Independent Component Analysis*

Xing Yi Liu, Weiyu Huang, Landon Choi[1]

***Abstract*— Independent component analysis (ICA) is a method of dimension reduction used to separate data into independent vectors. The method attempts to decipher the latent variables which make up a signal, thus having useful applications in signal processing and voice recognition. We derive the mathematical theory behind ICA and demonstrate its effectiveness using datasets.**

I. INTRODUCTION

In recent years, signal processing and voice recognition have become an increasingly important area of machine learning. Real-life applications of ICA include Siri, which can differentiate people's voices from background noise, and music recognition, which can identify a song in a short time. In this project, we experiment with more extensive applications, such as separating instruments playing music.

Moreover, this study derives the mathematical theory behind independent component analysis (ICA), preprocessing steps necessary to make ICA more successful, and the derivation of various ICA estimation algorithms.

In Section II, we illustrate the mathematical formulation of ICA, and delve into independence and non-Gaussianity assumptions of the framework. We proceed to preprocessing, or whitening that simplifies the ICA problem and increases its success rate, followed by deriving maximum likelihood estimation and the FastICA algorithm. In Section III, we apply FastICA to real-life recordings mixed with different signals, with the goal of separating them. Finally, in Section IV, we discuss some conclusions and briefly suggest future directions.

II. MATHEMATICAL FOUNDATIONS

This section will examine the mathematical basis of the ICA problem and different methods used to solve it. Through our understanding of the mathematical basis, we will uncover the assumptions that lay the groundwork for the solutions and the potential shortcomings ICA brings.

A. Mathematical Formulation

The mathematical formulation behind ICA is closely related to a real-life situation. Suppose there are m microphones recording m different sources emitted simultaneously. Hence, the overall recordings are of a mix of sounds. We let the data set be represented by $\{\boldsymbol{x}_i\}_{i=1}^N$. Each $\boldsymbol{x_i}$ is defined by the observation of the signal from m microphones, or devices which capture the data. Embedded within each recording is the m latent independent sources mixed together. Let the latent independent sources be $\boldsymbol{s}_i = [s_{i1}, s_{i2}, ..., s_{im}]^\intercal$. If we assume that the observation of each microphone is a linear combination of the latent sources, then we have

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{pmatrix} = \begin{pmatrix} a_{11}s_{i1} + a_{12}s_{i2} + ... + a_{1m}s_{im} \\ a_{21}s_{i1} + a_{22}s_{i2} + ... + a_{2m}s_{im} \\ \vdots \\ a_{m1}s_{i1} + a_{m2}s_{i2} + ... + a_{mm}s_{im} \end{pmatrix}.$$

In matrix notation, this can be expressed as

$$\boldsymbol{x}_i = \mathbf{A}\boldsymbol{s}_i,$$

where

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix}$$

is the mixing matrix. Each element a_{ij} is the respective weight of the j^{th} source in the i^{th} observed (microphone) signal.

The goal of ICA is to estimate the matrix $\mathbf{A}$ of parameters, which transforms the independent sources to the observed signal. By taking the inverse of $\mathbf{A}$, we obtain the matrix $\mathbf{W} = \mathbf{A}^{-1}$ that recovers the sources from the observed signal, referred to as the unmixing matrix, and the best approximation of the latent sources $\boldsymbol{s}_i$, denoted as $\boldsymbol{y}_i$ through

$$\boldsymbol{y}_i = \mathbf{W}\boldsymbol{x}_i \approx \boldsymbol{s}_i$$

[2].

B. Assumptions

One of the main assumptions of ICA is the statistical independence of the sources or "latent variables", meaning that the joint probabilities of all sources is equal to the product of the marginal probabilities of each source [1]:

$$p(\boldsymbol{s}_i) = \prod_{j=1}^{M} p(s_{ij}).$$

Another assumption of ICA is the non-Gaussian distribution of the sources. Since the Central Limit Theorem states that any linear combination of random variables of any distribution will have a Gaussian distribution, the observations will all have a Gaussian distribution. This means that any variable will be more Gaussian with each variable addition. The essence and methodology of ICA is to "reverse" the Central Limit Theorem and uncover the independent components of

*This work was not supported by any organization.

