\mathbf{w}_i^t \mathbf{x})| \leq \max\{|\kappa(z_1)|, \ldots, |\kappa(z_d)|\}. \tag{6}$$

$$|\kappa(\mathbf{w}_1^t \mathbf{x})| + \cdots + |\kappa(\mathbf{w}_d^t \mathbf{x})| \leq |\kappa(z_1)| + \cdots + |\kappa(z_d)|, \tag{7}$$

where z_i's are the standardized independent components.

Thus, the problem is equivalent to finding an orthogonal transformation $\mathbf{W}$ with orthonormal rows $\{\mathbf{w}_i^t\}$ such that the objective function can be maximized. In the case of the objective function $L(\mathbf{w}) = |\kappa(\mathbf{w}^t \mathbf{x})|$, this amounts to iterating the $\mathbf{w}$ with gradient [4],

$$\frac{\partial |\kappa(\mathbf{w}^t \mathbf{x})|}{\partial \mathbf{w}} = 4 \ \text{sign}[\kappa(\mathbf{w}^t \mathbf{x})]\{\mathbb{E}[\mathbf{x}(\mathbf{w}^t \mathbf{x})^3] - 3\mathbf{w}||\mathbf{w}||^2\}. \tag{8}$$

This objective function captures both sub-Gaussian and super-Gaussian distributions for the source signals but the kurtosis is sensitive to the outliers of the observed data. Hence, some adaptive objective functions are used in practice more extensively based on negentropy [4].

Information theoretic approach

Apart from the moment-based approaches, there are several information theoretic methods. In this section, we will review all different information theoretic frameworks and show that essentially they are equivalent. Then we will talk about the learning rules for these theoretical formulations in the subsequent section. The centralized quantity in this formalism is the mutual information, which is defined as the Kullback–Leibler (KL) divergence of the multivariate distribution of the observed data $\mathbf{x}$ and product of all its marginal distributions, i.e.

$$I(\mathbf{x}) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_i p_i(x_i)}. \tag{9}$$

$I(\mathbf{x})$ is non-negative and is zero if and only if the $\mathbf{x}$ are independent. Several different information theoretic approaches have been formulated in history and can be unified in the concept of minimizing the mutual information of the separated components $\mathbf{u} = \mathbf{W}\mathbf{x}$.

One formulation in neural networks is to maximize information between inputs $\mathbf{x}$ and outputs $\mathbf{y}$, which implies that the output distributions are factorized and thus minimize the mutual information in the outputs $\mathbf{y}$. Maximizing the information between inputs and outputs is to maximize the output joint entropy $H(\mathbf{y}) = \sum_i H(y_i) - I(\mathbf{y})$, where $I(\mathbf{y})$ is the mutual information in the outputs $\mathbf{y}$. The mutual information is non-negative and is zero if and only if $\mathbf{y}$ are marginalized. Recall that in neural networks, the output is given by the nonlinearity $y_i = g_i(\mathbf{w}_i^t\mathbf{x}) := g_i(u_i)$. Thus,

$$p(y_i) = \left|\frac{\partial g_i}{\partial u_i}\right|^{-1} p(u_i). \tag{10}$$

The maximum is therefore obtained by considering the gradient,

$$\frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} = \frac{\partial(-I(\mathbf{y}))}{\partial \mathbf{W}} - \frac{\partial}{\partial \mathbf{W}} \sum_i \mathbb{E}\left[\log\left\{\left|\frac{\partial g_i}{\partial u_i}\right|^{-1} p(u_i)\right\}\right]. \tag{11}$$

This implies that the nonlinear function g_i in the neural networks, $y_i = g_i(\mathbf{w}_i^t\mathbf{x})$ is a cdf of the source distribution s_i in order to kill the second term. Together with the constraint that the outputs are marginalized, this gradient is zero. Therefore, a good estimation of the cdf of the source signal improves drastically the performance of ICA. Notice that, if $I(\mathbf{y}) = 0$, the $\mathbf{u} = \mathbf{W}\mathbf{x}$ should also satisfy $I(\mathbf{u}) = 0$ as g_i is an invertible monotonic function. Hence, the neural network approach can be eventually reduced to minimizing the mutual information $I(\mathbf{u})$ [1].

