Independent Component Analysis

Kittithat Krongchon[1] and Kannan Lu[1]

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

(Dated: May 2, 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 EEG 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'. Or in other words, data are composed by 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 will 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 formalisms, we first define the notations that we will use throughout the whole document. We will use boldface capital letters $\mathbf{A}$ for matrices, boldface lowercase letters $\mathbf{a}$ for vectors and ordinary lower case letters a for scalars. We will use subscripts for denoting 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 $k = 1, 2, ..., m$. Following these conventions,

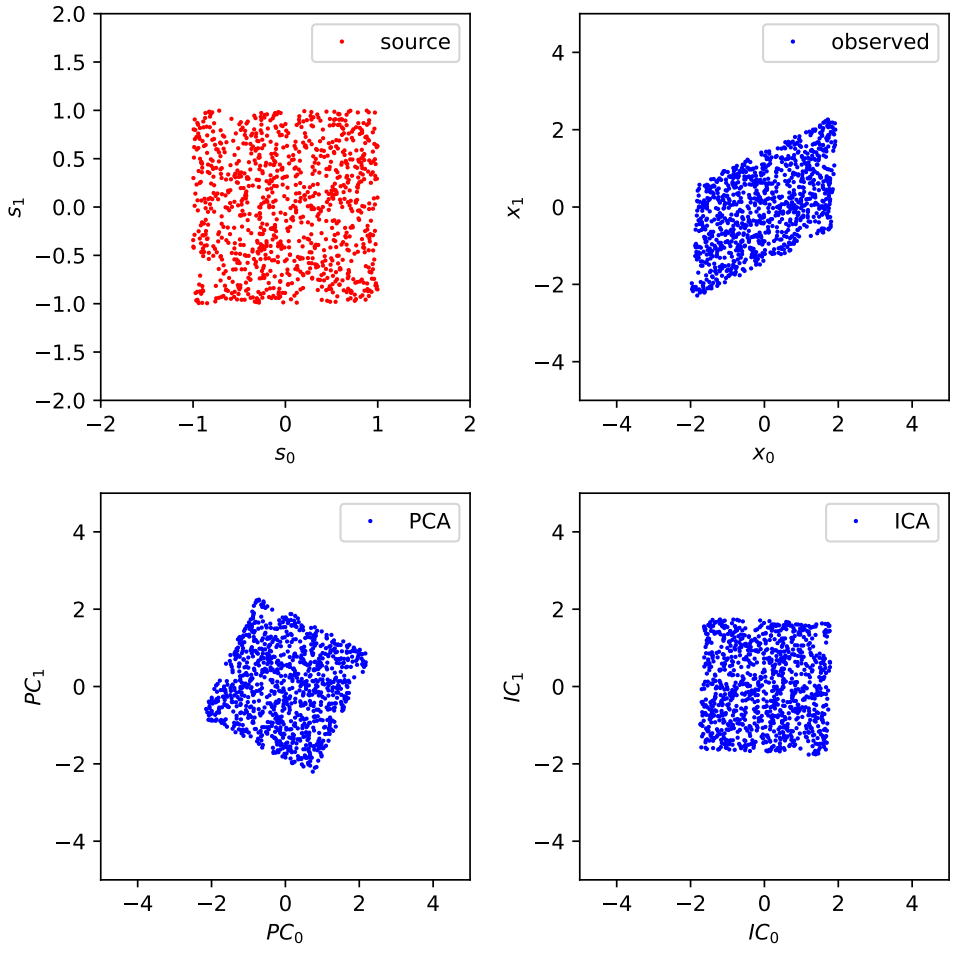

FIG. 1. shows the PCA and ICA result of unmixing the observed data x_i generated from two independent uniform variables s_i. The algorithm is based on maximizing the higher order moments.

we denote the original signal to be $\mathbf{s}^{(i)} \in \mathbb{R}^{\mathrm{d}}$. We consider the standard setting where the d dimensional signals $\mathbf{s}^{(i)} \in \mathbb{R}^{\mathrm{d}}$ (d original sources) are mixed and observed by d detectors $\mathbf{x}^{(i)} \in \mathbb{R}^{\mathrm{d}}$ (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 $\mathbf{A} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ 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 un-mixing 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 seperating 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 [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 condition (1) 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 origianl signals are all Gaussians, they cannot be learned through the un-mixing. 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 resulting in the same observed data $\mathbf{x}= \frac{1}{c}\mathbf{A}c\mathbf{s}$. This scaling ambiguity can be furtherly extended with regard to each component of the original signal. That is, for a particular component j, if we scale the component by c_j the corrsponding 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]. Or to be more precise, in order to separate the independent components, we require that the original signals can only have at most 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 $j = 1, 2, ..., d$. Then given the mixing matrix $\mathbf{A}$, we have