[1]X. Liu, W. Huang, and L. Choi are with the Department of Mathematics, University of California, Los Angeles, 520 Portola Plaza, Los Angeles, CA 90095, USA lxy@ucla.edu, ingridweiyu@ucla.edu, landon622@ucla.edu

a linear combination, by maximizing the non-Gaussianity of its sources. However, if the sources are Gaussian, then maximizing non-Gaussianity will be futile, and the method of ICA will not work [3].

C. Preprocessing

A commonly used preprocessing step is to center the data so that its mean equals to 0. Suppose $\bar{\boldsymbol{x}} = \mathbb{E}(\boldsymbol{x}_i)$. Then, let

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}},$$

where $\boldsymbol{x} \in \{\boldsymbol{x}_i\}_{i=1}^N$. The subscript is omitted briefly to simplify notation, but note that the following preprocessing steps should be applied to $\boldsymbol{x}_i$ for all $i = 1, \ldots, N$.

The unmixing matrix $\mathbf{W}$ remains the same after the centering. We will obtain the unmixing matrix $\boldsymbol{W}$ through the centered data $\boldsymbol{x_c}$. Therefore, to obtain the estimates of the independent latent sources, we need to add back $\bar{\boldsymbol{x}}$ to the centered data $\widetilde{\boldsymbol{x}}$. Our estimates for the separated latent variables will therefore be

$$\boldsymbol{y} = \mathbf{W}(\widetilde{\boldsymbol{x}} + \bar{\boldsymbol{x}}).$$

Next, preprocessing involves whitening the observed mixed signal $\boldsymbol{x}$, which transforms $\boldsymbol{x}$ into having independent components of unit variance. Since the whitened observation $\boldsymbol{x}_w$ has mean zero and unit variance, its covariance matrix satisfies

$$\mathbb{E}(\boldsymbol{x}_w \boldsymbol{x}_w^\intercal) = \mathbf{I}.$$

To obtain $\boldsymbol{x}_w$, we perform the following steps. The covariance matrix of the centered (mean zero) mixed signal $\widetilde{\boldsymbol{x}}$ is $\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^\intercal)$, an $m \times m$ square matrix on which we can perform the eigen-decomposition

$$\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^\intercal) = \mathbf{V}\mathbf{D}\mathbf{V}^{-1},$$

where $\mathbf{V}$ is a matrix whose columns are eigenvectors of $\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^\intercal)$, and $\mathbf{D}$ is a diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_m$. The eigenvalues are arranged in decreasing order down the diagonal, with the eigenvalue in the (i, i)-th entry of $\mathbf{D}$ corresponding to its eigenvector in the i-th column of $\mathbf{V}$.

Since $\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^\intercal)$ is a symmetric matrix, it has orthogonal eigenvectors. This means that $\mathbf{V}^\intercal\mathbf{V} = \mathbf{I}$, which is equivalent to $\mathbf{V}^{-1} = \mathbf{V}^\intercal$. The eigenvalue decomposition of $\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^\intercal)$ can thus be written as

$$\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^\intercal) = \mathbf{V}\mathbf{D}\mathbf{V}^\intercal,$$

which can be useful for reducing the computation cost of algorithms. Finally, the whitened signal can be obtained by

$$\boldsymbol{x}_w = \mathbf{V}\mathbf{D}^{-\frac{1}{2}}\mathbf{V}^\intercal\widetilde{\boldsymbol{x}} = \mathbf{A}_w\boldsymbol{s},$$

where the $-\frac{1}{2}$ exponent of $\mathbf{D}$ is applied element-wise on each diagonal entry, and $\mathbf{A}_w = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^\intercal\mathbf{A}$.