Another way to formulate this is to maximize the negentropy $J(u_i)$, which is the KL divergence $D(p(u_i)||p_G(u_i))$ between $p(u_i)$ and Gaussian distribution $p_G(u_i)$ with the same mean and covariance as $p(u_i)$. Recall that the Gaussian distribution has maximum entropy constrained with the mean and covariance. The negentropy thus measures non-Gaussianity, which is equivalent to higher order moment approach in principle. Requiring that the $\mathbf{u}$ can be factorized and decorrelated, the sum of negentropies can be written as

$$
\begin{aligned}
\sum_i J(u_i) &= \sum_i D(p(u_i)||p_G(u_i)) && (12)\\
&= \sum_i p(u_i)\log\frac{p(u_i)}{p_G(u_i)} && (13)\\
&= \sum_{\mathbf{u}} p(\mathbf{u})\log\frac{\prod_i p(u_i)}{\prod_i p_G(u_i)} && (14)\\
&= \sum_{\mathbf{u}} p(\mathbf{u})\log\frac{\prod_i p(u_i)}{p_G(\mathbf{u}_i)} && (15)\\
&= \sum_{\mathbf{u}} p(\mathbf{u})\log\frac{\prod_i p(u_i)}{p(\mathbf{u})} + \sum_{\mathbf{u}} p(\mathbf{u})\log\frac{p(\mathbf{u})}{p_G(\mathbf{u})} && (16)\\
&= D\left(\prod_i p(u_i)||p(\mathbf{u})\right) + J(\mathbf{u}) && (17)\\
&= -I(\mathbf{u}) + J(\mathbf{u}) && (18)\\
&= -I(\mathbf{u}) - H(\mathbf{u}) - \sum_{\mathbf{u}} p(\mathbf{u})\log p_G(\mathbf{u}) && (19)\\
&= -I(\mathbf{u}) - H(\mathbf{x}) - \log(|\det(\mathbf{W})|) - \frac{1}{2}\log((2\pi e)^d\det(\langle\mathbf{u},\mathbf{u}^t\rangle)) && (20)\\
&= -I(\mathbf{u}) - H(\mathbf{x}) - \frac{1}{2}\log((2\pi e)^d). && (21)
\end{aligned}
$$

In the derivation, the $\mathbf{u}$ are uncorrelated so the covariance matrix is identity. Therefore, maximizing the negentropy is equivalent to minimizing mutual information in $\mathbf{u} = \mathbf{W}\mathbf{x}$ [1]. In practice, the negentropy is difficult to evaluate so approximation schemes have been developed and have been discussed in the previous section.

Lastly, for the maximum likelihood estimation approach, we want to maximize the log likelihood over all samples observed upon choosing a parametrized distribution $\hat{p}_s(\mathbf{w}_i^t\mathbf{x})$ satisfying $p(\mathbf{x}) = \prod_{i=1}^d \hat{p}_s(\mathbf{w}_i^t\mathbf{x})|\mathbf{W}|$ [2]. The log-likelihood is

$$
l(\mathbf{W}) = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{d}\log(\hat{p}_s(\mathbf{w}_i^t\mathbf{x}^{(j)})) + \log|\mathbf{W}|\right). \qquad (22)
$$

If the approximation $\hat{p}_s(\mathbf{w}_i^t\mathbf{x})$ are close to the actual pdf, the first term is approximately $-\sum_i H(\mathbf{w}_i^t\mathbf{x})$ and is equal to the negative of the mutual information up to an additive constant of the total entropy of $\mathbf{x}$. Hence, all these historical information theoretic approaches are equivalent and the key idea is to approximate the probability distribution of the sources correctly. Historically, there has been several proposed parametric distributions for approximating super-Gaussian or sub-Gaussian distributions [1]. In the following section, we will discuss the algorithms associated with the moment-based and information-based approaches.

