Independent component analysis: An introduction

Alaa Tharwat
Faculty of Computer Science and Engineering,
Frankfurt University of Applied Sciences, Frankfurt am Main, Germany

Received 24 May 2018
Revised 26 August 2018
Accepted 29 August 2018

Abstract

Independent component analysis (ICA) is a widely-used blind source separation technique. ICA has been applied to many applications. ICA is usually utilized as a black box, without understanding its internal details. Therefore, in this paper, the basics of ICA are provided to show how it works to serve as a comprehensive source for researchers who are interested in this field. This paper starts by introducing the definition and underlying principles of ICA. Additionally, different numerical examples in a step-by-step approach are demonstrated to explain the preprocessing steps of ICA and the mixing and unmixing processes in ICA. Moreover, different ICA algorithms, challenges, and applications are presented.

Keywords Independent component analysis (ICA), Blind source separation (BSS), Cocktail party problem, Principal component analysis (PCA)

Paper type Original Article

1. Introduction

Measurements cannot be isolated from a noise which has a great impact on measured signals. For example, the recorded sound of a person in a street has sounds of footsteps, pedestrians, etc. Hence, it is difficult to record a clean measurement; this is due to (1) source signals always are corrupted with a noise, and (2) the other independent signals (e.g. car sounds) which are generated from different sources [31]. Thus, the measurements can be defined as a combination of many independent sources. The topic of separating these mixed signals is called *blind source separation* (BSS).The term blind indicates that the source signals can be separated even if little information is known about the source signals.

One of the most widely-used examples of BSS is to separate voice signals of people speaking at the same time, this is called *cocktail party problem* [31]. The *independent component analysis* (ICA) technique is one of the most well-known algorithms which are used for solving this problem [23]. The goal of this problem is to detect or extract the sound with a single object even though different sounds in the environment are superimposed on one another [31]. Figure 1 shows an example of the cocktail party problem. In this example, two voice signals are recorded from two different individuals, i.e., two independent source signals.

Publishers note: The publisher wishes to inform readers that the article “Independent component analysis: An introduction” was originally published by the previous publisher of Applied Computing and Informatics and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Tharwat, A. (2020), “Independent component analysis: An introduction”, New England Journal of Entrepreneurship, Vol. ahead-of-print No. ahead-of-print. https://10.1016/j.aci.2018.08.006. The original publication date for this paper was 31/08/2018.

Applied Computing and Informatics
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2210-8327
DOI 10.1016/j.aci.2018.08.006

Moreover, two sensors, i.e., microphones, are used for recording two signals, and the outputs from these sensors are two mixtures. The goal is to extract original signals[1] from mixtures of signals. This problem can be solved using *independent component analysis* (ICA) technique [23].

ICA was first introduced in the 80s by J. Hérault, C. Jutten and B. Ans, and the authors proposed an iterative real-time algorithm [15]. However, in that paper, there is no theoretical explanation was presented and the proposed algorithm was not applicable in a number of cases. However, the ICA technique remained mostly unknown till 1994, where the name of ICA appeared and introduced as a new concept [9]. The aim of ICA is to extract useful information or source signals from data (a set of measured mixture signals). These data can be in the form of images, stock markets, or sounds. Hence, ICA was used for extracting source signals in many applications such as medical signals [7,34], biological assays [3], and audio signals [2]. ICA is also considered as a dimensionality reduction algorithm when ICA can delete or retain a single source. This is also called filtering operation, where some signals can be filtered or removed [31].

ICA is considered as an extension of the *principal component analysis* (PCA) technique [9,33]. However, PCA optimizes the covariance matrix of the data which represents second-order statistics, while ICA optimizes higher-order statistics such as kurtosis. Hence, PCA finds uncorrelated components while ICA finds independent components [21,33]. As a consequence, PCA can extract independent sources when the higher-order correlations of mixture data are small or insignificant [21].

ICA has many algorithms such as *FastICA* [18], *projection pursuit* [21], and *Infomax* [21]. The main goal of these algorithms is to extract independent components by (1) maximizing the non-Gaussianity, (2) minimizing the mutual information, or (3) using maximum likelihood (ML) estimation method [20]. However, ICA suffers from a number of problems such as over-complete ICA and under-complete ICA.

Many studies treating the ICA technique as a black box without understanding the internal details. In this paper, in a step-by-step approach, the basic definitions of ICA, and how to use ICA for extracting independent signals are introduced. This paper is divided into eight sections. In Section 2, an overview of the definition of the main idea of ICA and its background are introduced. This section begins by explaining with illustrative numerical examples how signals are mixed to form mixture signals, and then the unmixing process is presented. Section 3 introduces with visualized steps and numerical examples two preprocessing steps of ICA, which greatly help for extracting source signals. Section 4 presents principles of how ICA extracts independent signals using different approaches such as maximizing the likelihood, maximizing the non-Gaussianity, or minimizing the mutual information. This section explains mathematically the steps of each approach. Different ICA algorithms are highlighted in Section 5. Section 6 lists some applications that use ICA for recovering independent sources from a set of sensed signals that result from a mixing set of

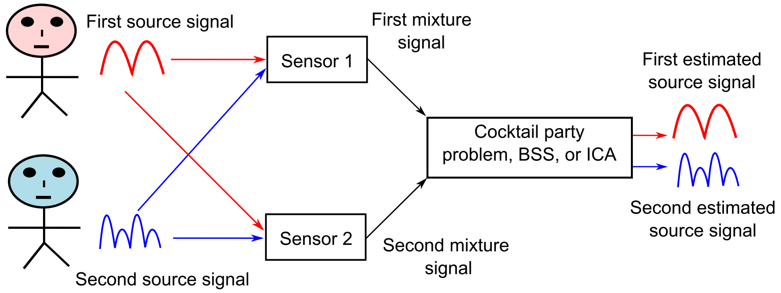


Figure 1.
Example of the cocktail party problem. Two source signals (e.g. sound signals) are generated from two individuals and then recorded by two sensors, e.g., microphones. Two microphones mixed the two source signals linearly. The goal of this problem is to recover the original signals from the mixed signals.

source signals. In Section 7, the most common problems of ICA are explained. Finally, concluding remarks will be given in Section 8.

2. ICA background

2.1 Mixing signals

Each signal varies over time and a signal is represented as follows, $\mathbf{s}_i = \{s_{i1}, s_{i2}, \ldots, s_{iN}\}$, where N is the number of time steps and s_{ij} represents the amplitude of the signal $\mathbf{s}_i$ at the jth time.[2] Given two independent source signals[3] $\mathbf{s}_1 = \{s_{11}, s_{12}, \ldots, s_{1N}\}$ and $\mathbf{s}_2 = \{s_{21}, s_{22}, \ldots, s_{2N}\}$ (see Figure 2). Both signals can be represented as follows:

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} (s_{11}, s_{12}, \ldots, s_{1N}) \\ (s_{21}, s_{22}, \ldots, s_{2N}) \end{pmatrix} \tag{1}$$

where $\mathbf{S} \in R^{p \times N}$ represents the space that is defined by source signals and p indicates the number of source signals.[4] The source signals ($\mathbf{s}_1$ and $\mathbf{s}_2$) can be mixed as follows, $\mathbf{x}_1 = a \times \mathbf{s}_1 + b \times \mathbf{s}_2$, where a and b are the mixing coefficients and $\mathbf{x}_1$ is the first mixture signal. Thus, the mixture $\mathbf{x}_1$ is the weighted sum of the two source signals ($\mathbf{s}_1$ and $\mathbf{s}_2$). Similarly, another mixture ($\mathbf{x}_2$) can be measured by changing the distance between the source signals and the sensing device, e.g. microphone, and it is calculated as follows, $\mathbf{x}_2 = c \times \mathbf{s}_1 + d \times \mathbf{s}_2$, where c and d are mixing coefficients. The two mixing coefficients a and b are different than the coefficients c and d because the two sensing devices which are used for sensing these signals are in different locations, so that each sensor measures a different mixture of source signals. As a consequence, each source signal has a different impact on output signals. The two mixtures can be represented as follows:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} a\mathbf{s}_1 + b\mathbf{s}_2 \\ c\mathbf{s}_1 + d\mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \mathbf{As} \tag{2}$$

where $\mathbf{X} \in R^{n \times N}$ is the space that is defined by the mixture signals and n is the number of mixtures. Therefore, simply, the mixing coefficients ($a, b, c,$ and d) are utilized for transforming linearly source signals from $\mathbf{S}$ space to mixed signals in $\mathbf{X}$ space as follows, $\mathbf{S} \rightarrow \mathbf{X}: \mathbf{X} = \mathbf{AS}$, where $\mathbf{A} \in R^{n \times p}$ is the mixing coefficients matrix (see Figure 2) and it is defined as:

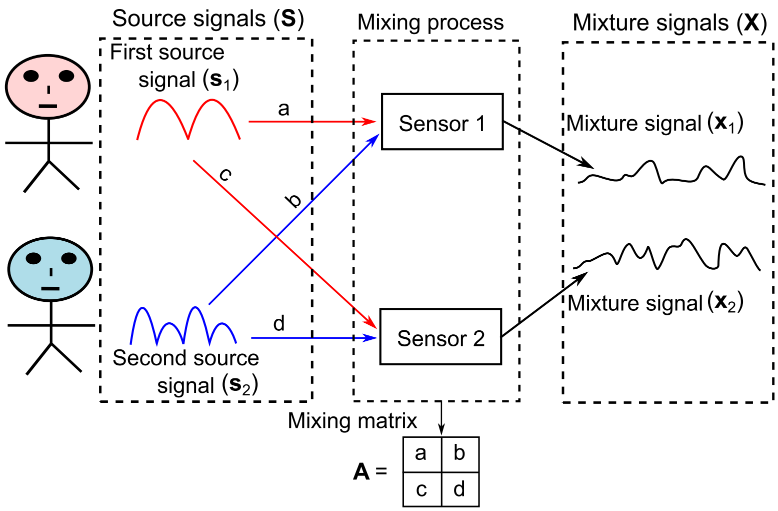


Figure 2. An illustrative example of the process of mixing signals. Two source signals are mixed linearly by the mixing matrix (**A**) to form two new mixture signals.

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{3}$$

2.1.1 Illustrative example. The goal of this example is to show the properties of source and mixture signals. Given two source signals $\mathbf{s}_1 = sin(a)$ and $\mathbf{s}_2 = r - 0.5$, where a is in the range of [1,30] with time step 0.05 and r indicates a random number in the range of [0,1]. Figure 3 shows source signals, histograms, and scatter diagram of both signals. As shown, the two source signals are independent and their histograms are not Gaussian. The scatter diagram in Figure 3(e) shows how the two source signals are independent, where each point represents the amplitude of both source signals. Figure 4 shows the mixture signals with their histograms and scatter diagram. As shown, the histograms of both mixture signals are approximately Gaussian, and the mixtures are not independent. Moreover, the mixture signals are more complex than the source signals. From this example, it is remarked that the mixed signals have the following properties:

1. **Independence**: if the source signals are independent (as in Figure 3(a and b)), their mixture signals are not (see Figure 4(a and b)). This is because the source signals are shared between both mixtures.

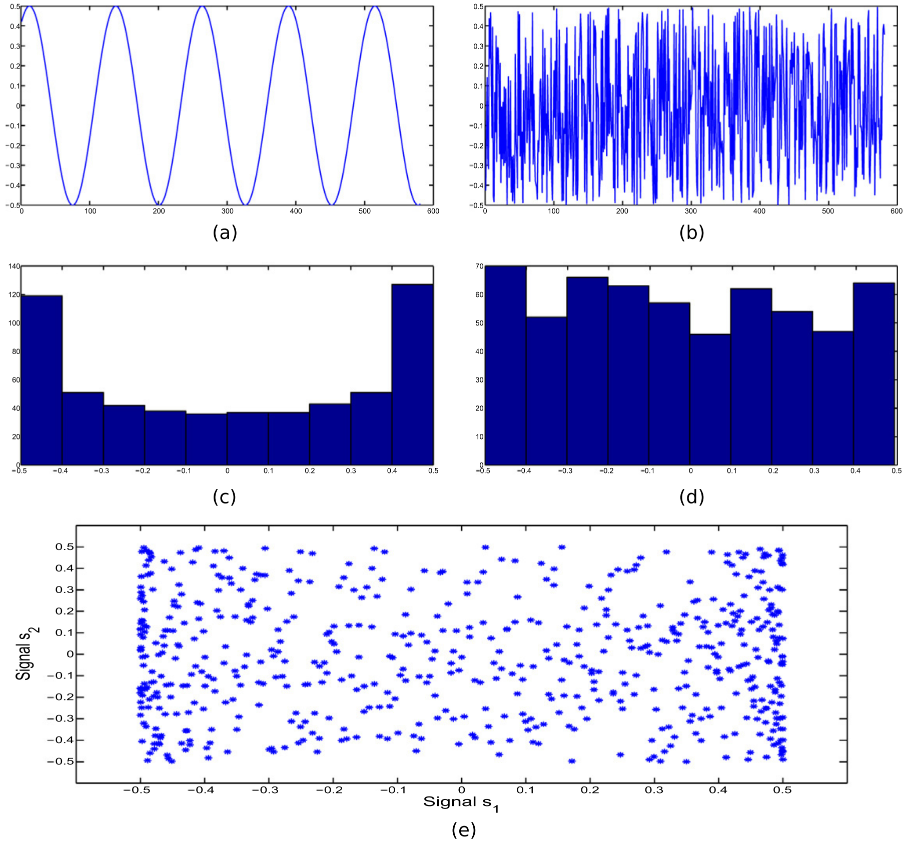


Figure 3.
An illustrative example for two source signals. (a) and (b) first and second source signals ($\mathbf{s}_1$ and $\mathbf{s}_2$), (c) and (d) histograms of $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively, (e) scatter diagram of source signals, where $\mathbf{s}_1$ and $\mathbf{s}_2$ represent the x-axis and y-axis, respectively.