Note that in ICA, estimation of $\mathbf{W}$ and $\boldsymbol{s}$ may vary by the factor of a constant. To illustrate qualitatively, since $\boldsymbol{s} = \mathbf{W}\boldsymbol{x}$, $\boldsymbol{s}$ and $\mathbf{W}$ may be freely scaled up or down depending on one another. Thus, in addition to assuming the independence of sources, we can assume without loss of generality that

$$\mathbb{E}(\boldsymbol{s}\boldsymbol{s}^\intercal) = \mathbf{I}.$$

Then,

$$\begin{aligned}\mathbb{E}(\boldsymbol{x}_w\boldsymbol{x}_w^\intercal) &= \mathbb{E}(\mathbf{A}_w\boldsymbol{s}\boldsymbol{s}^\intercal{\mathbf{A}_w}^\intercal) \\ &= \mathbf{A}_w\mathbb{E}(\boldsymbol{s}\boldsymbol{s}^\intercal)\mathbf{A}_w^\intercal \\ &= \mathbf{A}_w{\mathbf{A}_w}^\intercal.\end{aligned}$$

We assumed that the whitened observed mixed signal $\boldsymbol{x}_w$ satisfies $\mathbb{E}(\boldsymbol{x}_w\boldsymbol{x}_w^\intercal) = \mathbf{I}$, so

$$\mathbf{A}_w{\mathbf{A}_w}^\intercal = \mathbf{I}.$$

Thus, $\mathbf{A}_w = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^\intercal\mathbf{A}$ is an orthogonal matrix. Note that the original mixing matrix $\mathbf{A}$ had m^2 parameters, which is significantly more than the $\frac{m(m-1)}{2}$ degrees of freedom of the orthogonal matrix $\mathbf{A}_w$. Whitening has therefore dramatically reduced the complexity of the ICA problem [4].

D. Maximum Likelihood Estimation

A popular method for solving ICA is by maximizing the likelihood function with respect to the parameters, which in this case is $\mathbf{A}$. In that way, using $\mathbf{A}_{\mathbf{ML}}$ will give the highest likelihood of those observations occurring. Since $\boldsymbol{x}_i = \mathbf{A}\boldsymbol{s}_i$, we can derive the probability of the observations from the probability of the sources by using the popular definition for joint probability distributions under matrix transformations.

$$\begin{aligned}p(\boldsymbol{x}_i) &= \frac{1}{|\det(\mathbf{A})|}p(\boldsymbol{s}_i) \\ &= \frac{1}{|\det(\mathbf{A})|}p(\boldsymbol{s}_i)\end{aligned}$$

Since $\mathbf{W} = \mathbf{A}^{-1}$,

$$\begin{aligned}p(\boldsymbol{x}_i) &= \left|\frac{1}{\det(\mathbf{A})}\right| p(\boldsymbol{s}_i) \\ &= |\det(\mathbf{W})|\, p(\boldsymbol{s}_i) \\ &= |\det(\mathbf{W})| \prod_{j=1}^{m} p_i(s_{ij}) \\ &= |\det(\mathbf{W})| \prod_{j=1}^{m} p_i(\boldsymbol{w}_j^\intercal\boldsymbol{x}_i),\end{aligned}$$

where $\boldsymbol{w}_j^\intercal$ is the j-th row of $\mathbf{W}$ (a row vector).

By deriving the joint probability of the observations, we can derive the likelihood function for the observations. We assume that the observations are independently and identically distributed, so we take the product of $p(\boldsymbol{x}_i)$ over $i = 1, 2, \ldots, N$.

$$L(\mathbf{W}) = \prod_{i=1}^{N}\left(|\det(\mathbf{W})| \prod_{j=1}^{m} p_i(\boldsymbol{w}_j^\intercal\boldsymbol{x}_i)\right)$$

$$L(\mathbf{W}) = |\det(\mathbf{W})|^N \prod_{i=1}^{N}\prod_{j=1}^{m} p_i(\boldsymbol{w}_j^\intercal\boldsymbol{x}_i)$$

As with other maximum likelihood methods, it is easier to optimize the log-likelihood.

$$\log\left(L(\mathbf{W})\right) = N\log|\det(\mathbf{W})| + \sum_{i=1}^{N}\sum_{j=1}^{m}\log\left(p_i(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)\right)$$

$$\frac{1}{N}\log\left(L(\mathbf{W})\right) = \log|\det(\mathbf{W})| + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\log\left(p_i(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)\right)$$

Maximizing $\frac{1}{N}\log\left(L(\mathbf{W})\right)$ is equivalent to maximizing the log-likelihood, so we proceed to take its derivative with respect to $\mathbf{W}$.