ALGORITHMS

In this section, we discuss some of the common algorithms based on the previous mathematical formalism. We categorize them into moment-based approaches and information-based approaches. In either case, the problem is equivalent to an optimization problem. There are some detailed differences on how this optimization problem is implemented, i.e. ordinary gradient descent, Newton's method or fixed point algorithm. We simply specify what we implemented for various examples that we will discuss later.

Moment-based approach

In this method, we maximize the non-Gaussianity given by

$$J(u) \propto \{\mathbb{E}[G(u)] - \mathbb{E}[G(\nu)]\}^2, \tag{23}$$

where u is a random variable of zero mean and unit variance, which can be achieved by whitening the data, and ν is a Gaussian variable also of zero mean and unit variance. The function forms of G should not grow too fast to be robust. The following choice of G is proposed by Hyvärinen and Oja [4].

$$G(u) = \frac{1}{a_1} \log(\cosh a_1 u). \tag{24}$$

$$g(u) = \partial_u G(u) = \tanh(a_1 u). \tag{25}$$

We want to find the extrema of $\mathbb{E}[G(u)]$ to maximize $J(u)$ under the constraints

$$\mathbb{E}[(\mathbf{w}^t \mathbf{x})^2] = \|\mathbf{w}\|^2 = 1. \tag{26}$$

From the Kuhn–Tucker conditions, the extremum condition is satisfied when

$$\mathbb{E}[\mathbf{x} g(\mathbf{w}^t \mathbf{x})] - \beta \mathbf{w} = \mathbf{0}. \tag{27}$$

This equation can be solved by using Newton's method. Let $\mathbf{F}(\mathbf{w})$ denote the left-hand side of the equation, which we are trying to solve. We update the value of $\mathbf{w}$ in each iteration according to the following equation.

$$\begin{aligned}
\mathbf{w}_{n+1} &= \mathbf{w}_n - \frac{\mathbf{F}(\mathbf{w}_n)}{F'(\mathbf{w}_n)} && (28)\\
&= \mathbf{w}_n - \frac{\mathbb{E}[\mathbf{x}g(\mathbf{w}_n^t\mathbf{x})] - \beta\mathbf{w}_n}{\mathbb{E}[g'(\mathbf{w}_n^t\mathbf{x})] - \beta} && (29)\\
&= \frac{\mathbf{w}_n\mathbb{E}[g'(\mathbf{w}_n^t\mathbf{x})] - \beta\mathbf{w}_n - \mathbb{E}[\mathbf{x}g(\mathbf{w}_n^t\mathbf{x})] + \beta\mathbf{w}_n}{\mathbb{E}[g'(\mathbf{w}_n^t\mathbf{x})] - \beta} && (30)\\
&= \frac{\mathbf{w}_n\mathbb{E}[g'(\mathbf{w}_n^t\mathbf{x})] - \mathbb{E}[\mathbf{x}g(\mathbf{w}_n^t\mathbf{x})]}{\mathbb{E}[g'(\mathbf{w}_n^t\mathbf{x})] - \beta}. && (31)
\end{aligned}$$

Information theoretic approach

Based on what has been discussed in the previous section, we can approximate the source signal distribution with a parametrized pdf. There are multiple choices here tailored to different problems. A default choice is to assume the cdf of the source signal can be approximated by a sigmoid function

$$g(s_i) = \frac{1}{1+e^{-s_i}}. \tag{32}$$

Thus, the pdf is $p(s_i) = g'(s_i)$. Notice that this pdf is super-Gaussian. This means that ICA based on this pdf works well when the distribution of original sources have heavy tails and sharp peak around center. Recall the log-likelihood (without normalization) [2],

$$\begin{aligned}
l(\mathbf{W}) &= \sum_{j=1}^{N}\left(\sum_{i=1}^{d}\log(\hat{p}_s(\mathbf{w}_i^t\mathbf{x}^{(j)})) + \log|\mathbf{W}|\right) && (33)\\
&= \sum_{j=1}^{N}\left(\sum_{i=1}^{d}\log(g'(\mathbf{w}_i^t\mathbf{x}^{(j)})) + \log|\mathbf{W}|\right), && (34)
\end{aligned}$$