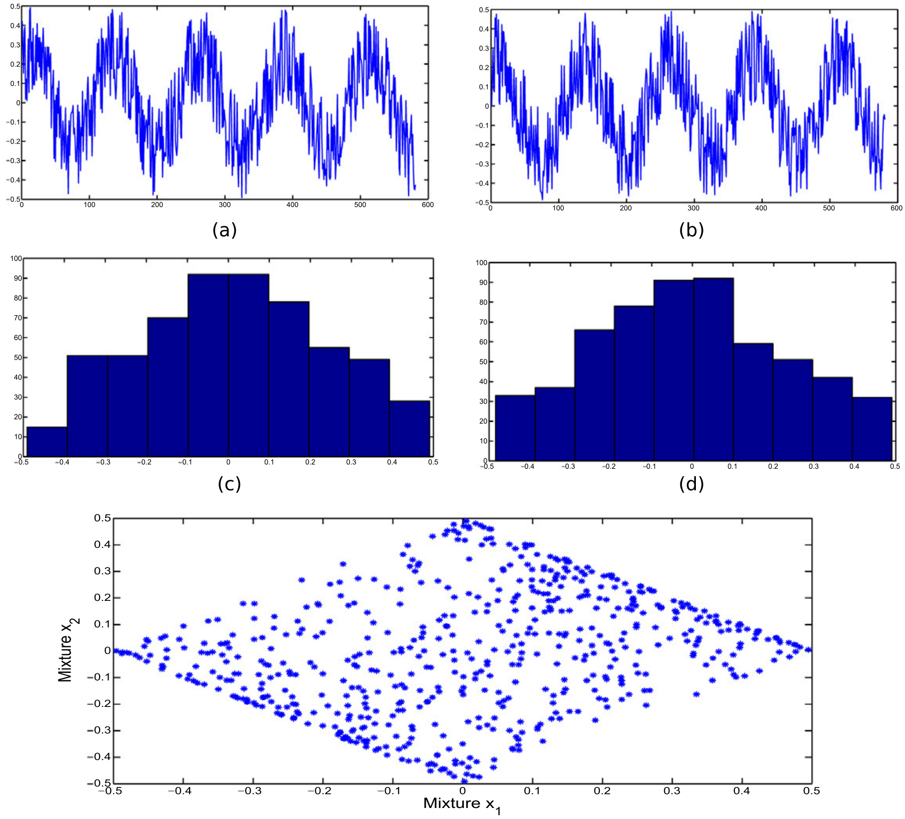


Figure 4.
An illustrative example for two mixture signals (a) and (b) first and second mixture signals $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, (c) and (d) the histogram of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, (e) scatter diagram of both mixture signals, where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the x-axis and y-axis, respectively.

2. **Gaussianity**: the histogram of mixed signals are bell-shaped histogram (see Figure 4e., Gaussian or normal. This property can be used for searching for non-Gaussian signals within mixture signals to extract source or independent signals. In other words, the source signals must be non-Gaussian, and this assumption is a fundamental restriction in ICA. Hence, the ICA model cannot estimate Gaussian independent components.

3. **Complexity**: It is clear from the previous example that mixed signals are more complex than source signals.

From these properties we can conclude that if the extracted signals from mixture signals are independent, have non-Gaussian histograms, or have low complexity than mixture signals; then these signals represent source signals.

2.1.2 Numerical example: Mixing signals. The goal of this example[5] is to explain how source signals are mixed to form mixture signals. Figure 5 shows two source signals $\mathbf{s}_1$ and $\mathbf{s}_2$ which form the space $\mathbf{S}$. The two axes of the $\mathbf{S}$ space ($\mathbf{s}_1$ and $\mathbf{s}_2$) represent the x-axis and y-axis, respectively. Additionally, the vector with coordinates $\begin{pmatrix} 1 & 0 \end{pmatrix}^T$ lie on the axis $\mathbf{s}_1$ in $\mathbf{S}$ and hence simply, the symbol $\mathbf{s}_1$ refers to this vector and similarly, $\mathbf{s}_2$ refers to the vector with the following coordinates $\begin{pmatrix} 0 & 1 \end{pmatrix}^T$. During the mixing process, the matrix $\mathbf{A}$ transforms $\mathbf{s}_1$ and $\mathbf{s}_2$ in the $\mathbf{S}$ space to $\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$, respectively, in the $\mathbf{X}$ space (see Eqs. (4) and (10)).

$$\mathbf{s}_1^{'} = \mathbf{A}\mathbf{s}_1 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix} \tag{4}$$

$$\mathbf{s}_2^{'} = \mathbf{A}\mathbf{s}_2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b \\ d \end{pmatrix} \tag{5}$$

In our example, assume that the mixing matrix is as follows, $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix}$. Given two source signals are as follows, $\mathbf{s}_1 = (1 \quad 2 \quad 1 \quad 2)$ and $\mathbf{s}_2 = (1 \quad 1 \quad 2 \quad 2)$. These two signals can be represented by four points which are plotted in the $\mathbf{S}$ space in black color (see Figure 5). The coordinates of these points are as follows:

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \tag{6}$$

The new axes in the $\mathbf{X}$ space ($\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$) are plotted in solid red and blue color, respectively (see Figure 5) and and they can be calculated as follows:

$$\mathbf{s}_1^{'} = \mathbf{A}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{7}$$

$$\mathbf{s}_2^{'} = \mathbf{A}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \tag{8}$$

The four points are transformed in the $\mathbf{X}$ space; these points are plotted in a red color in Figure 5; and the values of these new points are

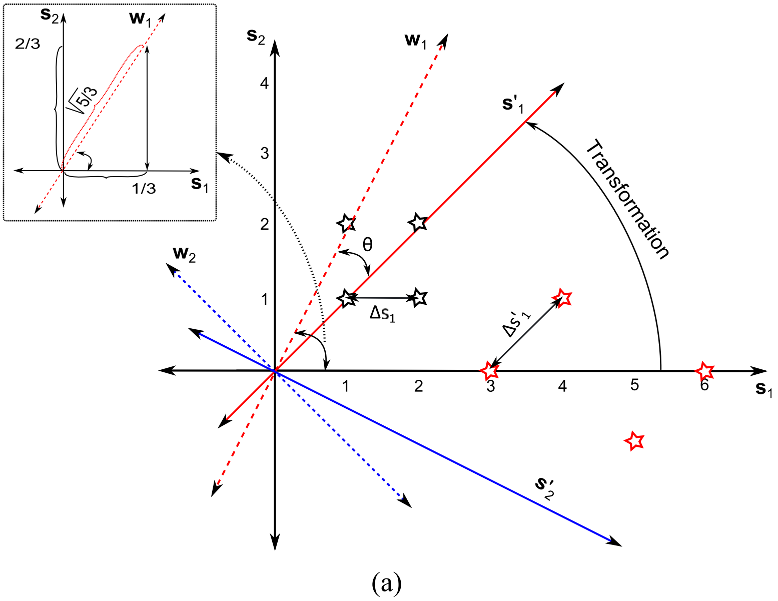


Figure 5.
An example of the mixing process. The mixing matrix $\mathbf{A}$ transforms the two source signals ($\mathbf{s}_1$ and $\mathbf{s}_2$) in the $\mathbf{S}$ space to ($\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$) in the mixture space X. The two source signals can be represented by four points (in black color) in the $\mathbf{S}$ space. These points are also transformed using the mixing matrix $\mathbf{A}$ into different four points (in red color) in the $\mathbf{X}$ space. Additionally, the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are used to extract the source signal $\mathbf{s}_1$ and $\mathbf{s}_2$, and they are plotted in dotted red and blue lines, respectively. $\mathbf{w}_1$ and $\mathbf{w}_2$ are orthogonal on $\mathbf{s}_2^{'}$ and $\mathbf{s}_1^{'}$, respectively.

$$\begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}4\\1\end{pmatrix}, \begin{pmatrix}5\\-1\end{pmatrix}, \begin{pmatrix}6\\0\end{pmatrix} \tag{9}$$

Assumed the second source $\mathbf{s}_2$ is silent/OFF; hence, the sensors record only the signal that is generated from $\mathbf{s}_1$ (see Figure 6(a)). The mixed signals are laid along $\mathbf{s}_1^{'} = (a \quad c)^T$ and the distribution of the projected samples onto $\mathbf{s}_1^{'}$ are depicted in Figure 6(a). Similarly, Figure 6(b) shows the projection onto $\mathbf{s}_2^{'} = (b \quad d)^T$ when the first source is silent; this projection represents the mixed data. It is worth mentioning that the new axes $\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$ need not to be orthogonal on the $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. Figure 5 is the combination of Figure 6(a) and (b) when both source signals are played together and the sensors measure the two signals simultaneously.

A related point to consider is that the number of red points in Figure 6(a) which represent the projected points onto $\mathbf{s}_1^{'}$ is three while the number of original points was four. This can be interpreted mathematically by calculating the coordinates of the projected points onto $\mathbf{s}_1^{'}$. For example, the projection of the first point $(1 \quad 1)^T$ is calculated as follows, $\mathbf{s}_1^{'}(1 \quad 1)^T = \begin{pmatrix}1\\1\end{pmatrix}(1 \quad 1)^T = 2$. Similarly, the projection of the second, third, and fourth points are 3, 3, and 4, respectively. Therefore, the second and third samples were projected onto the same position onto $\mathbf{s}_1^{'}$. This is the reason why the number of projected points is three.

2.2 Unmixing signals

In this section, the unmixing process for extracting source signals will be presented. Given a mixing matrix $\mathbf{A}$, independent components can be estimated by inverting the linear system as in Eq. (2), but we know neither $\mathbf{S}$ nor $\mathbf{A}$; hence, the problem is considerably more difficult. Assume that the matrix ($\mathbf{A}$) is known; hence, source signals can be extracted. For simplicity, we assume that the number of sources and mixture signals are the same and hence the unmixing matrix is a square matrix.

Given two mixture signals $\mathbf{x}_1$ and $\mathbf{x}_2$. The aim is to extract source signals, and this can be achieved by searching for unmixing coefficients as follows:

$$\begin{aligned}\mathbf{y}_1 &= \alpha\mathbf{x}_1 + \beta\mathbf{x}_2\\ \mathbf{y}_2 &= \gamma\mathbf{x}_1 + \delta\mathbf{x}_2\end{aligned} \tag{10}$$

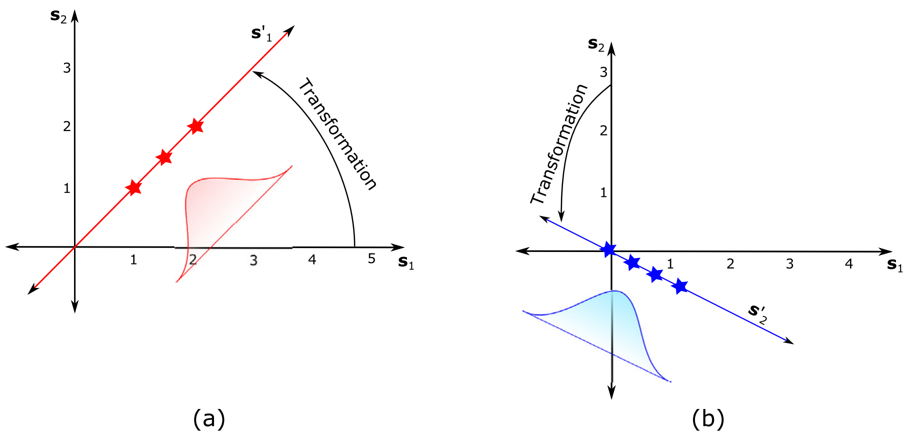


Figure 6.
An example of the mixing process. The mixing matrix $\mathbf{A}$ transforms source signals as follows: (a) $\mathbf{s}_1$ is transformed from $\mathbf{S}$ space to $\mathbf{s}_1^{'} = (a, c)^T$ (solid red line) which is one of the axes of the mixture space $\mathbf{X}$. The red stars represent the projection of the data points onto $\mathbf{s}_1^{'}$. These red stars represent all samples that are generated from the first source $\mathbf{s}_1$. (b) $\mathbf{s}_2$ is transformed from $\mathbf{S}$ space to $\mathbf{s}_2^{'} = (b, d)^T$ (solid blue line) which is one of the axes of the mixture space $\mathbf{X}$. The blue stars represent the projection of the data points onto $\mathbf{s}_2^{'}$. These blue stars represent all samples that are generated from the second source $\mathbf{s}_2$.

where α, β, γ, and δ represent unmixing coefficients, which are used for transforming the mixture signals into a set of independent signals as follow, $\mathbf{X} \rightarrow \mathbf{Y}$: $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$, where $\mathbf{W} \in R^{n \times p}$ is the unmixing coefficients matrix as shown in Figure 7. Simply we can say that the first source signal, $\mathbf{y}_1$, can be extracted from the mixtures ($\mathbf{x}_1$ and $\mathbf{x}_2$) using two unmixing coefficients (α and β). This pair of unmixing coefficients defines a point with coordinates (α, β), where $\mathbf{w}_1 = (\alpha \quad \beta)^T$ is a weight vector (see Eq. (11)). Similarly, $\mathbf{y}_2$ can be extracted using the two unmixing coefficients γ and δ which define the weight vector $\mathbf{w}_2 = (\gamma \quad \delta)^T$ (see Eq. (11))

$$\begin{aligned} \mathbf{y}_1 &= \alpha\mathbf{x}_1 + \beta\mathbf{x}_2 = \mathbf{w}_1^T\mathbf{X} \\ \mathbf{y}_2 &= \gamma\mathbf{x}_1 + \delta\mathbf{x}_2 = \mathbf{w}_2^T\mathbf{X} \end{aligned} \tag{11}$$

$\mathbf{W} = (\mathbf{w}_1 \quad \mathbf{w}_2)^T$ is the unmixing matrix and it represents the inverse of $\mathbf{A}$. The unmixing process can be achieved by rotating the rows of $\mathbf{W}$. This rotation will continue till each row in $\mathbf{W}$ ($\mathbf{w}_1$ or $\mathbf{w}_2$) finds the orientation which is orthogonal on other transformed signals. For example, in our example, $\mathbf{w}_1$ is orthogonal on $\mathbf{s}_2^{'}$ (see Figure 5). The source signals are then extracted by projecting mixture signals onto that orientation.