$$\frac{\partial}{\partial \mathbf{W}}\frac{1}{N}\log\left(L\left(\mathbf{W}\right)\right) = \left(\mathbf{W}^T\right)^{-1} + \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{g}(\mathbf{W}\boldsymbol{x}_i)\boldsymbol{x}_i^\intercal,$$

where $\boldsymbol{g}$ is defined by

$$\boldsymbol{g}(\boldsymbol{a}) = \begin{pmatrix} g_1(a_1) \\ g_2(a_2) \\ \vdots \\ g_m(a_m) \end{pmatrix}.$$

The functions $g_i = (\log(p_i))' = \frac{p_i'}{p_i}, i = 1, \ldots, m$ are called score functions and depend on the probability distribution p_i of the source s_i. We can then apply an iterative process such as gradient descent to find the maximum likelihood estimator for $\mathbf{W}$. This formulation gives us the Bell-Sejnowski algorithm [2].

Algorithm 1 Bell-Sejnowski Algorithm with Preprocessing

Require: $\{\boldsymbol{x}_n\}_{n=1}^N, \{p_i(s_i)\}_{i=1}^m$

1: Calculate the sample mean $\bar{\boldsymbol{x}}$ of the data and center the data by $\widetilde{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_i - \bar{\boldsymbol{x}}$ for $i = 1, \ldots, N$.
2: Calculate $\mathbf{V}, \mathbf{D}$ such that $\mathbb{E}(\widetilde{\boldsymbol{x}}_i\widetilde{\boldsymbol{x}}_i^\intercal) = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ is the eigen-decomposition of $\mathbb{E}(\widetilde{\boldsymbol{x}}_i\widetilde{\boldsymbol{x}}_i^\intercal)$.
3: Preprocess the data by $\boldsymbol{x}_{wi} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^\intercal\widetilde{\boldsymbol{x}}_i$ for $i = 1, \ldots, N$.
4: Initialize $\mathbf{W} \in \mathbf{R}^{m\times m}$.
5: Update $\mathbf{W} \leftarrow \mathbf{W} - \eta\left(\left(\mathbf{W}^T\right)^{-1} + \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{g}(\mathbf{W}\boldsymbol{x}_{wi})\boldsymbol{x}_{wi}^\intercal\right)$, where $\eta > 0$
6: If stopping criterion is satisfied, proceed to step 7. Otherwise, return to step 5.
7: Estimates for the separated latent variables will be given by $\boldsymbol{y}_i = \mathbf{W}\boldsymbol{x}_i$.

E. Choosing a Source Distribution

Choosing a source distribution is difficult, because it varies based on the specific task at hand. In general, two principles should be followed. The distribution must be parametric and non-Gaussian [3]. The distribution must be parametric in that it must be explicitly defined by a finite number of parameters. As seen before, the distribution must also be non-Gaussian due to the nature of ICA's goal, but it is vague as to what a non-Gaussian distribution exactly is. In order to define a non-Gaussian distribution, a classic measure of non-Gaussianity, kurtosis, is used [3]. The kurtosis of a distribution of a random variable y is defined as

$$\kappa(y) = \mathbb{E}(y^4) - 3(\mathbb{E}(y^2))^2.$$

If we also assume that the distribution is centered and of unit variance, then the kurtosis can be simplified to the fourth moment.

$$\kappa(y) = \mathbb{E}(y^4) - 3$$

Gaussian distributions have a kurtosis of 0. Therefore, non-Gaussian distributions have a non-zero kurtosis, with more extreme values being more "non-Gaussian". Popular source distributions thus include $p_s = (1 - \tanh(s)^2)$ or $p_s = \frac{e^{-s}}{(1+e^{-s})^2}$, which have notably high kurtosis.

Although kurtosis is not the most robust measure of non-Gaussianity, and other measures are often used in place of it, its simplicity and elegancy prove useful in certain scenarios [3].