where $g'(\mathbf{w}_i^t\mathbf{x}) = g'(\mathbf{u}_i) = g(1-g)$. Thus, the gradient of log-likelihood is

$$\frac{\partial l(\mathbf{W})}{\partial \mathbf{W}} = \sum_{j=1}^{N}[\mathbf{q}^{(j)}\mathbf{x}^{t,(j)} + (\mathbf{W}^t)^{-1}], \tag{35}$$

where $\mathbf{q}^{t,(j)} = [1-2g(u_1^{(j)}),\ldots,1-2g(u_d^{(j)})]$. In our implementation, we used the stochastic gradient descent method to update $\mathbf{W}$:

$$\mathbf{W} = \mathbf{W} + \alpha[\mathbf{q}^{(j)}\mathbf{x}^{t,(j)} + (\mathbf{W}^t)^{-1}]. \tag{36}$$

This means that we update $\mathbf{W}$ for each sample of the observed data instead of calculating the exact summation for all samples at each step of iteration. This iteration scheme is accompanied by randomizing the order of samples for each iteration (making the iteration more stochastic).

APPLICATIONS

We implement the moment-based approach according to the formalism described in the previous section and apply it to the observed data $\mathbf{x}^{(i)}$ in order to find the source matrix $\mathbf{s}^{(i)}$ given by

$$\mathbf{S} = \begin{bmatrix} s_1(t_1) & s_1(t_2) & \cdots & s_1(t_N) \\ s_2(t_1) & s_2(t_2) & \cdots & s_2(t_N) \\ \vdots & \vdots & \ddots & \vdots \\ s_d(t_1) & s_d(t_2) & \cdots & s_d(t_N) \end{bmatrix}, \tag{37}$$

where N is the number of time steps, d is the number of detectors, and s_i's are independent signal sources.

Application 1: sine, square, and sawtooth waves

In this first example, we define the independent signal sources according to the following equations and plot them in Fig. 3(a).

$$\begin{aligned} s_1 &= \sin(2t). \\ s_2 &= 2 \ \mathrm{sign}[\sin(3t)]. \\ s_3 &= 4 \ \left[\frac{1}{\pi}(2\pi t \bmod 2\pi) - 1\right]. \end{aligned} \tag{38}$$

The mixture signals $\mathbf{x}^{(i)}$ (Fig. 3(b)) that we want to apply the moment-based algorithm on is constructed by the equation

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{39}$$

where $\mathbf{A}$ is the mixing matrix, which in this example, is set to be

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{2} & 2 & 1 \\ \frac{3}{2} & 1 & 2 \end{bmatrix}. \tag{40}$$

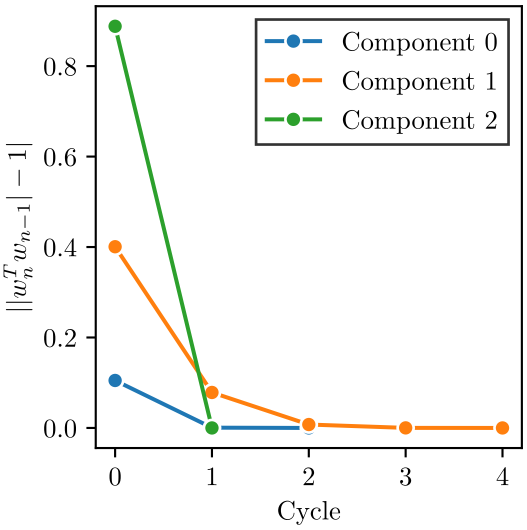

FIG. 2. The convergence criterion $||\mathbf{w}_n^t \mathbf{w}_{n-1}| - 1|$ as a function of cycle for each row of W.