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

and

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

where $\mathbf{M}$ and $\boldsymbol{\Sigma}$ are diagonal matrices of $\mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ with diagonal elements to be $\{\mu_j^2\}$ and $\{\sigma_j^2\}$, respectively. Due to the scale 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,

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

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 $\tilde{\mathbf{A}}\mathbf{R}$. Unpon this rotation, we would observe $\mathbf{x}'$ as $\mathbf{x}' = \tilde{\mathbf{A}}\mathbf{R}\tilde{\mathbf{s}}$. $\mathbf{x}'$ is again normally distributed and has covariance matrix,

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

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 seperate 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 will not affect the seperated 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 $\tilde{\mathbf{x}} = \boldsymbol{\Sigma}^{-1/2}(\mathbf{x} - \mu)$ (whitening). Then, to recover the source signal, we need to find $\mathbf{W}$ with orthonormal rows $\{\mathbf{w}_i^t\}$ such that $\mathbf{s} = \mathbf{W}\boldsymbol{\Sigma}^{-1/2}\mathbf{x}$.

Higher order statistics approach

The non-Gaussianity thus directly provide 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 distibutions can be factorized into marginal distributions by investigating into higher-order moments. There are multiple moment-based objective functions that have been used along the development of ICA. We will define some of the notations and discuss one of the moment-based objective functions in detail.

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

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

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

$$|\mathbb{E}[(\mathbf{w}^t\mathbf{x})^4] - 3| \leq max\{|\mathbb{E}[x_1^4] - 3|, ..., |\mathbb{E}[x_d^4] - 3|\} \tag{5}$$

$$|\mathbb{E}[(\mathbf{w}_1^t\mathbf{x})^4 - 3]| + ... + |\mathbb{E}[(\mathbf{w}_d^t\mathbf{x})^4 - 3]| \leq |\mathbb{E}[x_1^4] - 3| + ... + |\mathbb{E}[x_d^4] - 3| \tag{6}$$

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],

$$\frac{\partial|\kappa(\mathbf{w}^t\mathbf{x})|}{\partial\mathbf{w}} = 4sign(\kappa(\mathbf{w}^t\mathbf{x}))(\mathbb{E}[\mathbf{x}(\mathbf{w}^t\mathbf{x}))^3]) - 3\mathbf{w}||\mathbf{w}||^2). \tag{7}$$

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 ideas (to be covered later) such as

$$G_1(u) = \frac{1}{a_1}\log(\cosh a_1 u) \tag{8}$$

$$G_2(u) = -\exp(-u^2/2), \tag{9}$$

where $u = \mathbf{w}^t\mathbf{x}$ [4]. We will use G_1 for our implementation of one of the algorithms in later sections.

Information theoretic approach

Apart from the moment based approaches, there are several information theoetic formalisms. Here 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 subsequant setion.

The centralized quantity in all these formalisms is the mutual information, which is defined as the KL divergence of the multivariate distribution of the observed data $\mathbf{x}$ and product of all its marginal distributions, i.e.

$$I(\mathbf{x}) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_i p_i(x_i)}. \tag{10}$$

$I(\mathbf{x})$ is non-negative and is 0 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 seperated components $\mathbf{u} = \mathbf{W}\mathbf{x}$.

One of them is a key finding in neural networks, that is, maximizing information between inputs $\mathbf{x}$ and outputs $\mathbf{y}$ implies that the output distributions are factorized, thus minimizes 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 $I(\mathbf{y}) = \sum_{\mathbf{y}} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_i p_i(y_i)}$ 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 network, the output is given by the nonlinearity $y_i = g_i(\mathbf{w}_i^t \mathbf{x}) := g_i(u_i)$, thus $p(y_i) = \frac{p(u_i)}{|\frac{\partial g_i}{\partial u_i}|}$. The maximum is therefore obtained by considering the gradient,