F. FastICA

Another measure of non-Gaussianity involves entropy, defined by

$$H(\boldsymbol{y}) = -\int f(\boldsymbol{y})\log f(\boldsymbol{y})d\boldsymbol{y}.$$

Since Gaussian distributions have the largest entropy of all random variable distributions, by comparing the entropy of a distribution to that of a Gaussian distribution with similar variance, we can see how "Gaussian" or "non-Gaussian" a distribution is. This naturally brings in the definition of negentropy:

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{\text{Gauss}}) - H(\boldsymbol{y}),$$

where $\boldsymbol{y}_{\text{Gauss}}$ follows a Gaussian distribution with the same covariance matrix as $\boldsymbol{y}$. However, the definition of negentropy is hard to work with at times, so an approximation is used [2]:

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

where G is a non-quadratic function, and ν is a Gaussian random variable with zero mean and unit variance. This definition of non-Gaussianity forms the basis for FastICA.

FastICA is a fixed point algorithm that intends to maximize non-Gaussianity by maximizing the approximation of negentropy stated above. But, instead of maximizing the full term, it only intends to maximize $\mathbb{E}\{G(y)\}$. Suppose $\boldsymbol{w}_i^\intercal$ is the i-th row of $\mathbf{W}$ and $\boldsymbol{x}$ is an observation. Under the Kuhn-Tucker conditions, the optimum of $\mathbb{E}\{G(\boldsymbol{w}_i^\intercal\boldsymbol{x})\}$, given that $\|\boldsymbol{w}_i\| = 1$ (This is due to $\boldsymbol{x}$ being whitened as $\mathbb{E}(\boldsymbol{w}_i^\intercal\boldsymbol{x}\boldsymbol{x}^\intercal\boldsymbol{w}_i) = \boldsymbol{w}_i^\intercal\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^\intercal)\boldsymbol{w}_i = \boldsymbol{w}_i^\intercal\boldsymbol{w}_i = \|\boldsymbol{w}_i\| = 1$), is at

$$\mathbb{E}\{\boldsymbol{x}g(\boldsymbol{w}_i^\intercal\boldsymbol{x})\} - \beta\boldsymbol{w}_i = 0,$$

where $g = G'$ and $\beta = E\{\boldsymbol{w}_{i,\text{opt}}^\intercal \mathbf{x} g(\boldsymbol{w}_{i,\text{opt}}^\intercal \boldsymbol{x})\}$ is a constant. By using Newton's Method, we can then find the root

$$\boldsymbol{w}_i^+ = \boldsymbol{w}_i - (\mathbb{J}(\boldsymbol{w}_i))^{-1}(\mathbb{E}\{\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta \boldsymbol{w}_i).$$

Newton's Method requires the computation of the Jacobian. So the Jacobian can be calculated as such

$$\begin{aligned}\mathbb{J}(\boldsymbol{w}_i) &= \frac{\partial}{\partial \boldsymbol{w}_i}[\mathbb{E}\{\boldsymbol{x} g(\boldsymbol{x}^\intercal \boldsymbol{x})\} - \beta \boldsymbol{w}_i] \\ &= \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\intercal g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta \mathbf{I}\end{aligned}$$

However, it is difficult to compute the inverse of the exact Jacobian, so we instead simplify the expression by making several approximations. Due to the data being whitened, it is reasonable to assume that $\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\intercal g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} \approx \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\intercal\}\mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\}$, due to the lack of correlation between data points. Also, whitening the data allows us to assume that $\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\intercal\}\mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} = \mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\}\mathbf{I}$, giving an approximate Jacobian of

$$\mathbb{J}(\boldsymbol{w}_i) = (\mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta)\mathbf{I}.$$

By plugging this in, we get the expression

$$\boldsymbol{w}_i^+ = \boldsymbol{w}_i - \left(\frac{\mathbb{E}\{\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta \boldsymbol{w}_i}{\mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta}\right).$$

In order to make this expression algebraically simpler, we can then multiply both sides by $\beta - \mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\}$ to get

$$\begin{aligned}(\beta - \mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\})\boldsymbol{w}_i^+ &= -(\mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta)\boldsymbol{w}_i + \\ &\quad (\mathbb{E}\{\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \beta \boldsymbol{w}_i) \\ &= \mathbb{E}\{\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\},\end{aligned}$$

which, after normalizing the result, forms the iterative part of the FastICA algorithm [2].