Newton's method converages rather quickly within only four cycles for each row of **W** as shown in Fig. 2. The sources predicted by the algorithm as implemented in our code and in sklearn.decomposition.FastICA are shown in Fig. 3(d, e). Both of the results are very similar in terms of the ability to separate out independent sources. The main difference is the sign of the sawtooth wave because independence is up to a minus sign as discussed previously in the formalism section. From the predicted sources $\mathbf{s}_{\text{predicted}}$, we can verify the accuracy of FastICA results by inverting the mixing and whitening transformations and adding back the mean of each mixture to retrieve the input signals.

$$\mathbf{x}_{\text{retrived}} = (\mathbf{WK})^{-1}\mathbf{s}_{\text{predicted}} + \mathbb{E}[\mathbf{x}], \tag{41}$$

where **K** is the whitening transformation. The retrieved signals $\mathbf{x}_{\text{retrieved}}$ are shown to be equal to the original input signals $\mathbf{x}$ (Fig. 3(c)), which confirms that we can exactly reproduce the observed data from the output of ICA. In some applications, such as separation of independent sound sources, we might not need to retrieve the original mixtures. However, this procedure becomes crucial in other applications, such as blink removals of the electroencephalogram (EEG) data. The discussion of both mentioned applications follows.

In this specific example, the result from sklearn.decomposition.PCA is also plotted in Fig. 3(f). The figure shows that PCA is not suitable for separating independent sources.

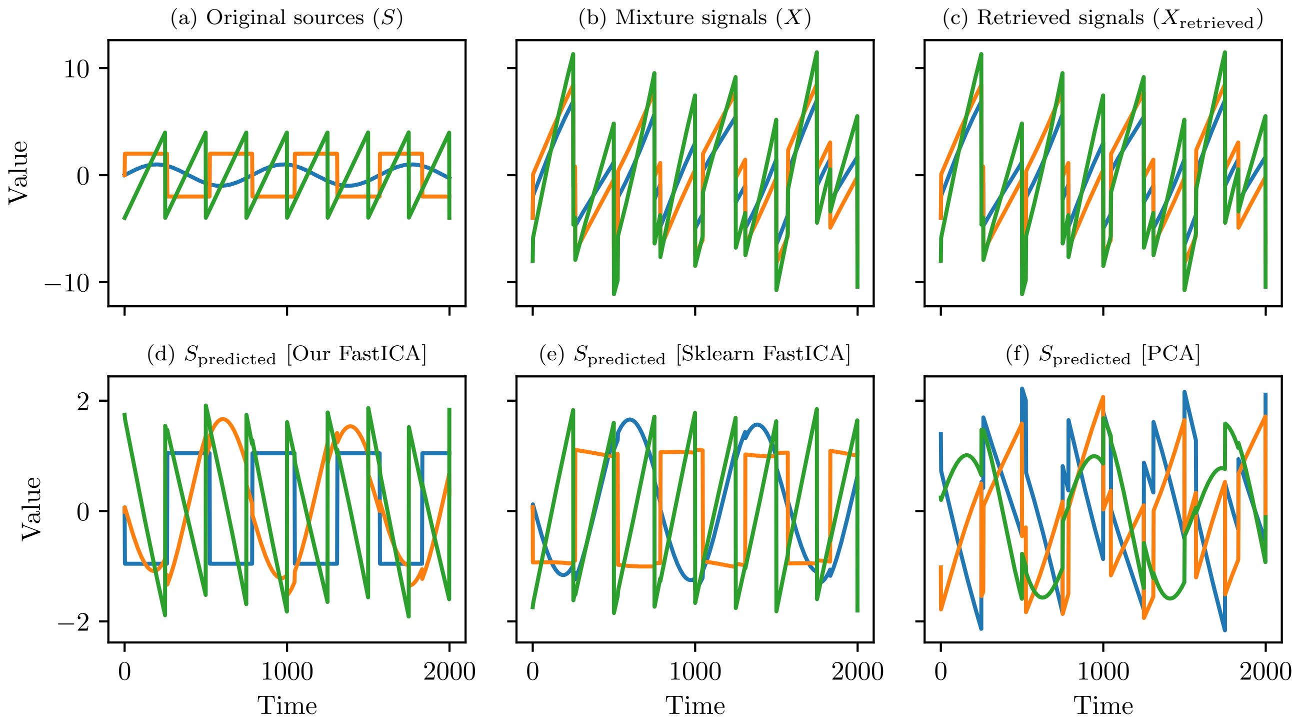

FIG. 3. (a) Original sources as defined in Eq. (38). (b) Mixture signals whose mixing matrix is defined by Eq. (40). (c) Retrieved input signals from the predicted sources according to Eq. (41). (d) Separated sources using our implementation of FastICA. (e) Separated sources using sklearn.decomposition.FastICA. (f) Separated sources using sklearn.decomposition.PCA.