In practice, changing the length or orientation of weight vectors has a great influence on the extracted signals ($\mathbf{Y}$). This is the reason why the extracted signals may be not identical to original source signals. The consequences of changing the length or orientation of the weight vectors are as follows:

- **Length**: The length of the weight vector $\mathbf{w}_1$ is $|\mathbf{w}_1| = \sqrt{\alpha^2 + \beta^2}$, and assume that the length of $\mathbf{w}_1$ is changed by a factor λ as follows, $\lambda|\mathbf{w}_1| = \lambda\sqrt{\alpha^2 + \beta^2} = \sqrt{(\lambda\alpha)^2 + (\lambda\beta)^2}$. The extracted signal or the best approximation of $\mathbf{s}_1$ is denoted by $\mathbf{y}_1 = \mathbf{w}_1^T\mathbf{X}$ and it is estimated as in Eq. (12). Hence, the extracted signal is a scaled version of the source signal and the length of the weight vector affects only the amplitude of the extracted signal.

$$\begin{aligned} \mathbf{y}_1 &= \left(\lambda\mathbf{w}_1^T\right)\mathbf{X} = (\lambda\alpha)\mathbf{x}_1 + (\lambda\beta)\mathbf{x}_2 \\ &= \lambda(\alpha\mathbf{x}_1 + \beta\mathbf{x}_2) = \lambda\mathbf{s}_1 \end{aligned} \tag{12}$$

- **Orientation**: As mentioned before, the source signals $\mathbf{s}_1$ and $\mathbf{s}_2$ in the $\mathbf{S}$ space are transformed to $\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$ (see Eqs. (4) and (5)), respectively, where $\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$ form the mixture space $\mathbf{X}$. The signal $(\mathbf{s}_1)$ is extracted only if $\mathbf{w}_1$ is orthogonal to $\mathbf{s}_2^{'}$ and hence at different orientations, different signals are extracted. This is because the inner product

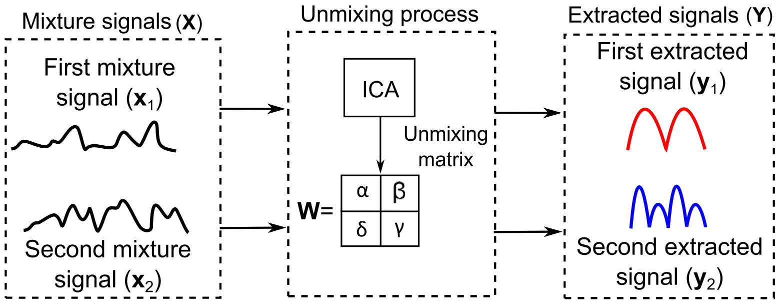

Mixture signals (X)
First mixture signal (x1)
Second mixture signal (x2)
Unmixing process
ICA
Unmixing matrix
W=
α
β
δ
γ
Extracted signals (Y)
First extracted signal (y1)
Second extracted signal (y2)

Figure 7.
An illustrative example of the process of extracting signals. Two source signals ($\mathbf{y}_1$ and $\mathbf{y}_2$) are extracted from two mixture signals ($\mathbf{x}_1$ and $\mathbf{x}_2$) using the unmixing matrix $\mathbf{W}$.

for any orthogonal vectors is zero as follows, $\mathbf{y}_1 = \mathbf{w}_1^T\mathbf{X} = \mathbf{w}_1^T\mathbf{A}\mathbf{S} = \mathbf{w}_1^T(\,\mathbf{s}_1' \quad \mathbf{s}_2'\,)$, where $\mathbf{w}_1\mathbf{s}_2' = 0$ because $\mathbf{w}_1$ is orthogonal to $\mathbf{s}_2'$, and the inner product of $\mathbf{w}_1$ and $\mathbf{s}_1'$ is as follows, $\mathbf{w}_1^T\mathbf{s}_1' = |\mathbf{w}_1||\mathbf{s}_1'|cos\theta = |\mathbf{w}_1||\mathbf{A}\mathbf{s}_1|cos\theta = k\mathbf{s}_1$, where θ is the angle between $\mathbf{w}_1$ and $\mathbf{s}_1'$ as shown in Figure 5, and k is a constant. The value of k depends on the length of $\mathbf{w}_1$ and $\mathbf{s}_1'$ and the angle θ. The extracted signal will be as follows, $\mathbf{y}_1 = \mathbf{w}_1^T(\,\mathbf{s}_1' \quad \mathbf{s}_2'\,) = (\mathbf{w}_1^T\mathbf{s}_1' + \mathbf{w}_1^T\mathbf{s}_2') = k\mathbf{s}_1$. The extracted signal ($k\mathbf{s}_1$) is a scaled version from the source signal ($\mathbf{s}_1$), and $k\mathbf{s}_1$ is extracted from $\mathbf{X}$ by taking the inner product of all mixture signals with $\mathbf{w}_1$ which is orthogonal to $\mathbf{s}_2'$. Thus, it is difficult to recover the amplitude of source signals.

Figure 8 displays the mixing and unmixing steps of ICA. As shown, the first mixture signal $\mathbf{x}_1$ is observed using only the first row in $\mathbf{A}$ matrix, where the first element in $\mathbf{x}_1$ is calculated as follows, $x_{11} = \{a_{11}s_{11} + a_{12}s_{21} + \ldots + a_{1p}s_{p1}\}$. Moreover, the number of mixture signals and the number of source signals are not always the same. This is because, the number of mixture signals depends on the number of sensors. Additionally, the dimension of $\mathbf{W}$ is not agree with $\mathbf{X}$; hence, $\mathbf{W}$ is transposed, and the first element in the first extracted signal ($\mathbf{y}_1$) is estimated as follows, $y_{11} = \{w_{11}x_{11} + w_{21}x_{21} + \ldots + w_{n1}x_{n1}\}$. Similarly all the other elements of all extracted signals can be estimated.

2.2.1 Numerical examples: Unmixing signals. The goal of this example is to extract source signals which are mixed in the numerical example in Section 2.1.2. The matrix $\mathbf{W}$ is the inverse of $\mathbf{A}$ and the value of $\mathbf{W}$ is $\mathbf{W} = \begin{pmatrix}\frac{1}{3} & \frac{2}{3}\\ \frac{1}{3} & \frac{-1}{3}\end{pmatrix}$, where the vector $\mathbf{w}_1$ in $\mathbf{W}$ is orthogonal to $\mathbf{s}_2'$, i.e., the inner product $\begin{pmatrix}\frac{1}{3} & \frac{2}{3}\end{pmatrix}(\,2 \quad -1\,)^T = 0$, and similarly, the vector $\mathbf{w}_2$ is orthogonal to $\mathbf{s}_1'$ (see Figure 5). Moreover, the source signal $\mathbf{s}_1$ is extracted as follows, $\mathbf{s}_1 = \mathbf{w}_1^T\ \mathbf{X}\ = \begin{pmatrix}\frac{1}{3} & \frac{2}{3}\end{pmatrix}\begin{pmatrix}1 & 2\\ 1 & -1\end{pmatrix} = \begin{pmatrix}1\\ 0\end{pmatrix}$, and similarly, $\mathbf{s}_2$ is extracted as follows,

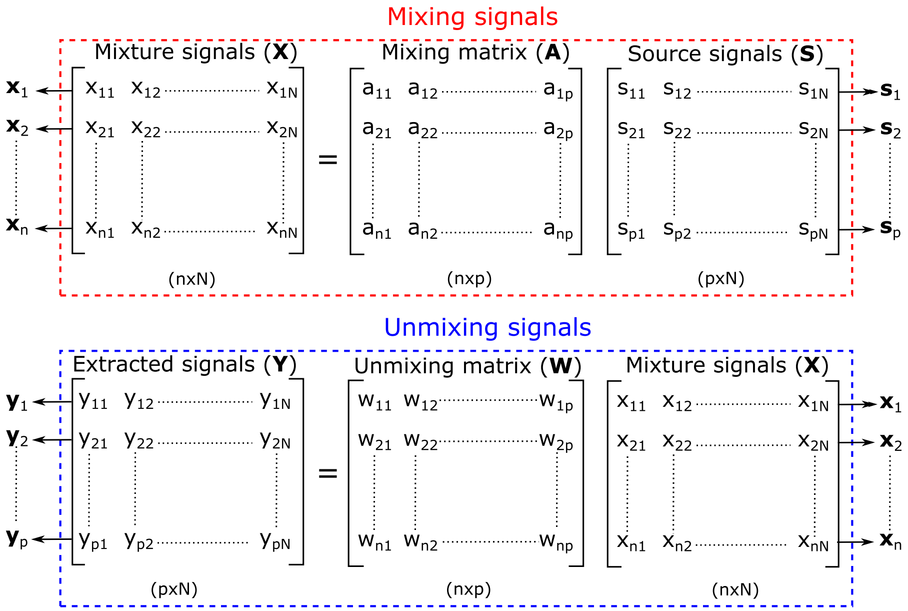


Figure 8. Block diagram of the ICA mixing and unmixing steps. a_{ij} is the mixing coefficient for the ith mixture signal and jth source signal, and w_{ij} is the unmixing coefficient for the ith extracted signal and jth mixture signal.

$\mathbf{s}_2 = \mathbf{w}_2^T\mathbf{X} = \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Hence, the original source signals are extracted perfectly. This is because $k \approx 1$ and hence according to Eq. (12) the extracted signal is identical to the source signal. As mentioned before, the value of k is calculated as follows, $k = |\mathbf{w}_1||\mathbf{s}_1^{'}|cos\theta$, and the value of $|\mathbf{w}_1| = \sqrt{(\frac{1}{3})^2 + (\frac{2}{3})^2} = \frac{\sqrt{5}}{3}$, and the value of $|\mathbf{s}_1^{'}| = \sqrt{(1)^2 + (1)^2} = \sqrt{2}$. The angle between $\mathbf{s}_1^{'}$ and the $\mathbf{s}_1$ axes is 45° because $\mathbf{s}_1^{'} = (1 \quad 1)^T$; and similarly, the angle between $\mathbf{w}_1$ and $\mathbf{s}_1$ is $cos^{-1}(\frac{1/3}{\sqrt{5/3}}) = cos^{-1}(\frac{1}{\sqrt{5}}) \approx 63°$ (see Figure 5 top left corner). Therefore, $\theta \approx 63° - 45° \approx 18°$, and hence $k = \sqrt{\frac{5}{9}}\sqrt{2}cos18° \approx 1$. Hence, changing the orientation of $\mathbf{w}_1$ leads to a different extracted signal.

2.3 Ambiguities of ICA

ICA has some ambiguities such as:

- **The order of independent components**: In ICA, the weight vector $(\mathbf{w}_i)$ is initialized randomly and then rotated to find one independent component. During the rotation, the value of $\mathbf{w}_i$ is updated iteratively. Thus, $\mathbf{w}_i$ extracts source signals but not in a specific order.

- **The sign of independent components**: Changing the sign of independent components has not any influence on the ICA model. In other words, we can multiply the weight vectors in $\mathbf{W}$ by -1 without affecting the extracted signal. In our example, in Section 2.2.1, the value of $\mathbf{w}_1$ was $\begin{pmatrix} \frac{1}{3} & \frac{2}{3} \end{pmatrix}$. Multiplying $\mathbf{w}_1$ by -1, i.e., $\mathbf{w}_1 = \begin{pmatrix} -\frac{1}{3} & -\frac{2}{3} \end{pmatrix}$ has no influence because $\mathbf{w}_1$ still in the same direction with the same magnitude and hence the value of k will not be changed, and the extracted signal s_1 will be with the same values but with a different sign, i.e., $\mathbf{s}_1 = \mathbf{w}_1^T\mathbf{X} = (-1 \quad 0)^T$. As a result, the matrix $\mathbf{W}$ in n-dimensional space has $2n$ local maxima, i.e., two local maxima for each independent component, corresponding to $\mathbf{s}_i$ and $-\mathbf{s}_i$ [21]. This problem is insignificant in many applications [16,19].