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

This implies that the nonlinear function g_i in the neural network $y_i = g_i(\mathbf{w}_i^t \mathbf{x})$ 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

$$\sum_i J(u_i) = \sum_i D(p(u_i)||p_G(u_i)) \tag{12}$$

$$= \sum_i p(u_i) \log \frac{p(u_i)}{p_G(u_i)} \tag{13}$$

$$= \sum_{\mathbf{u}} p(\mathbf{u}) \log \frac{\prod_i p(u_i)}{\prod_i p_G(u_i)} \tag{14}$$

$$= \sum_{\mathbf{u}} p(\mathbf{u}) \log \frac{\prod_i p(u_i)}{p_G(\mathbf{u}_i)} \tag{15}$$

$$= \sum_{\mathbf{u}} p(\mathbf{u}) \log \frac{\prod_i p(u_i)}{p(\mathbf{u})} + \sum_{\mathbf{u}} p(\mathbf{u}) \log \frac{p(\mathbf{u})}{p_G(\mathbf{u})} \tag{16}$$

$$= D(\prod_i p(u_i)||p(\mathbf{u})) + J(\mathbf{u}) \tag{17}$$

$$= -I(\mathbf{u}) + J(\mathbf{u}) \tag{18}$$

$$= -I(\mathbf{u}) - H(\mathbf{u}) - \sum_{\mathbf{u}} p(\mathbf{u}) \log p_G(\mathbf{u}) \tag{19}$$

$$= -I(\mathbf{u}) - H(\mathbf{x}) - \log(|det(\mathbf{W})|) - \frac{1}{2} \log((2\pi e)^d det(< \mathbf{u}, \mathbf{u}^t >)) \tag{20}$$

$$= -I(\mathbf{u}) - H(\mathbf{x}) - \frac{1}{2} \log((2\pi e)^d). \tag{21}$$

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 $\hat{p}_s(\mathbf{w}_i\mathbf{x})$ satisfying $p(\mathbf{x}) = \prod_{i=1}^d \hat{p}_s(\mathbf{w}_i\mathbf{x})|\mathbf{W}|$ [2]. The log-likelihood is

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

If the approximation $\hat{p}_s(\mathbf{w}_i\mathbf{x})$ are close to the actual pdf, the first term is approximately $-\sum_i H(\mathbf{w}_i\mathbf{x})$ 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. We will proceed to discuss the algorithm associated with the information based approach in the next section.

ALGORITHMS

In this section, we will discuss some of the common algorithms based on the previous mathematical formalisms. We will 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 will 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

$$J(Y) \propto \{E[G(Y)] - E[G(V)]\}^2, \tag{23}$$

where Y is a random variable of zero mean and unit variance, which can be achieved by whitening the data, and V is a Gaussian variable also of zero mean and unit variance. In function forms of G_1 and G_2 should not grow too fast to be robust. The following choice of G is proposed by Hyvrinen and Oja [4].

$$G_1(u) = \frac{1}{a_1} \log(\cosh a_1 u). \tag{24}$$

$$G_1'(u) = \tanh(a_1 u). \tag{25}$$

We want to find the extrema of $E[G(Y)]$ to maximize $J(Y)$ under the constraints

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

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

$$E[xg(w^T x)] - \beta w = 0. \tag{27}$$

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

$$w_{n+1} = w_n - \frac{F(w_n)}{F'(w_n)} \tag{28}$$

$$= w_n - \frac{E[xg(w_n^T x)] - \beta w_n}{E[g'(w_n^T x)] - \beta} \tag{29}$$

$$= \frac{w_n E[g'(w_n^T x)] - \beta w_n - E[xg(w_n^T x)] + \beta w_n}{E[g'(w_n^T x)] - \beta} \tag{30}$$

$$= \frac{w_n E[g'(w_n^T x)] - E[xg(w_n^T x)]}{E[g'(w_n^T x)] - \beta}. \tag{31}$$

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 $g(s_i) = \frac{1}{1+e^{-s_s}}$, 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],

$$l(\mathbf{W}) = \sum_{j=1}^{N}(\sum_{i=1}^{d} \log(\hat{p}_s(\mathbf{w}_i^t \mathbf{x}^{(j)})) + \log|\mathbf{W}|) \tag{32}$$

$$= \sum_{j=1}^{N}(\sum_{i=1}^{d} \log(g'(\mathbf{w}_i^t \mathbf{x}^{(j)})) + \log|\mathbf{W}|), \tag{33}$$

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

$$\frac{\partial l(\mathbf{W})}{\partial \mathbf{W}} = \sum_{j=1}^{N}(\mathbf{q}^{(j)} \mathbf{x}^{t,(j)} + (\mathbf{W}^t)^{-1}), \tag{34}$$

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

$$\mathbf{W} = \mathbf{W} + \alpha(\mathbf{q}^{(j)} \mathbf{x}^{t,(j)} + (\mathbf{W}^t)^{-1}). \tag{35}$$

This means that we update the $\mathbf{W}$ for each sample of the observed data instead of calculating the exact summation for all samples at each step of iteration. This iteration schemes is

accompanied by randomize the ordering of samples initially for each iteration (making the iteration more stochastic).