Application 2: audio source separation

We employ the maximum likelihood algorithm to separate out the independent components of five mixed sound tracks [2]. The mixed sound waves are shown in Fig. 4 left panels (red). The separated independent sound tracks are shown in the right panels (blue) in Fig. 4. The separated sound tracks can be played and listened one by one to verify that they are meaningful.

Application 3: blinking removal in EEG signals

In Fig. 5, we decompose the EEG signals [5] from 64 detectors into 64 independent components. We are able to identity the 48th component to be associated with blinking signals because of the four visible peaks. The 48th component shows a different time structure in Fig. 5. We reconstruct the signals $\mathbf{x}_{\mathrm{retrived}}$ using Eq. 41 with the 48th component set to zero. The final result is shown in Fig. 6 in orange.

CONCLUSIONS

In this review paper of ICA, we discuss the traditional and basic notions of this technique. We also implement the algorithms and utilize our codes in three different applications. We notice that the objective function plays an important role in the performance of ICA in various examples we examined. In the basic setting, the performance of the algorithm lies

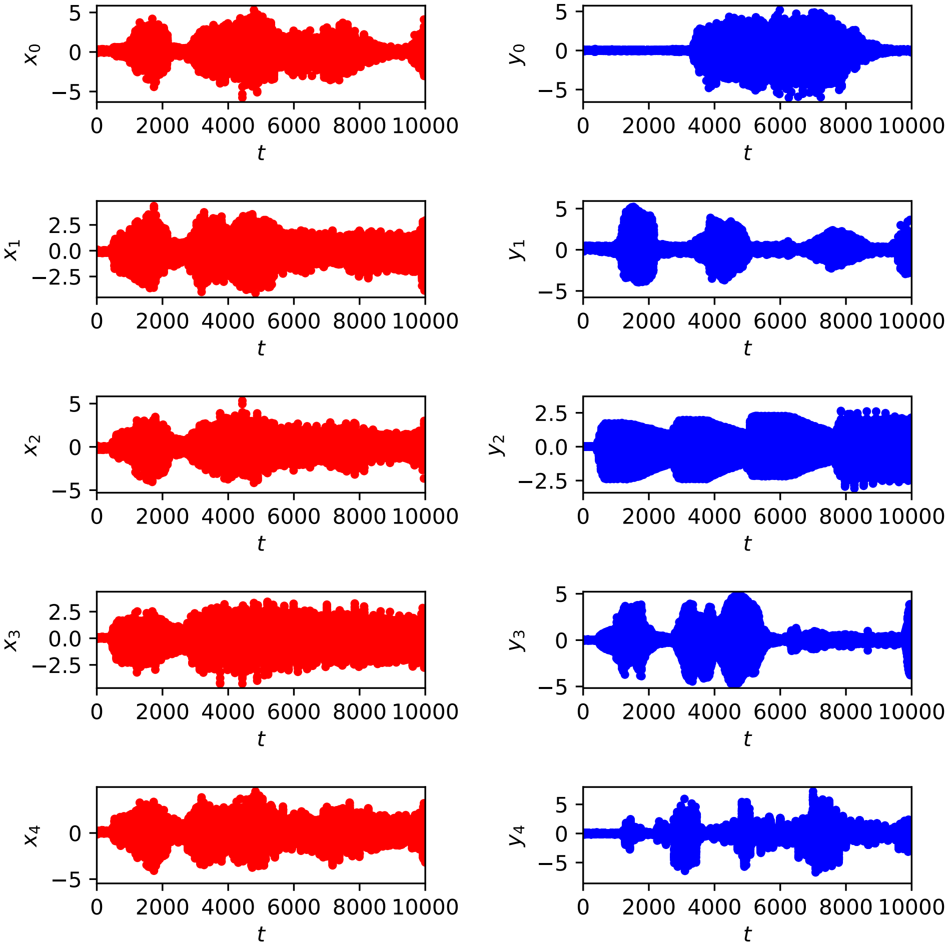

FIG. 4. shows the ICA results of separating the five mixed sound waves in the left panels (red). The separated independent components are shown in the right panels (blue). When the separated signals are played, one can clearly recognize these meaningful audio signals.