3. ICA: Preprocessing phase

This section explains the preprocessing steps of the ICA technique. This phase has two main steps: *centering* and *whitening*.

3.1 The centering step

The goal of this step is to center the data by subtracting the mean from all signals. Given n mixture signals $(\mathbf{X})$, the mean is μ and the centering step can be calculated as follows:

$$\mathbf{D} = \mathbf{X} - \mu = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_n \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 - \mu \\ \mathbf{x}_2 - \mu \\ \vdots \\ \mathbf{x}_n - \mu \end{pmatrix} \tag{13}$$

where $\mathbf{D}$ is the mixture signals after the centering step as in Figure 9A) and $\mu \in R^{1 \times N}$ is the mean of all mixture signals. The mean vector can be added back to independent components after applying ICA.

3.2 The whitening data step

This step aims to whiten the data which means transforming signals into uncorrelated signals and then rescale each signal to be with a unit variance. This step includes two main steps as follows.

1. **Decorrelation**: The goal of this step is to decorrelate signals; in other words, make each signal uncorrelated with each other. Two random variables are considered uncorrelated if their covariance is zero.

   In ICA, the PCA technique can be used for decorrelating signals. In PCA, eigenvectors which form the new PCA space are calculated.In PCA, first, the covariance matrix is calculated. The covariance matrix of any two variables $(x_i x_j)$ is defined as follows, $\Sigma_{ij} = E\{x_i x_j\} - E\{x_i\}E\{x_j\} = E[(x_i - \mu_i)(x_j - \mu_j)]$. With many variables, the covariance matrix is calculated as follows, $\Sigma = E[\mathbf{DD}^T]$, where $\mathbf{D}$ is the centered data (see Figure 9**B**)). The covariance matrix is solved by calculating the eigenvalues (λ) and eigenvectors $(\mathbf{V})$ as follows, $\mathbf{V}\Sigma = \lambda\mathbf{V}$, where the eigenvectors represent the principal components which represent the directions of the PCA space and the eigenvalues are

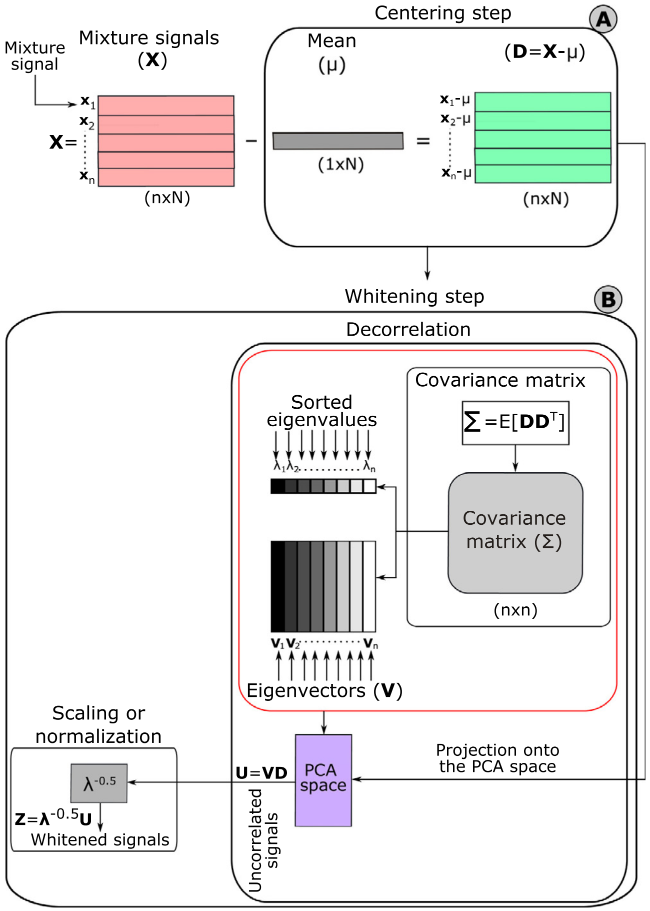


Figure 9.
Visualization for the preprocessing steps in ICA. (A) the centering step, (B) The whitening data step.

scalar values which represent the magnitude of the eigenvectors. The eigenvector which has the maximum eigenvalue is the first principal component (PC_1) and it has the maximum variance [33]. For decorrelating mixture signals, they are projected onto the calculated PCA space as follows, $\mathbf{U} = \mathbf{VD}$.

2. **Scaling**: the goal here is to scale each decorrelated signal to be with a unit variance. Hence, each vector in $\mathbf{U}$ has a unit length and is then rescaled to be with a unit variance as follows, $\mathbf{Z} = \lambda^{-\frac{1}{2}}\mathbf{U} = \lambda^{-\frac{1}{2}}\mathbf{VD}$, where $\mathbf{Z}$ is the whitened or sphered data and $\lambda^{\frac{-1}{2}}$ is calculated by simple component-wise operation as follows, $\lambda^{-\frac{1}{2}} = \{\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}}, \ldots, \lambda_n^{-\frac{1}{2}}\}$. After the scaling step, the data becomes rotationally symmetric like a sphere; therefore, the whitening step is also called *sphering* [32].

3.3 Numerical example

Given eight mixture signals $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_8\}$, each mixture signal is represented by one row in $\mathbf{X}$ as in Eq. (14).[6] The mean (μ) was then calculated and its value was $\mu = 2.63 \quad 3.63$.

$$\mathbf{X}^T = \begin{bmatrix} 1.00 & 1.00 & 2.00 & 0.00 & 5.00 & 4.00 & 5.00 & 3.00 \\ 3.00 & 2.00 & 3.00 & 3.00 & 4.00 & 5.00 & 5.00 & 4.00 \end{bmatrix} \tag{14}$$

In the centering step, the data are centered by subtracting the mean from each signal and the value of $\mathbf{D}$ will be as follows:

$$\mathbf{D}^T = \begin{bmatrix} -1.63 & -1.63 & -0.63 & -2.63 & 2.38 & 1.38 & 2.38 & 0.38 \\ -0.63 & -1.63 & -0.63 & -0.63 & 0.38 & 1.38 & 1.38 & 0.38 \end{bmatrix} \tag{15}$$

The covariance matrix (Σ) and its eigenvalues (λ) and eigenvectors ($\mathbf{V}$) are then calculated as follows:

$$\Sigma = \begin{bmatrix} 3.70 & 1.70 \\ 1.70 & 1.13 \end{bmatrix}, \lambda = \begin{bmatrix} 0.28 & 0.00 \\ 0.00 & 4.54 \end{bmatrix}, \text{ and } \mathbf{V} = \begin{bmatrix} 0.45 & -0.90 \\ -0.90 & -0.45 \end{bmatrix} \tag{16}$$

From Eq. (16) it can be remarked that the two eigenvectors are orthogonal as shown in Figure 10, i.e., $\mathbf{v}_1^T\mathbf{v}_2 = [0.45 - 0.9] - 0.90 - 0.45^T = 0$, where $\mathbf{v}_1$ and $\mathbf{v}_2$ represent the first and second eigenvectors, respectively. Moreover, the value of the second eigenvalue (λ_2) was more than the first one (λ_1), and λ_2 represents $\frac{4.54}{0.28+4.54} \approx 94.19\%$ of the total eigenvalues; thus, $\mathbf{v}_2$ and $\mathbf{v}_1$ represent the first and second principal components of the PCA space, respectively, and $\mathbf{v}_2$ points to the direction of the maximum variance (see Figure 10).

The two signals are decorrelated by projecting the centered data onto the PCA space as follows, $\mathbf{U} = \mathbf{VD}$. The values of $\mathbf{U}$ is

$$\mathbf{U}^T = \begin{bmatrix} -0.16 & 0.73 & 0.28 & -0.61 & 0.72 & -0.62 & -0.18 & -0.17 \\ 1.73 & 2.18 & 0.84 & 2.63 & -2.29 & -1.84 & -2.74 & -0.50 \end{bmatrix} \tag{17}$$

The matrix $\mathbf{U}$ is already centered; thus, the covariance matrix for $\mathbf{U}$ is given by

$$E(\mathbf{U}\mathbf{U}^T) = \begin{bmatrix} 0.28 & 0 \\ 0 & 4.54 \end{bmatrix} \tag{18}$$

From Eq. (18) it is remarked that the two mixture signals are decorrelated by projecting them onto the PCA space. Thus, the covariance matrix is diagonal and the off-diagonal elements which represent the covariance between two mixture signals are zeros. Figure 10 displays the contour of the two mixtures is ellipsoid centered at the mean. The projection of mixture signals onto the PCA space rotates the ellipse so that the principal components are aligned with the x_1 and x_2 axes. After the decorrelation step, the signals are then rescaled to be with a unit variance (see Figure 10). The whitening can be calculated as follows, $\mathbf{Z} = \lambda^{-\frac{1}{2}}\mathbf{VD}$, and the values of the mixture signals after the scaling step are

$$\mathbf{Z}^T = \begin{bmatrix} -0.31 & 1.38 & 0.53 & -1.15 & 1.36 & -1.17 & -0.33 & -0.32 \\ 0.81 & 1.02 & 0.39 & 1.23 & -1.08 & -0.87 & -1.29 & -0.24 \end{bmatrix} \tag{19}$$

The covariance matrix for the whitened data is $E[\mathbf{Z}\mathbf{Z}^T] = E[(\lambda^{-0.5}\mathbf{VD})(\lambda^{-0.5}\mathbf{VD})^T] = E[(\lambda^{-0.5}\mathbf{VD})(\mathbf{D}^T\mathbf{V}^T\lambda^{-0.5})]$. λ is diagonal; thus, $\lambda = \lambda^T$, and $[\mathbf{D}\mathbf{D}^T]$ is the covariance matrix (Σ) which is equal to $\mathbf{V}^T\lambda\mathbf{V}$. Hence, $E[\mathbf{Z}\mathbf{Z}^T] = E[\lambda^{-0.5}\mathbf{V}\mathbf{V}^T\lambda\mathbf{V}\mathbf{V}^T\lambda^{-0.5}] = \mathbf{I}$, where $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ because $\mathbf{V}$ is orthonormal.[7] This means that the covariance matrix of the whitened data is the identity matrix (see Eq. (20)) which means that the data are decorrelated and have unit variance.

$$E(\mathbf{Z}\mathbf{Z}^T) = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix} \tag{20}$$

Figure 11 displays the scatter plot for two mixtures, where each mixture signal is represented by 500-time steps. As shown in Figure 11(a), the scatter of the original mixtures forms an

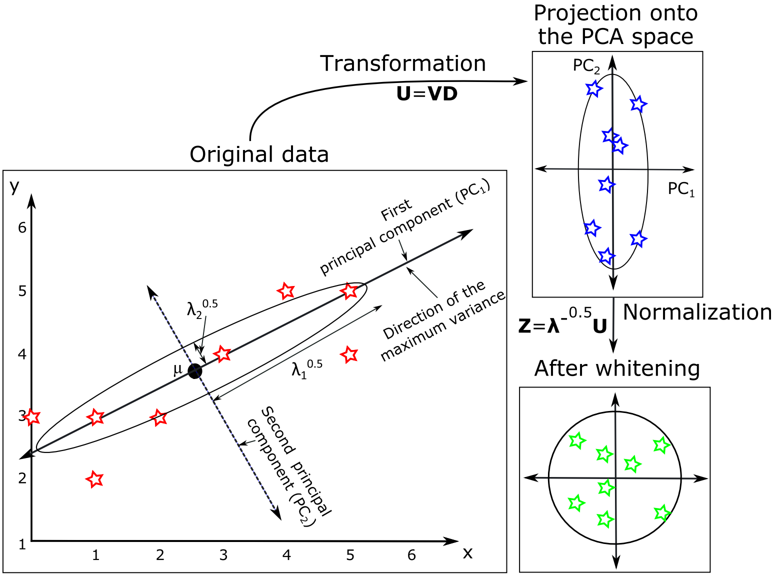


Figure 10.
Visualization for our whitening example. In the left panel, the mixture signals are plotted in red stars. This panel also shows the principal components (*PC*$_1$ and *PC*$_2$). In the top right panel, the data in a blue color represent the projected data onto the PCA space. The data are then normalized to be with a unit variance (the bottom right panel). In this panel, the data in a green color represent the whitened data.