Algorithm 2 FastICA

Require: $\{\boldsymbol{x}_n\}_{n=1}^N \subset \mathbb{R}^D$

1: Pick a random vector for $\boldsymbol{w}_i$.
2: Compute $\boldsymbol{w}_i^+ = \mathbb{E}\{\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})\} - \mathbb{E}\{g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})\}$.
3: Compute $\boldsymbol{w}_i \leftarrow \frac{\boldsymbol{w}_i^+}{\|\boldsymbol{w}_i^+\|}$.
4: If $\boldsymbol{w}_i$ has not converged, then repeat step 2.

Note that computing the expectation of $\boldsymbol{x} g(\boldsymbol{w}_i^\intercal \boldsymbol{x})$ and $g'(\boldsymbol{w}_i^\intercal \boldsymbol{x})$ is difficult, so it is often easier to compute the sample mean for all the observations $\boldsymbol{x}_i, i = 1, \ldots, N$, within the dataset.

III. EXPERIMENTS

In this section, we apply ICA to both synthetic data and real audio data.

A. Synthetic Data

For our synthetic dataset, we generated two sets of values from the uniform distribution which range from 0 to 1. We note these as $\{x_{1i}\}_{i=1}^N$ and $\{x_{2i}\}_{i=1}^N$, and we then apply an arbitrary 2×2 mixing matrix, $\mathbf{A}$, such that two new mixed sets of data $\{t_{1i}\}_{i=1}^N$ and $\{t_{2i}\}_{i=1}^N$ are obtained. That is,

$$\begin{pmatrix} t_{1i} \\ t_{2i} \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix}.$$

Our goal is to then identify the source distribution and the mixing matrix from the mixed sets of data using ICA.

First, we apply prepossessing by centering the mixed data and whitening.

Notice how in both Figure 1 and Figure 2, the independent sources can clearly be seen along the edges of the parallelogram. We can then find a transformation/rotation that will cause the parallelogram to appear as non-Gaussian as possible. For reference, a multivariate Gaussian distribution centered at zero with unit covariance is depicted in Figure 3. Note that any rotation applied to the Gaussian distribution will still bring about a Gaussian distribution, due to it appearing circular.

Using the FastICA algortihm, we can then find an unmixing matrix that will maximize the non-Gaussianity of the sources. From the unmixing matrix we can then get back the sources and compare them to the synthetic data set we created.