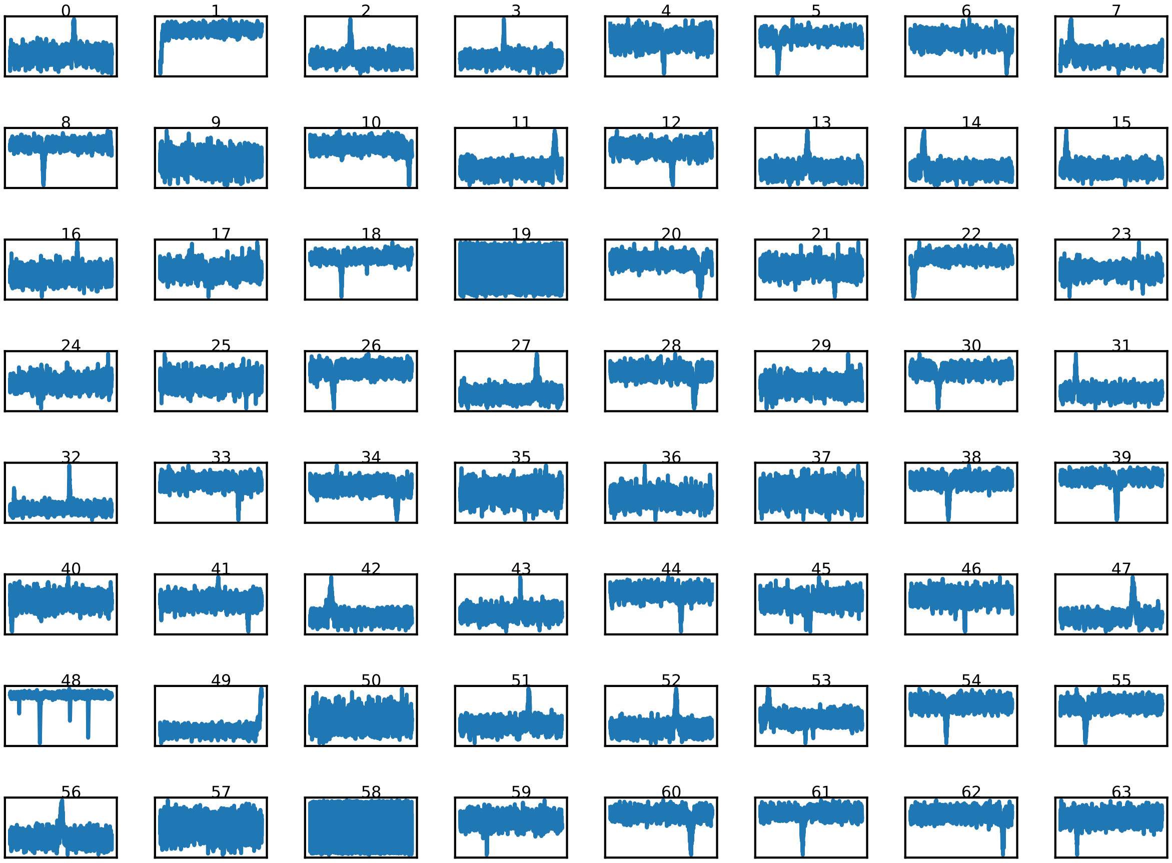

FIG. 5. All 64 independent components separated by the moment-based algorithm.

in choosing the correct and robust objective function. Several objective functions have been proposed to improve the robustness and computation efficiency. However, if we know in advance what the probability distributions look like for the sources or latent variables, it is always the best to tailor the objective function to include this information. Nonetheless, ICA has already shown reliable performance in examples such as sound waves identification and EEG signal separation using some default robust objective functions. The ongoing research in this field include cases where the number of detectors is smaller than the number of sources and non-linear mixing problems.