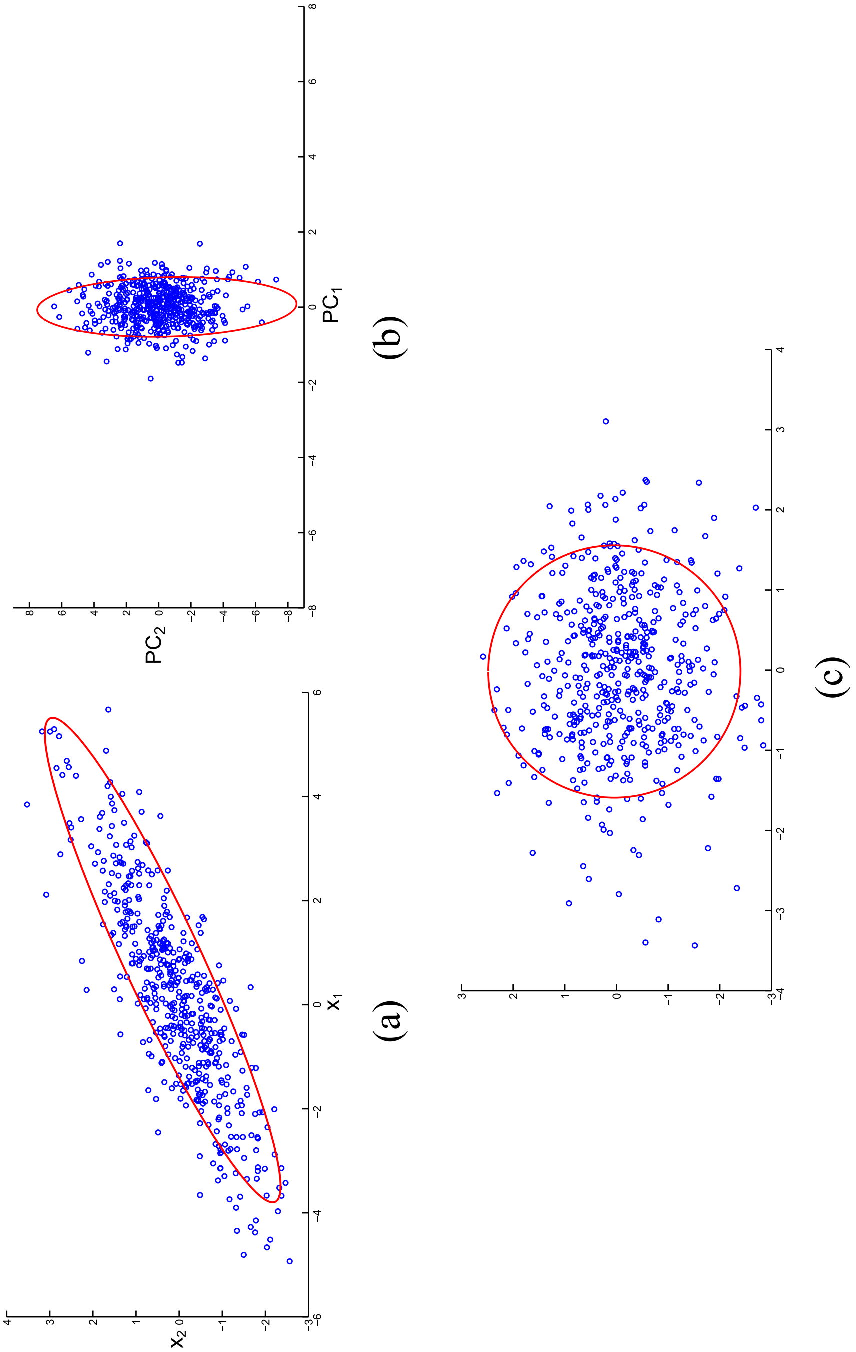


Figure 11.
Visualization for mixture signals during the whitening step. (a) scatter plot for two mixture signals x_1 and x_2, (b) the projection of mixture signals onto the PCA space, i.e., decorrelation, (c) mixture signals after the whitening step are scaled to have a unit variance.

ellipse centered at the origin. Projecting the mixture signals onto the PCA space rotates the principal components to be aligned with the x_1 and x_2 axes and hence the ellipse is also rotated as shown in Figure 11(b). After the whitening step, the contour of the mixture signals forms a circle. This is because the signals have unit variance.

4. Principles of ICA estimation

In ICA, the goal is to find the unmixing matrix ($\mathbf{W}$) and then projecting the whitened data onto that matrix for extracting independent signals. This matrix can be estimated using three main approaches of independence, which result in slightly different unmixing matrices. The first is based on the non-Gaussianity. This can be measured by some measures such as *negentropy* and *kurtosis*, and the goal of this approach is to find independent components which maximize the non-Gaussianity [25,30]. In the second approach, the ICA goal can be obtained by minimizing the mutual information [22,14]. Independent components can be also estimated by using maximum likelihood (ML) estimation [28]. All approaches simply search for a rotation or unmixing matrix $\mathbf{W}$. Projecting the whitened data onto that rotation matrix extracts independent signals. The preprocessing steps are calculated from the data, but the rotation matrix is approximated numerically through an optimization procedure. Searching for the optimal solution is difficult due to the local minima exists in the objective function. In this section, different approaches are introduced for extracting independent components.

4.1 Measures of non-Gaussianity

Searching for independent components can be achieved by maximizing the non-Gaussianity of extracted signals [23]. Two measures are used for measuring the non-Gaussianity, namely, Kurtosis and negative entropy.

4.1.1 Kurtosis. Kurtosis can be used as a measure of non-Gaussianity, and the extracted signal can be obtained by finding the unmixing vector which maximizes the kurtosis of the extracted signal [4]. In other words, the source signals can be extracted by finding the orientation of the weight vectors which maximize the kurtosis.

Kurtosis is simple to calculate; however, it is sensitive for outliers. Thus, it is not robust enough for measuring the non-Gaussianity [21]. The Kurtosis (K) of any *probability density function* (pdf) is defined as follow,

$$K(x) = E\left[x^4\right] - 3\left[E\left[x^2\right]\right]^2 \tag{21}$$

where the normalized kurtosis ($\widehat{K}$) is the ratio between the fourth and second central moments, and it is given by

$$\widehat{K}(x) = \frac{E[x^4]}{E[x^2]^2} - 3 = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right)^2} - 3 \tag{22}$$

For whitened data ($\mathbf{Z}$), $E[\mathbf{Z}^2] = 1$ because $\mathbf{Z}$ with a unit variance. Therefore, the kurtosis will be

$$K(\mathbf{Z}) = \widehat{K}(\mathbf{Z}) = E\left[\mathbf{Z}^4\right] - 3 \tag{23}$$

As reported in [20], the fourth moment for Gaussian signals is $3(E[\mathbf{Z}^2])^2$ and hence $\widehat{K}(x) = E[\mathbf{Z}^4] - 3 = E[3(E[\mathbf{Z}^2])^2] - 3 = E[3(1)^2] - 3 = 0$, where $E[\mathbf{Z}^2] = 1$. As a consequence, Gaussian pdfs have zero kurtosis.

Kurtosis has an additivity property as follows:

$$K(x_1 + x_2) = K(x_1) + K(x_2), \tag{24}$$

and for any scalar parameter α,

$$K(\alpha x_1) = \alpha^4 K(x_1) \tag{25}$$

where α is a scalar.

These properties can be used for interpreting one of the ambiguities of ICA that are mentioned in Section 2.3, which is the sign of independent components. Given two source signals $\mathbf{s}_1$ and $\mathbf{s}_2$, and the matrix $\mathbf{Q} = \mathbf{A}^T\mathbf{W} = \mathbf{A}^{-1}\mathbf{W}$. Hence,

$$\mathbf{Y} = \mathbf{W}^T\mathbf{X} = \mathbf{W}^T\mathbf{AS} = \mathbf{QS} = \mathbf{q}_1\mathbf{s}_1 + \mathbf{q}_2\mathbf{s}_2 \tag{26}$$

Using the kurtosis properties in Eqs. (24) and (25), we have

$$K(\mathbf{Y}) = K(\mathbf{q}_1\mathbf{s}_1) + K(\mathbf{q}_2\mathbf{s}_2) = \mathbf{q}_1^4 K(\mathbf{s}_1) + \mathbf{q}_2^4 K(\mathbf{s}_2) \tag{27}$$

Assume that $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{Y}$ have a unit variance. This implies that $E[\mathbf{Y}^2] = \mathbf{q}_1^2 E[\mathbf{s}_1] + \mathbf{q}_2^2 E[\mathbf{s}_2] = \mathbf{q}_1^2 + \mathbf{q}_2^2 = 1$. Geometrically, this means that $\mathbf{Q}$ is constrained to a unit circle in the two-dimensional space. The aim of ICA is to maximize the kurtosis $(K(\mathbf{Y}) = \mathbf{q}_1^4 \boldsymbol{K}(\mathbf{s}_1) + \mathbf{q}_2^4 \boldsymbol{K}(\mathbf{s}_2))$ on the unit circle. The optimal solutions, i.e., maxima, are the points when one of $\mathbf{Q}$ is zero and the other is nonzero; this is due to the unit circle constraint, and the nonzero element must be 1 or -1 [11]. These optimal solutions are the ones which are used to extract $\pm\mathbf{s}_i$. Generally, $\mathbf{Q} = \mathbf{A}^T\mathbf{W} = \mathbf{I}$ means that each vector in the matrix $\mathbf{Q}$ extracts only one source signal.

The ICs can be obtained by finding the ICs which maximizes kurtosis of extracted signals $\mathbf{Y} = \mathbf{W}^T\mathbf{Z}$. The kurtosis of $\mathbf{Y}$ is then calculated as in Eq. (23), where the term $(E[\mathbf{y}_i^2])^2$ in Eq. (22) is equal one because $\mathbf{W}$ and $\mathbf{Z}$ have a unit length. $\mathbf{W}$ has a unit length because it is scaled to be with a unit length, and $\mathbf{Z}$ is the whitened data, so, it has a unit length. Thus, the kurtosis can be expressed as:

$$K(\mathbf{Y}) = E\left[\left(\mathbf{W}^T\mathbf{Z}\right)^4\right] - 3 \tag{28}$$

The gradient of the kurtosis of $\mathbf{Y}$ is given by, $\frac{\partial K(\mathbf{W}^T\mathbf{Z})}{\partial \mathbf{W}} = cE[\mathbf{Z}(\mathbf{W}^T\mathbf{Z})^3]$, where c is a constant, which we set to unity for convenience. The weight vector is updated in each iteration as follows, $\mathbf{w}_{new} = \mathbf{w}_{old} + \eta E[\mathbf{Z}(\mathbf{w}_{old}^T\mathbf{Z})^3]$, where η is the step size for the gradient ascent. Since we are optimizing the kurtosis on the unit circle $\|\mathbf{w}\| = 1$, the gradient method must be complemented by projecting $\mathbf{w}$ onto the unit circle after every step. This can be done by normalizing the weight vectors $\mathbf{w}_{new}$ through dividing it by its norm as follows, $\mathbf{w}_{new} = \mathbf{w}_{new}/\|\mathbf{w}_{new}\|$. The value of $\mathbf{w}_{new}$ is updated in each iteration.

4.1.2 Negative entropy. Negative entropy is termed negentropy, and it is defined as follows, $J(y) = H(y_{Gaussian}) - H(y)$, where $H(y_{Gaussian})$ is the entropy of a Gaussian random variable whose covariance matrix is equal to the covariance matrix of y. The entropy of a random variable Q which has N possible outcomes is

$$H(Q) = -E\left[log\, p_q(q)\right] = -\frac{1}{N}\sum_{t}^{N} log\, p_q\left(q^t\right) \tag{29}$$

where $p_q(q^t)$ is the probability of the event q^t, $t = 1, 2, \ldots, N$.

The negentropy is zero when all variables are Gaussian, i.e., $H(y_{Gaussian}) = H(y)$. Negentropy is always nonnegative because the entropy of Gaussian variable is the maximum among all other random variables with the same variance. Moreover, it is invariant for invertible linear transformation and it is scale-invariant [21]. However, calculating the entropy from a finite data is computationally difficult. Hence, different approximations have been introduced for calculating the negentropy [21]. For example,

$$J(y) \approx \frac{1}{12} E\left[y^3\right]^2 + \frac{1}{48} K(y)^2 \tag{30}$$

where y is assumed to be with zero mean. This approximation suffers from the sensitivity of kurtosis; therefore, Hyvarinen proposed another approximation based on the maximum entropy principle as follows [23]:

$$J(y) \approx \sum_{i=1}^{p} k_i (E[G_i(y)] - E[G_i(v)])^2, \tag{31}$$

where k_i are some positive constants, v indicates a Gaussian variable with zero mean and unit variance, G_i represent some quadratic functions [23,20]. The function G has different choices such as

$$G_1(y) = \frac{1}{a_1} log\, cosh\, a_1 y \text{ and } G_2(y) = -exp\left(-y^2/2\right) \tag{32}$$

where $1 \leq a_1 \leq 2$. These two functions are widely used, and these approximations give a very good compromise between the kurtosis and negentropy properties which are the two classical non-Gaussianity measures.

4.2 Minimization of mutual information

Minimizing mutual information between independent components is one of the well-known approaches for ICA estimation. In ICA, maximizing the entropy of $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ can be achieved by spreading out the points in $\mathbf{Y}$ as much as possible. Signals $\widehat{\mathbf{Y}}$ can be obtained by transforming $\mathbf{Y}$ by g as follows, $\widehat{\mathbf{Y}} = g(\mathbf{Y})$, where g is assumed to be the *cumulative density function* cdf of source signals. Hence, $\widehat{\mathbf{Y}}$ have a uniform joint distribution.