APPLICATIONS

Sklearn example

We implement the moment-based approach according to the formalism described in the previous section. The algorithm is applied to a following set of signal functions (example from sklearn module):

$$s_1 = \sin(2t) \tag{36}$$

$$s_2 = \text{sign}[\sin(3t)] \tag{37}$$

$$s_3 = \frac{1}{2\pi}(t \bmod 2\pi). \tag{38}$$

The mixing matrix is taken to be

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0.5 & 2 & 1 \\ 1.5 & 1 & 2 \end{bmatrix}. \tag{39}$$

The result from our implementation and sklearn module are shown in Fig. **??**.

Sound waves

We employed the maximum likelihood algorithm to separate out the independent components of five mixed sound tracks (data obtained from Standford cs229 online course materials). The mixed sound waves are shown in the Fig. 2 left panels (red). The separated independent sound tracks are shown in the right panels (blue) in Fig. 2. The separated sound tracks can be played and listened one by one to verify that they are meaningful.

Blinking removal in EEG signal

DISCUSSIONS AND CONCLUSIONS

In this review paper of ICA, we have discussed the traditional and basic notions of this technique. We have also implemented the algorithms and utilized our codes to solve several different examples. We have noticed 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 in choosing the correct and robust objective function. Several objective functions have been proposed to improve the robusteness and computation

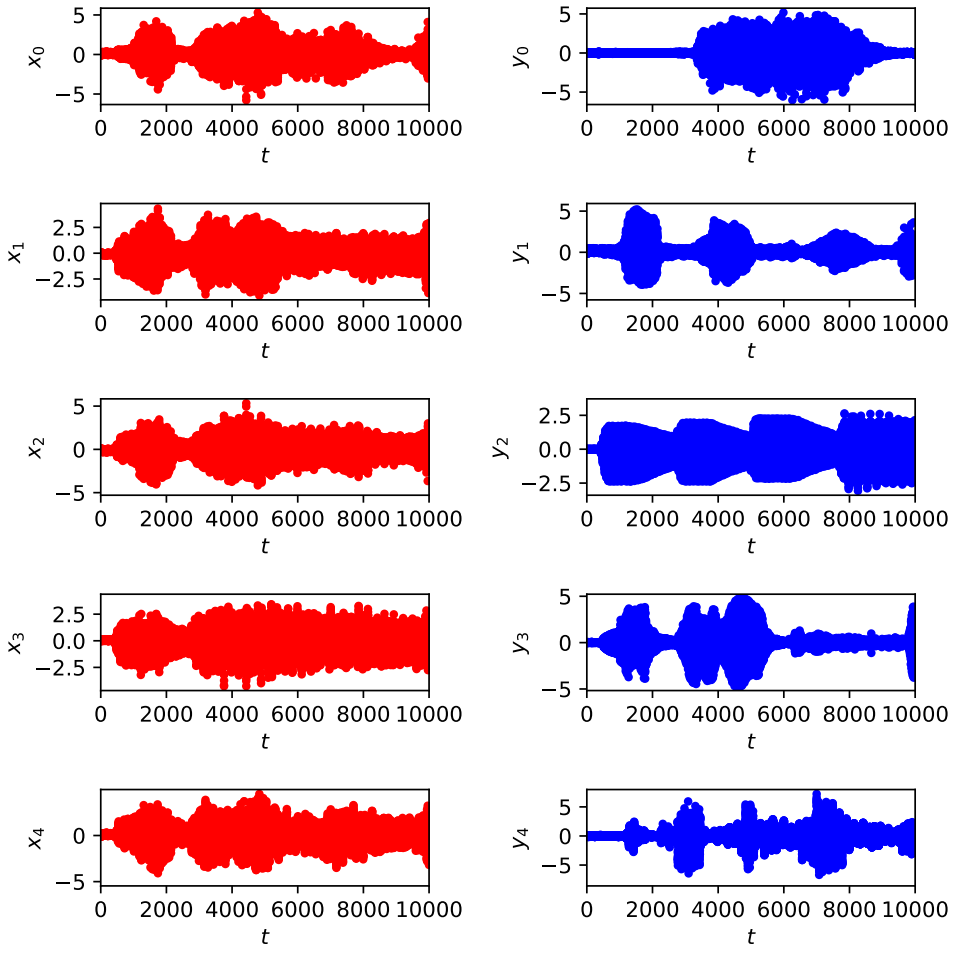

FIG. 2. shows the ICA results of separating the five mixed sound waves in the left panels (red). The separated independent components are shown in the right panels (blue). When the separated signals are played, one can clearly recognize these meaningful audio signals.

efficiency. However, if we know in advance what the probability distributions look like for the sources (latent variables), it would always be 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 signals separation using some default robust objective functions. The ongoing research in this field focuses on cases where the number of detectors 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).

[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).