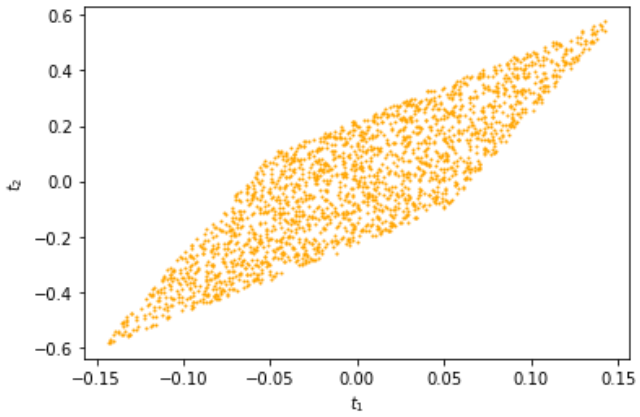

Fig. 1: Plotted data after centering the mixed data

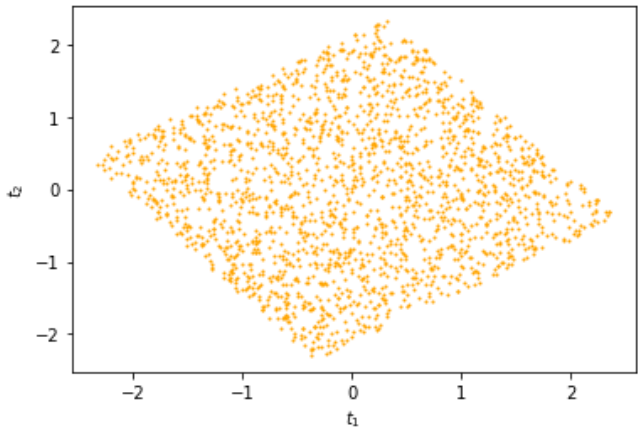

Fig. 2: Plotted data after whitening the mixed data

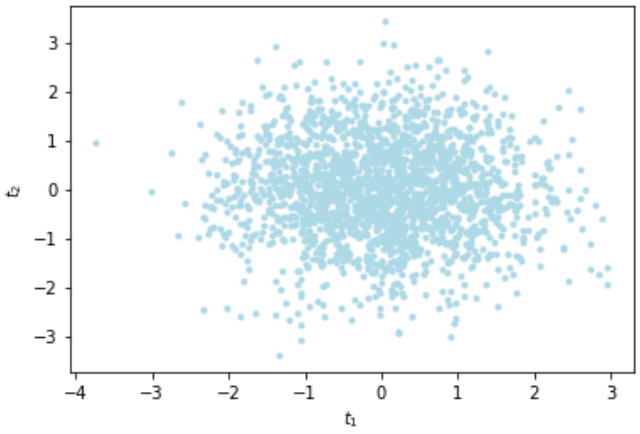

Fig. 3: Plotted data from a multivariate Gaussian distribution

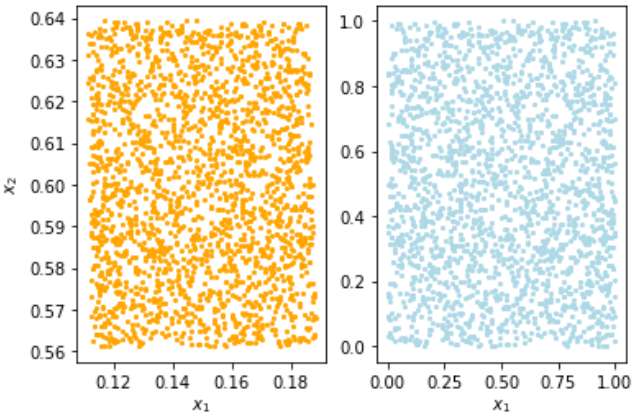

Fig. 4: Source distribution recovered vs true source distribution

Notice that while ICA can restore the shape of the distribution, it cannot restore the scaling of the distribution. This is because the ICA problem uncovers two variables, the unmixing matrix and the sources, from one variable, the observation. For any constant α,

$$\begin{aligned} \boldsymbol{x} &= \mathbf{A}\boldsymbol{s} \\ &= \left(\frac{1}{\alpha}\mathbf{A}\right)(\alpha \boldsymbol{s}). \end{aligned}$$

Therefore, while ICA cannot restore the exact source distribution, it can theoretically restore the shape of the distribution as long as it is non-Gaussian.

B. Musical Instruments Mix

Our data of mixed musical instruments was obtained from Kaggle's "ICA - The Musical Way" Python notebook [5]. The dataset contained three `.wav` files of a mixed sound signal, composed of three different musical sources. The sources were a piano, a cello, and a constant ring tone. The observations of each of the three `.wav` files are plotted in Figure 5.

In each `.wav` file, there were $N = 264515$ data points (scalars), indicating the channel value for the mixed sound.