The pdf of the linear transformation $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ is, $p_Y(\mathbf{Y}) = p_X(\mathbf{X})/|\mathbf{W}|$, where $|\mathbf{W}|$ represents $|\partial\mathbf{Y}/\partial\mathbf{X}|$. Similarly, $p_{\widehat{Y}}(\widehat{\mathbf{Y}}) = p_Y(\mathbf{Y})/\left|\frac{d\widehat{\mathbf{Y}}}{d\mathbf{Y}}\right| = \frac{p_Y(\mathbf{Y})}{p_S(\mathbf{Y})}$, where $\left|\frac{d\widehat{\mathbf{Y}}}{d\mathbf{Y}}\right|$ is equal to $g'(\mathbf{y})$ which represents the pdf for source signals (p_S).

This can be substituted in Eq. (29) and the entropy will be

$$H(\widehat{\mathbf{Y}}) = -\frac{1}{N} \sum_{t=1}^{N} log\, p_{\widehat{\mathbf{Y}}}(\widehat{\mathbf{Y}}_t) = -\frac{1}{N} \sum_{t}^{N} log \frac{p_Y(\mathbf{Y})}{p_S(\mathbf{Y})} = -\frac{1}{N} \sum_{t=1}^{N} log \frac{p_X(\mathbf{x}_t)}{|\mathbf{W}| p_S(\mathbf{y}_t)}$$

$$= \frac{1}{N} \sum_{t=1}^{N} log\, p_S(\mathbf{y}_t) + log\, |\mathbf{W}| - \frac{1}{N} \sum_{t=1}^{N} log\, p_X(\mathbf{x}_t) \tag{33}$$

In Eq. (33), increasing the matching between the extracted and source signals, the ratio $\frac{p_Y(\mathbf{Y})}{p_S(\mathbf{Y})}$ will be one. As a consequence, the $p_{\widehat{Y}}(\widehat{\mathbf{Y}}) = \frac{p_Y(\mathbf{Y})}{p_S(\mathbf{Y})}$ becomes uniform which maximizes the entropy of $p_{\widehat{Y}}(\widehat{\mathbf{Y}})$. Moreover, the term $\frac{-1}{N}\sum_{t=1}^{N} log\, p_X(\mathbf{X}_t)$ represents the entropy of $\mathbf{X}$; hence, Eq. (33) is given by

$$H(\widehat{\mathbf{Y}}) = \frac{1}{N}\sum_{t=1}^{N} log\, p_S(\mathbf{y}_t) + log\,|\mathbf{W}| + H(\mathbf{X}) \tag{34}$$

Hence, from Eq. (34), $H(\mathbf{Y}) = H(\mathbf{X}) + log\,|\mathbf{W}|$. This means that in the linear transformation $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$, the entropy is changed (increased or decreased) by $log|\mathbf{W}|$. As mentioned before, the entropy $H(\mathbf{X})$ is not affected by $\mathbf{W}$ and $\mathbf{W}$ maximizes only the entropy $H(\widehat{\mathbf{Y}})$ and hence $H(\mathbf{X})$ is removed from Eq. (34), and final form of the entropy with M marginal pdfs is

$$H(\widehat{\mathbf{Y}}) = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{M} log\, p_S\left(\mathbf{y}_{i_t}\right) + log\,|\mathbf{W}| \tag{35}$$

Mutual information measures the independence between random variables. Thus, independent components can be obtained by minimizing the mutual information between different components [6]. Given two random variables x and y, the mutual information is denoted by I, and it is given by

$$\begin{aligned} I(x,y) &= \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \\ &= H(x) - H(x|y) = H(y) - H(y|x) \\ &= H(x) + H(y) - H(x,y) \\ &= H(x,y) - H(x|y) - H(y|x) \end{aligned} \tag{36}$$

where $H(x)$ and $H(y)$ represent the marginal entropies, $H(x|y)$ and $H(y|x)$ are conditional entropies, and $H(x,y)$ is the joint entropy of x and y. The value of I is zero if and only if the variables are independent; otherwise, I is non-negative. Mutual information between m random variables $(y_i,\ i = 1,\ 2,\ \ldots,\ m)$ is given by

$$I(y_1,\ y_2,\ \ldots,\ y_m) = \sum_{i=1}^{m} H(y_i) - H(y) \tag{37}$$

In ICA, where $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ and $H(\mathbf{Y}) = H(\mathbf{X}) + log\,|\mathbf{W}|$, Eq. (37) can be written as

$$\begin{aligned} I(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m) &= \sum_{i=1}^{m} H(\mathbf{y}_i) - H(\mathbf{Y}) \\ &= \sum_{i=1}^{m} H(\mathbf{y}_i) - H(\mathbf{X}) - log|det\mathbf{W}| \end{aligned} \tag{38}$$

where $det\mathbf{W}$ is a notation for a determine of the matrix $\mathbf{W}$. When $\mathbf{Y}$ is whitened; thus, $E[\mathbf{Y}\mathbf{Y}^T] = \mathbf{W}E[\mathbf{X}\mathbf{X}^T]\mathbf{W}^T = \mathbf{I} \Rightarrow det(\mathbf{W}E[\mathbf{X}\mathbf{X}^T]\mathbf{W}^T) = (det\mathbf{W})\ (detE[\mathbf{X}\mathbf{X}^T])(det\mathbf{W}^T) \Rightarrow det\ (\mathbf{W}E[\mathbf{X}\mathbf{X}^T]\mathbf{W}^T) = det\mathbf{I} = 1$. As a consequence, $det\mathbf{W}$ is a constant, and the definition of mutual information is

$$I(y_1, y_2, \ldots, y_m) = C - \sum_i J(y_i) \tag{39}$$

where C is a constant.

From Eq. (39), it is clear that maximizing negentropy is related to minimizing mutual information and they differ only by a sign and a constant C. Moreover, non-Gaussianity measures enable the deflationary (one-by-one) estimation of the ICs which is not possible with mutual information or likelihood approaches.[8] Further, with the non-Gaussianity approach, all signals are enforced to be uncorrelated, while this constraint is not necessary using mutual information approach.

4.3 Maximum Likelihood (ML)

Maximum likelihood (ML) estimation method is used for estimating parameters of statistical models given a set of observations. In ICA, this method is used for estimating the unmixing matrix ($\mathbf{W}$) which provides the best fit for extracted signals $\mathbf{Y}$.

The likelihood is formulated in the noise-free ICA model as follows, $\mathbf{X} = \mathbf{AS}$, and this model can be estimated using ML method [6]. Hence, $p_X(\mathbf{X}) = \frac{p_S(\mathbf{S})}{|det\mathbf{A}|} = |det\mathbf{W}|p_S(\mathbf{S})$. For independent source signals, (i.e. $p_S(\mathbf{S}) = p_1(\mathbf{s}_1)p_2(\mathbf{s}_2)\dots p_p(\mathbf{s}_p) = \prod_i p_i(\mathbf{s}_i)$), $p_X(\mathbf{X})$ is given by

$$p_x(\mathbf{X}) = |det\mathbf{W}|\prod_i p_i(\mathbf{s}_i) = |det\mathbf{W}|\prod_i p_i\left(\mathbf{w}_i^T\mathbf{X}\right) \tag{40}$$

Given T observations of $\mathbf{X}$, the log-likelihood of $\mathbf{W}$ which is denoted by $L(\mathbf{W})$ is given by

$$L(\mathbf{W}) = \prod_t^T \prod_i^p |det\mathbf{W}|p_i\left(\mathbf{w}_i^T\mathbf{x}(t)\right) \tag{41}$$

Practically, the likelihood is usually simplified using the logarithm, this is called log-likelihood, which makes Eq. (41) more simpler as follows:

$$logL(\mathbf{W}) = \sum_{i=1}^p log\,p_i\left(\mathbf{w}_i^T\mathbf{x}(t)\right) + Tlog|det\mathbf{W}| \tag{42}$$

The mean of any random variable x can be calculated as $E[x] = \frac{1}{T}\sum_{i=1}^T x_t \Rightarrow \sum_{i=1}^T x_t = TE[x]$. Hence, Eq. (42) can be simplified to

$$\frac{1}{T}logL(\mathbf{W}) = E\sum_{i=1}^p log\,p_i\left(\mathbf{w}_i^T\mathbf{X}\right) + log|det\mathbf{W}| \tag{43}$$

The first term $E\sum_{i=1} log\,p_i(\mathbf{w}_i^T\mathbf{X}) = -\sum_{i=1}H(\mathbf{w}_i^T\mathbf{X})$; therefore, the likelihood and mutual information are approximately equal, and they differ only by a sign and an additive constant. It is worth mentioning that maximum likelihood estimation will give wrong results if the information of ICs are not correct; but, with the non-Gaussianity approach, we need not for any prior information [23].

5. ICA algorithms

In this section, different ICA algorithms are introduced.

5.1 Projection pursuit

Projection pursuit (PP) is a statistical technique for finding possible projections of multi-dimensional data [13]. In the basic one-dimensional projection pursuit, the aim is to find the directions where the projections of the data onto these directions have distributions which are deviated from Gaussian distribution, and this exactly is the same goal of ICA [13]. Hence, ICA is considered as a variant of projection pursuit.

In PP, one source signal is extracted from each projection, which is different than ICA algorithms that extract p signals simultaneously from n mixtures. Simply, in PP, after finding the first projection which maximizes the non-Gaussianity, the same process is repeated to find new projections for extracting next source signal(s) from the reduced set of mixture signals, and this sequential process is called *deflation* [17].

Given n mixture signals which represent the axes of the n-dimensional space ($\mathbf{X}$). The nth source signal can be extracted using the vector $\mathbf{w}_n$ which is orthogonal to the other $n-1$ axes. These mixture signals in the n-dimensional space are projected onto the $(n-1)$-dimensional space which has $n-1$ transformed axes. For example, assume $n = 3$, and the third source signal can be extracted by finding $\mathbf{w}_3$ which is orthogonal to the plane that is defined by the other two transformed axes $s_1^{'}$ and $s_2^{'}$; this plane is denoted by $p_{1,2}^{'}$. Hence, the data points in three-dimensional space are projected onto the plane $p_{1,2}^{'}$ which is a two-dimensional space. This process is continued until all source signals are extracted [20,32].

Given three source signals each source signal has 10000 time-steps as shown in Figure 12. These signals represent sound signals. These sound signals were collected from Matlab, where the first signal is called *Chrip*, the second signal is called *gong*, and the third is called *train*. Figure 12 (d, e, and f) shows the histogram for each signal. As shown, the histograms are non-Gaussian. These three signals were mixed, and the mixing matrix was as follows:

$$A = \begin{pmatrix} 1.5 & 0.7 & 0.2 \\ 0.6 & 0.2 & 0.9 \\ 0.1 & 1 & 0.6 \end{pmatrix} \tag{44}$$

Figure 13 shows the mixed signals and the histogram for these mixture signals. As shown in the figure, the mixture signals follow all the properties that were mentioned in Section 2.1.1, where (1) source signals are more independent than mixture signals, (2) the histograms of mixture signals in Figure 13 are much more Gaussian than the histogram of source signals in Figure 12 mixtures signals (see Figure 13 are more complex than source signals (see Figure 12)).

In the projection pursuit algorithm, mixture signals are first whitened, and then the values of the first weight vector ($\mathbf{w}_1$) are initialized randomly. The value of $\mathbf{w}_1$ is listed in Table 1. This weight vector is then normalized, and it will be used for extracting one source signal ($\mathbf{y}_1$). The kurtosis for the extracted signal is then calculated and the weight vector is updated to maximize the kurtosis iteratively. Table 1 shows the kurtosis of the extracted signal during some iterations of the projection pursuit algorithm. It is remarked that the kurtosis increases during the iterations as shown in Figure 14(a). Moreover, in this example, the correlation between the extracted signal ($\mathbf{y}_1$) and all source signals ($\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$) were calculated. This may help to understand that how the extracted signal is correlated with one source signal and not correlated with the other signals. From the table, it can be remarked that the correlation between $\mathbf{y}_1$ and source signals are changed iteratively, and the correlation between $\mathbf{y}_1$ and $\mathbf{s}_1$ was 1 at the end of iterations.

Figure 15 shows the histogram of the extracted signal during the iteration. As shown in Figure 15(a), the extracted signal is Gaussian; hence, its kurtosis value which represents the measure of non-Gaussianity in the projection pursuit algorithm is small (0.18). The kurtosis value of the extracted signal increased to 0.21, 3.92, and 4.06 after the 10th, 100th, and 1000th iterations, respectively. This reflects that the non-Gaussianity of $\mathbf{y}_1$ increased during the iterations of the projection pursuit algorithm. Additionally, Figure 14(b) shows the angle between the optimal vector and the gradient vector (α). As shown, the value of the angle is dramatically decreased and it reached zero which means that both the optimal and gradient vectors have the same direction.

5.2 FastICA

FastICA algorithm extracts independent components by maximizing the non-Gaussianity by maximizing the negentropy for the extracted signals using a fixed-point iteration scheme [18].