[1] T.-W. Lee, Independent component analysis, in *Independent component analysis* (Springer, 1998) pp. 27–66.

[2] A. Ng, Independent component analysis, in *CS229 Lecture Notes* (2020).

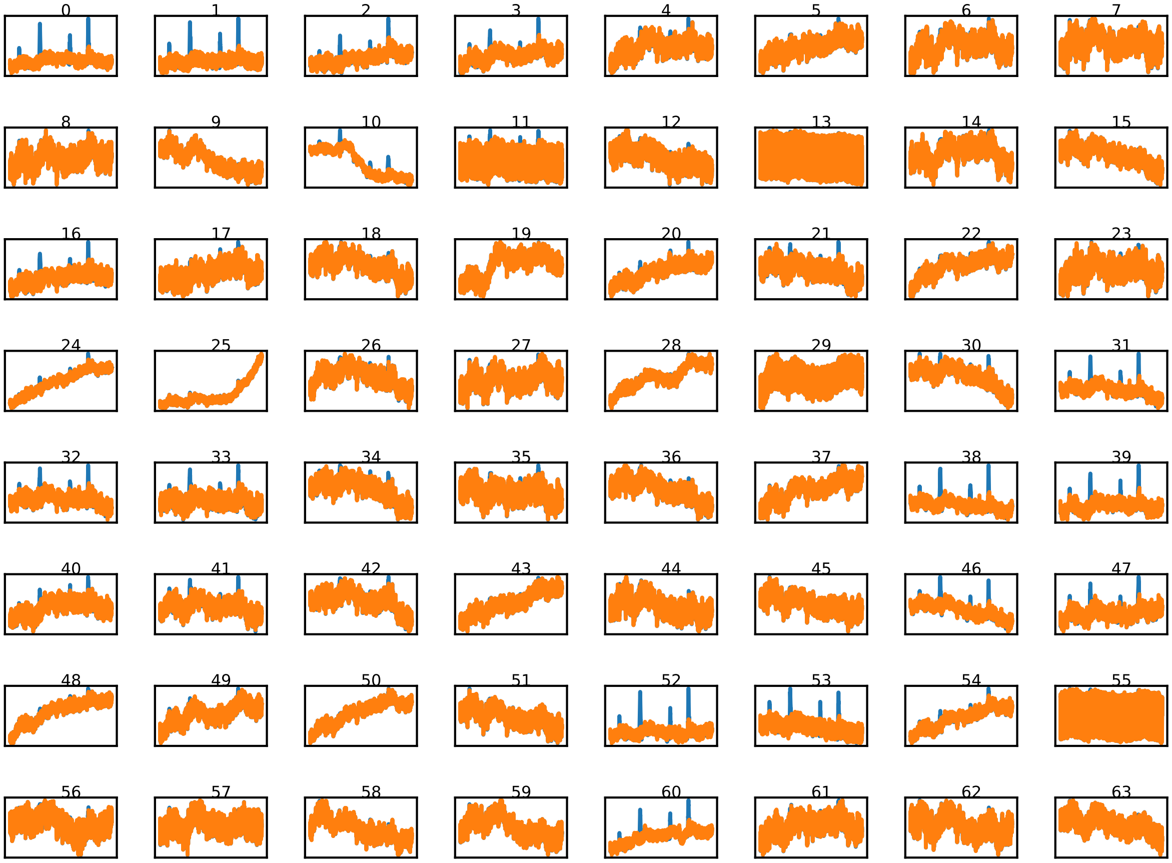

FIG. 6. Original signals (blue) and reconstructed signals with the blinking component removed (orange).

[3] J. Miettinen, S. Taskinen, K. Nordhausen, and H. Oja, Fourth moments and independent component analysis, Statistical science **30**, 372 (2015).

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

[5] S. Talebi, Independent component analysis (2021).