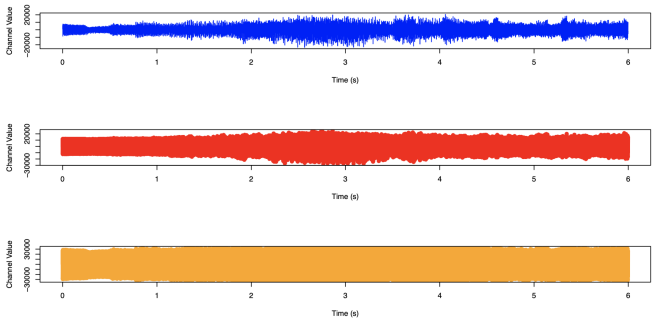

Fig. 5: Plot of mixed sounds in each `.wav` file

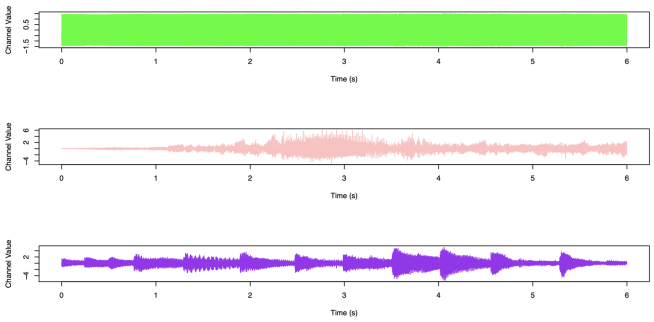

Fig. 6: Plot of extracted sound sources

The sampling rate was 44100 Hz, which means that each audio file lasted for $\frac{264515}{44100} \approx 6$ seconds. The x-axis of the plots represents time, hence the range from 0 to 6 seconds. The y-axis represents the channel value.

For all three files containing the mixed sound, the plot shows no discernible structure to the progression of channel values. This is as expected, since the three sound sources are independently mixed together, creating a cacophony of noises embedded in the files. The third plot in Figure 5 appears consistent, because its corresponding file was highly dominated by the constant ring tone sound, even though the other two sources were audibly present.

Next, in Python, we applied both the previously stated Bell-Sejnowski Algorithm with Preprocessing, as well as FastICA. The Bell-Seknowski Algorithm is not commonly used in practice due to its slow convergence, and such was precisely the case for this dataset. The estimate for $\mathbf{W}$ oscillated between two regions with very little change between iterations, and the Bell-Sejnowski Algorithm failed to convergence within a reasonable amount of time.

On the other hand, FastICA yielded an estimate for $\mathbf{W}$ almost immediately. The plot of the resulting three extracted sound sources are shown in Figure 6. The constant ring tone was separated as an independent source and is depicted in the top plot of Figure 6 (green). The cello sound's channel values were also successfully separated and is depicted in pink. Lastly, the piano's sound is plotted in purple. Compared to the plot of mixed sounds in Figure 5, more structure can clearly be observed in the sound waves of Figure 6. The audio of the three extracted sound sources clearly muted the other two sources and clarified one. Notice also that the top

plot of the constant ring tone (green) appeared much more structured than even the bottom plot of 5.

In summary, FastICA was highly effective in separating the three instrumental sources that made up the mixed signal.

IV. CONCLUSIONS

This paper described the mathematical formulation of independent component analysis, including its primary assumptions, preprocessing steps, and maximum likelihood estimation of the unmixing matrix $\mathbf{W}$. The maximum likelihood estimation technique led to the Bell-Sejnowski algorithm, though this is not often used in practice due to its slow convergence speed.

Given the need to assume a source distribution when estimating the parameters, we outlined techniques like maximizing non-Gaussianity of the source distributions using measures of kurtosis and negentropy. The technique of maximizing negentropy directly led to the derivation of the FastICA algorithm, which is commonly used in practice. Its convergence speed far exceeded that of the Bell-Sejnowski algorithm, and existing FastICA libraries in popular programming languages include preprocessing steps by default.

For experiments, we verified the effectiveness of ICA by generating synthetic data and successfully applying FastICA. We then conducted a practical experiment of separating musical instruments mixed together in three `.wav` files. The result of FastICA applied on this dataset was highly effective, clearly extracting the three independent, instrumental sources.

Further studies may involve identifying why certain sound mixes are not successfully separated. For example, some of our other trials indicate that a mix of human speech and recorded music are not successfully separated by ICA. At a superficial level, one may hypothesize that due to the inclusion of lyrics in the recorded music, the song and human speech may correlate in the sound files' channel values, resulting in the failure of ICA. However, the human ear transcends such difficulties and can still clearly decipher the two sources. Thus, one possibility for improvement in this specific application is the training of a neural network to map the relationship between the mixed signals and the independent components.

REFERENCES

[1] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[2] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, Inc., 2001.

[3] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[4] G. R. Naik. *Independent Component Analysis for Audio and Biosignal Applications*. IntechOpen, 2012.

[5] C. Patel. Ica - the musical way. https://www.kaggle.com/chittalpatel/ica-the-musical-way, 2020. Accessed: 2021-03-05.