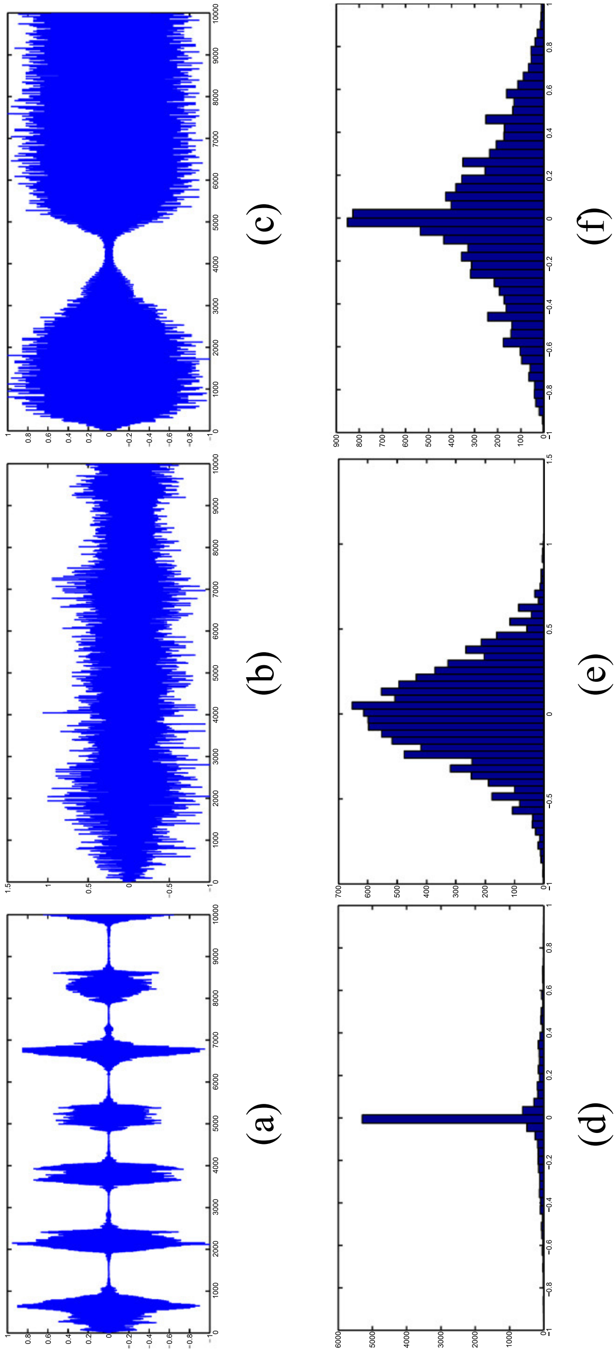


Figure 12.
Three source signals in our example (a, b, and c) and their histograms (d, e, and f).

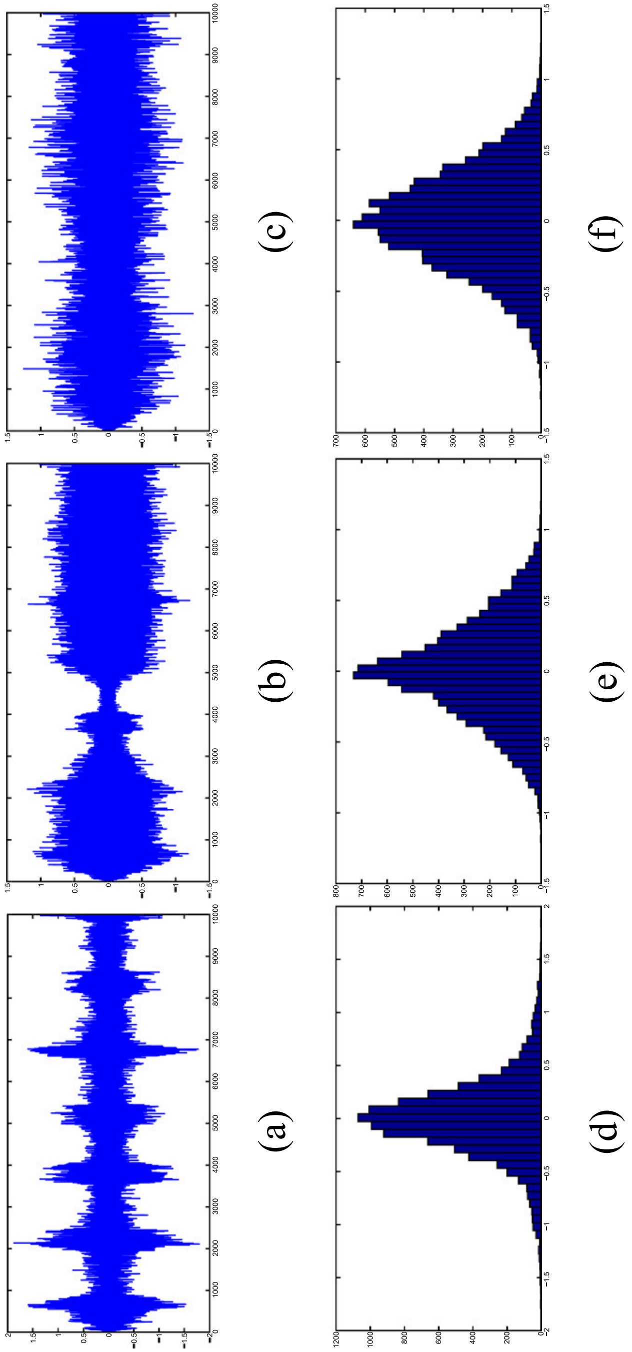


Figure 13.
Three mixture signals in our example (a, b, and c) and their histograms (d, e, and f).

FastICA has a cubic or at least quadratic convergence speed and hence it is much faster than Gradient-based algorithms that have linear convergence. Additionally, FastICA has no learning rate or other adjustable parameters which makes it easy to use.

FastICA can be used for extracting one IC, this is called *one-unit*, where FastICA finds the weight vector ($\mathbf{w}$) that extracts one independent component. The values of $\mathbf{w}$ are updated by a learning rule that searches for a direction which maximizes the non-Gaussianity.

The derivative of the function G in Eq. (31) is denoted by g, and the derivatives for G_1 and G_2 in Eq. (32) are:

$$g_1(y) = tanh(a_1 u)\,\text{and}\,g_2(y) = u\,exp\left(-u^2/2\right) \tag{45}$$

where $1 \le a_1 \le 2$ is a constant, and often $a_1 = 1$. In FastICA, the convergence means that the dot-product between the current and old weight vectors is almost equal to one and hence the values of the new and old weight vectors are in the same direction. The maxima of the approximation of the negentropy of $\mathbf{w}^T\mathbf{X}$ is calculated at a certain optima of $E[G(\mathbf{w}^T\mathbf{X})]$, where $E[(\mathbf{w}^T\mathbf{X})^2] = \|\mathbf{w}^2\| = 1$. The optimal solution is obtained where, $E[\mathbf{X}g(\mathbf{w}^T\mathbf{X})] - \beta\mathbf{w} = 0$, and this equation can be solved using Newton's method.[9] Let $F(\mathbf{w}) = E[\mathbf{X}g(\mathbf{w}^T\mathbf{X})] - \beta\mathbf{w}$; hence, the Jacobian matrix is given by, $JF(\mathbf{w}) = \frac{\partial F}{\partial \mathbf{w}} = E[\mathbf{X}\mathbf{X}^T g'(\mathbf{w}^T\mathbf{X})] - \beta\mathbf{I}$. Since the data are whitened; thus, $[\mathbf{X}\mathbf{X}^T g'(\mathbf{w}^T\mathbf{X})] \approx E[\mathbf{X}\mathbf{X}^T] E[g'(\mathbf{w}^T\mathbf{X})] \Rightarrow E[\mathbf{X}\mathbf{X}^T g'(\mathbf{w}^T\mathbf{X})] = E[g'(\mathbf{w}^T\mathbf{X})]\mathbf{I}$ and hence the Jacobian matrix becomes diagonal, which is easily inverted. The value of $\mathbf{w}$ can be updated according to Newton's method as follows:

$$\mathbf{w}^+ = \mathbf{w} - \frac{F(\mathbf{w})}{F'(\mathbf{w})} = \mathbf{w} - \frac{E[\mathbf{X}g(\mathbf{w}^T\mathbf{X})] - \beta\mathbf{w}}{E[g'(\mathbf{w}^T\mathbf{X})] - \beta} \tag{46}$$

Eq. (46) can be further simplified by multiplying both sides by $\beta - E[g'(\mathbf{w}^T\mathbf{X})]$ as follows:

$$\mathbf{w}^+ = E\left[\mathbf{X}g\left(\mathbf{w}^T\mathbf{X}\right)\right] - E\left[g'\left(\mathbf{w}^T\mathbf{X}\right)\right]\mathbf{w} \tag{47}$$

Several units of FastICA can be used for extracting several independent components, the output $\mathbf{w}_i^T\mathbf{X}$ is decorrelated iteratively with the other outputs which were calculated in the previous iterations ($\mathbf{w}_1^T\mathbf{X}, \mathbf{w}_2^T\mathbf{X}, \ldots, \mathbf{w}_{i-1}^T\mathbf{X}$). This decorrelation step prevents different vectors from converging to the same optima. *Deflation orthogonalization* method is similar to the projection pursuit, where the independent components are estimated one by one. For each iteration, the projections of the previously estimated weight vectors $(\mathbf{w}_p\mathbf{w}_j)\mathbf{w}_j$ are subtracted from $\mathbf{w}_p$, where $j = 1, 2, \ldots, p-1$, and then $\mathbf{w}_p$ is normalized as in Eq. (48). In this method, estimation errors in the first vectors are cumulated over the next ones by orthogonalization. *Symmetric orthogonalization* method can be used when a symmetric correlation, i.e., no vectors are privileged over others, is required [18]. Hence, the vectors $\mathbf{w}_i$ can be estimated in

Table 1. Results of the projection pursuit algorithm in terms of the correlation between the extracted signal ($\mathbf{y}_1$) and source signals, values of the weight vector ($\mathbf{w}_1$), kurtosis of $\mathbf{y}_1$, and the angle between the optimal vector and the gradient vector (α) during the iterations of the projection pursuit algorithm.

Results	*iter* = 1	*iter* =10	*iter* = 100	*iter* = 1000
0.3	0.33	0.99	1.00	$corr(\mathbf{y}_1, \mathbf{s}_1)$
0.94	0.93	0.14	0.00	$corr(\mathbf{y}_1, \mathbf{s}_2)$
0.14	0.13	0.01	0.01	$corr(\mathbf{y}_1, \mathbf{s}_3)$
$\mathbf{w}_1$	$\begin{pmatrix} -0.50 \\ 0.23 \\ -0.84 \end{pmatrix}$	$\begin{pmatrix} -0.46 \\ 0.24 \\ -0.85 \end{pmatrix}$	$\begin{pmatrix} 0.42 \\ 0.72 \\ -0.5 \end{pmatrix}$	$\begin{pmatrix} 0.50 \\ 0.74 \\ -0.45 \end{pmatrix}$
$K(\mathbf{y}_1)$	0.18	0.21	3.92	4.06
α	1.22	1.18	0.07	$5.3e^{0-5}$

parallel which enables parallel computation. This method calculates all w_i vectors using one-unit algorithm in parallel, and then the orthogonalization step is applied for all vectors using symmetric method as follows, $\mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}\mathbf{W}$, where $(\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}$ is calculated from the eigenvalue decomposition as follows, $\mathbf{V}(\mathbf{W}\mathbf{W}^T) = \lambda\mathbf{V}$; thus, $(\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}} = \mathbf{V}^T\lambda^{-\frac{1}{2}}\mathbf{V}$.

$$\begin{aligned} &1.\mathbf{w}_p = \mathbf{w}_p - \sum_{j=1}^{p} \mathbf{w}_p^T \mathbf{w}_j \mathbf{w}_j \\ &2.\mathbf{w}_p = \frac{\mathbf{w}_p}{\sqrt{\mathbf{w}_p^T \mathbf{w}_p}} \end{aligned} \tag{48}$$

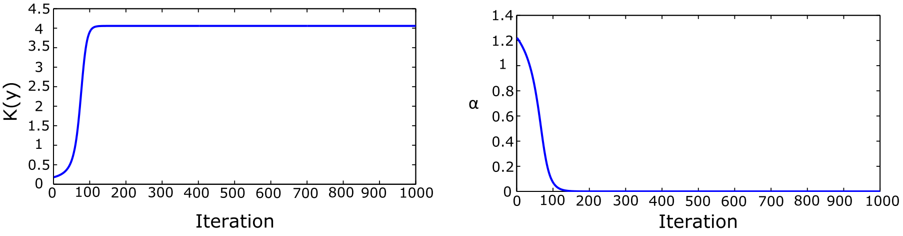
6. Applications

ICA has been used in many applications for extracting source signals from a set of mixed signals. These applications include:

- **Biomedical applications:** ICA was used for removing artifacts which mixed with different biomedical signals such as *Electroencephalogram* (EEG), *functional magnetic resonance imaging* (fMRI), and *Magnetoencephalography* (MEG) signals [5]. Also, ICA was used for removing the *electrocardiogram* (ECG) interference from EEG signals, or for differentiating between the brain signals and the other signals that are generated from different activities as in [29].
- **Audio signal processing:** ICA has been widely used in audio signals for removing noise [36]. Additionally, ICA was used as a feature extraction method to design robust automatic speech recognition models [8].
- **Biometrics:** ICA is for extracting discriminative features in different biometrics such as face recognition [10], ear recognition [35], and finger print [27].
- **Image processing:** ICA is used in image segmentation to extract different layers from the original image [12]. Moreover, ICA is widely used for noise removing from raw images which represent the original signals [24].

7. Challenges of ICA

ICA is used for estimating the unknown matrix $\mathbf{W} = \mathbf{A}^{-1}$. When the number of sources (p) and the number of mixture signals (n) are equal, the matrix $\mathbf{A}$ is invertible. When the number



(a)

(b)

Figure 14.
Results of the projection pursuit algorithm. (a) Kurtosis of the extracted signal ($\mathbf{y}_1$) during some iterations of the projection pursuit algorithm, (b) the angle between the optimal vector and gradient vector (α) during some iterations of the projection pursuit algorithm.

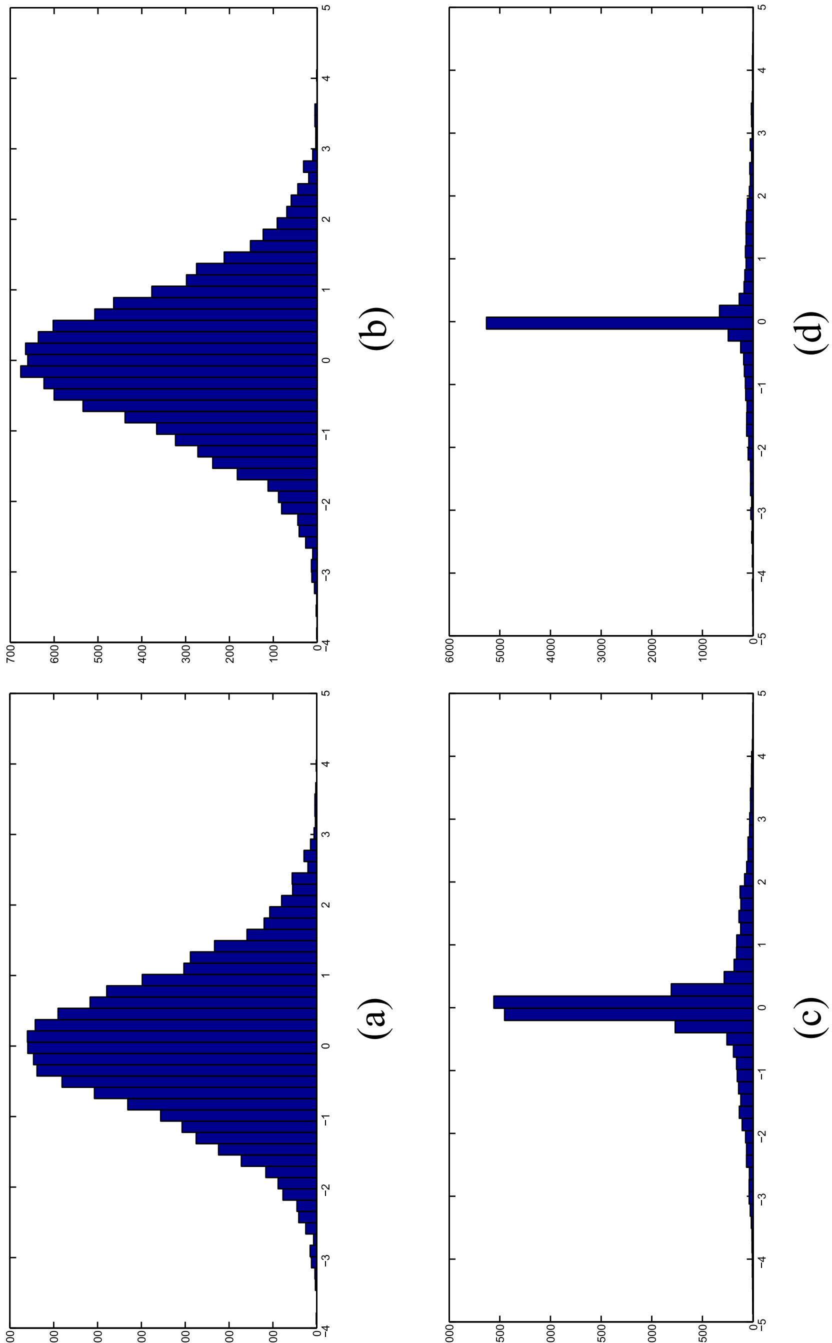


Figure 15.
Histogram of the extracted signal ($\mathbf{y}_1$). (a) after the first iteration, (b) after the tenth iteration, (c) after the 100th iteration, and (d) after the 1000th iteration.

of mixtures is less than the number of source signals $(n < p)$ this is called the *over-complete* problem; thus, A is not square and not invertible [26]. This representation sometimes is advantageous as it uses as few "basis" elements as possible; this is called sparse coding. On the other hand, when $n > p$ means that the number of mixtures is higher than the number of source signals and this is called the *Under-complete* problem. This problem can be solved by deleting some mixtures using dimensionality reduction techniques such as PCA to decrease the number of mixtures [1].

8. Conclusions

ICA is a widely-used statistical technique which is used for estimating independent components (ICs) through maximizing the non-Gaussianity of ICs, maximizing the likelihood of ICs, or minimizing mutual information between ICs. These approaches are approximately equivalent; however, each approach has its own limitations.

This paper followed the approach of not only explaining the steps for estimating ICs, but also presenting illustrative visualizations of the ICA steps to make it easy to understand. Moreover, a number of numerical examples are introduced and graphically illustrated to explain (1) how signals are mixed to form mixture signals, (2) how to estimate source signals, and (3) the preprocessing steps of ICA. Different ICA algorithms are introduced with detailed explanations. Moreover, ICA common challenges and applications are briefly highlighted.

Notes

[1] In this paper, original signals, source signals, or *independent components* (ICs) are the same.

[2] In this paper, source and mixture signals are represented as random variables instead of time series or time signals, i.e., the time index is dropped.

[3] Two signals $\mathbf{s}_1$ and $\mathbf{s}_2$ are independent if the amplitude of $\mathbf{s}_1$ is independent of the amplitude of $\mathbf{s}_2$.

[4] In this paper, all bold lowercase letters denote vectors and bold uppercase letters indicate matrices.

[5] In all numerical examples, the numbers are rounded up to the nearest hundredths (two numbers after the decimal point).

[6] Due to the paper size, Eq. (14) indicates $\mathbf{X}^T$ instead of $\mathbf{X}$; hence, each column represents one signal/sample. Similarly, $\mathbf{D}$ in Eq. (15), $\mathbf{U}$ in Eq. (17), and $\mathbf{Z}$ in Eq. (19).

[7] Two vectors x and y are orthonormal if they are orthogonal, i.e., the dot product $x.y = 0$, and they are unit vectors, i.e., $(x) = (y) = 1$.

[8] Maximum Likelihood approach will be introduced in the next section.

[9] Assume $f(x) = 0$, using Newton's method, the solution is calculated as follows, $x_{i+1} = x_i - \frac{f(x)}{f'(x)}$.

References

[1] S.-I. Amari, Natural gradient learning for over-and under-complete bases in ICA, Neural Comput. 11 (8) (1999) 1875–1883.

[2] A. Asaei, H. Bourlard, M.J. Taghizadeh, V. Cevher, Computational methods for underdetermined convolutive speech localization and separation via model-based sparse component analysis, Speech Commun. 76 (2016) 201–217.

[3] R. Aziz, C. Verma, N. Srivastava, A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data, Genomics data 8 (2016) 4–15.

[4] E. Bingham, A. Hyvärinen, A fast fixed-point algorithm for independent component analysis of complex valued signals, Int. J. Neural Syst. 10 (01) (2000) 1–8.

[5] V.D. Calhoun, J. Liu, T. Adal, A review of group ICA for FMRA data and ICA for joint inference of imaging, genetic, and ERP data, Neuroimage 45 (1) (2009) S163–S172.

[6] J.-F. Cardoso, Infomax and maximum likelihood for blind source separation, IEEE Sig. Process. Lett. 4 (4) (1997) 112–114.

[7] R. Chai, G.R. Naik, T.N. Nguyen, S.H. Ling, Y. Tran, A. Craig, H.T. Nguyen, Driver fatigue classification with independent component by entropy rate bound minimization analysis in an eeg-based system, IEEE J. Biomed. Health Inf. 21 (3) (2017) 715–724.

[8] J.-W. Cho, H.-M. Park, Independent vector analysis followed by hmm-based feature enhancement for robust speech recognition, Sig. Process. 120 (2016) 200–208.

[9] P. Comon, Independent component analysis, a new concept?, Sig. Process. 36 (3) (1994) 287–314.

[10] I. Dagher, R. Nachar, Face recognition using ipca-ica algorithm, IEEE Trans. Pattern Anal. Machine Intell. 28 (6) (2006) 996–1000.

[11] N. Delfosse, P. Loubaton, Adaptive blind separation of independent sources: a deflation approach, Sig. Process. 45 (1) (1995) 59–83.

[12] S. Derrode, G. Mercier, W. Pieczynski, Unsupervised multicomponent image segmentation combining a vectorial hmc model and ica, in: Proceedings of International Conference on Image Processing (ICIP), Vol. 2, IEEE, 2003, pp. II–407.

[13] J.H. Friedman, J.W. Tukey, A projection pursuit algorithm for exploratory data analysis, IEEE Trans. Comput. 100 (9) (1974) 881–890.

[14] S.S. Haykin, S.S. Haykin, S.S. Haykin, S.S. Haykin, Neural Netw. Learn. Machines, Vol. 3, Pearson Upper Saddle River, NJ, USA, 2009.

[15] J. Hérault, C. Jutten, B. Ans, Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In: 10 Colloque sur le traitement du signal et des images, FRA, 1985.GRETSI, Groupe d'Etudes du Traitement du Signal et des Images 1985.

[16] A. Hyvärinen, Independent component analysis in the presence of gaussian noise by maximizing joint likelihood, Neurocomputing 22 (1) (1998) 49–67.

[17] A. Hyvärinen, New approximations of differential entropy for independent component analysis and projection pursuit. In: Advances in neural information processing systems. (1998b) pp. 273–279.

[18] A. Hyvarinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks 10 (3) (1999) 626–634.

[19] A. Hyvarinen, Gaussian moments for noisy independent component analysis, IEEE Signal Process. Lett. 6 (6) (1999) 145–147.

[20] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Vol. 46, John Wiley & Sons, 2004.

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

[22] D. Langlois, S. Chartier, D. Gosselin, An introduction to independent component analysis: Infomax and fastica algorithms, Tutorials Quantit. Methods Psychol. 6 (1) (2010) 31–38.

[23] T.-W. Lee, Independent component analysis, in: Independent Component Analysis, Springer, 1998, pp. 27–66.

[24] T.-W. Lee, M.S. Lewicki, Unsupervised image classification, segmentation, and enhancement using ica mixture models, IEEE Trans. Image Process. 11 (3) (2002) 270–279.

[25] T.-W. Lee, M.S. Lewicki, T.J. Sejnowski, Ica mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1078–1089.

[26] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Learning 12 (2) (2006).

[27] F. Long, B. Kong, Independent component analysis and its application in the fingerprint image preprocessing, in: Proceedings. International Conference on Information Acquisition, IEEE, 2004, pp. 365–368.

[28] B.A. Pearlmutter, L.C. Parra, Maximum likelihood blind source separation: A context-sensitive generalization of ica. In: Advances in neural information processing systems. 1997, pp. 613–619.

[29] M.B. Pontifex, K.L. Gwizdala, A.C. Parks, M. Billinger, C. Brunner, Variability of ica decomposition may impact eeg signals when used to remove eyeblink artifacts, Psychophysiology 54 (3) (2017) 386–398.

[30] S. Shimizu, P.O. Hoyer, A. Hyvärinen, A. Kerminen, A linear non-gaussian acyclic model for causal discovery, J. Mach. Learn. Res. 7 (Oct) (2006) 2003–2030.

[31] J. Shlens, A tutorial on independent component analysis. arXiv preprint arXiv:1404.2986, 2014.

[32] J.V. Stone, 2004. Independent component analysis. A tutorial introduction. A bradford book.

[33] A. Tharwat, Principal component analysis-a tutorial, Int. J. Appl. Pattern Recognit. 3 (3) (2016) 197–240.

[34] J. Xie, P.K. Douglas, Y.N. Wu, A.L. Brody, A.E. Anderson, Decoding the encoding of functional brain networks: an fmri classification comparison of non-negative matrix factorization (nmf), independent component analysis (ica), and sparse coding algorithms, J. Neurosci. Methods 282 (2017) 81–94.

[35] H.-J. Zhang, Z.-C. Mu, W. Qu, L.-M. Liu, C.-Y. Zhang, A novel approach for ear recognition based on ica and rbf network, in: Proceedings of International Conference on Machine Learning and Cybernetics, Vol. 7, IEEE, 2005, pp. 4511–4515.

[36] M. Zibulevsky, B.A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural Computat. 13 (4) (2001) 863–882.

Corresponding author
Alaa Tharwat can be contacted at: aothman@fb2.fra-uas.de