Tutorials in Quantitative Methods for Psychology
2010, Vol. 6(1), p. 31-38.

An Introduction to Independent Component Analysis: InfoMax and FastICA algorithms

Dominic Langlois, Sylvain Chartier, and Dominique Gosselin
University of Ottawa

This paper presents an introduction to independent component analysis (ICA). Unlike principal component analysis, which is based on the assumptions of uncorrelatedness and normality, ICA is rooted in the assumption of statistical independence. Foundations and basic knowledge necessary to understand the technique are provided hereafter. Also included is a short tutorial illustrating the implementation of two ICA algorithms (FastICA and InfoMax) with the use of the Mathematica software.

Nowadays, performing statistical analysis is only a few clicks away. However, before anyone carries out the desired analysis, some assumptions must be met. Of all the assumptions required, one of the most frequently encountered is about the normality of the distribution (Gaussianity). However, there are many situations in which Gaussianity does not hold. Human speech (amplitude by time), electrical signals from different brain areas and natural images are all examples not normally distributed. The well-known "cocktail party effect" illustrates this concept well. Let us imagine two people standing in a room and speaking simultaneously. If two microphones are placed in two different places in the room, they will each record a particular linear combination of the two voices. Using only the recordings, would it then be possible to identify the voice of each speaker (Figure 1a)? If Gaussianity was assumed, we could perform a Principal Component Analysis (PCA) or a Factorial Analysis (FA). The resulting components would be two new orderly voice combinations (Figure 1a). Therefore, such a technique fails to isolate each speaker's voice.

On the other hand, if non-Gaussianity is assumed, then Independent Component Analysis (ICA) could be applied to the same problem and the result would be quite different. ICA is able to distinguish the voice of each speaker from the linear combination of their voices (Figure 1b). This reasoning can be applied to many biological recording involving multiple source signals (e.g. EEG). However, the readers must bear in mind that there are two main differences in the interpretation of extracted components using ICA instead of PCA. First, in ICA, there is no order of magnitude associated with each component. In other words, there is no better or worst components (unless the user decides to order them following his own criteria). Second, the extracted components are invariant to the sign of the sources. For example, in image processing, a white letter on a black background is the same as a black letter on a white background.

The remainder of the paper is comprised of a first section that briefly exposes the theoretical foundations of ICA[1], and of a second section that gives an example of its application using two different implemented algorithms (supplemental material). The second section also presents a short discussion on future tracks of research.

Theoretical foundations of ICA

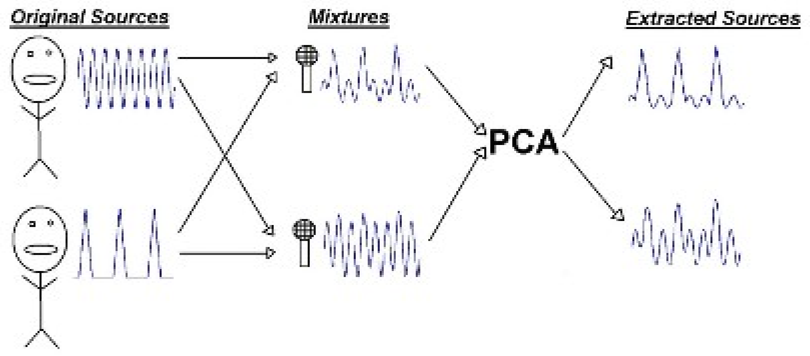
Let us denote the random observed vector $\mathbf{X} = [X_1, X_2, \ldots, X_m]^\mathrm{T}$ whose m elements are mixtures of m independent elements of a random vector $\mathbf{S} = [S_1, S_2, \ldots, S_m]^\mathrm{T}$ given by

$$\mathbf{X} = \mathbf{AS} \qquad (1)$$

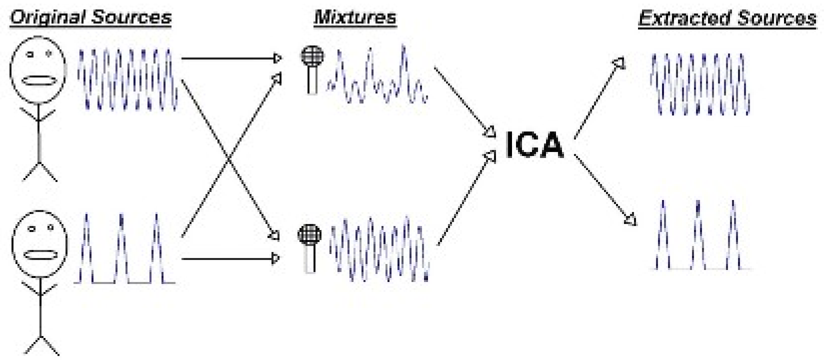
We wish to thank Marianna Gati for her valuable comments and helpful suggestions. This work was supported by scholarships from the Fonds québécois de recherche sur la nature et les technologies (FQRNT) and the Ontario Graduate Scholarship Program (OGS).

Figure 1. Comparison between PCA (a) and ICA (b) in the context of the "cocktail party effect".

a)



b)



Where $\mathbf{A}$ represents an $m \times m$ mixing matrix, the sample value of X_j is denoted by x_j and j=1, 2, ..., m. The goal of ICA is to find the unmixing matrix **W** (i.e. the inverse of **A**) that will give **Y**, the best possible approximation of **S**:

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \cong \mathbf{S} \tag{2}$$

In order to use ICA, five assumptions must be met. First, statistical independence between each of the sources S_i from the sources vector **S** is assumed (independence is at the core of ICA and will be discussed further in the next subsection). Second, the mixing matrix must be square and full rank. In other words, the number of mixtures must be equal to the number of sources and the mixtures must be linearly independent from each other.[2] Third, the only source of stochasticity in the model is the source vector **S** (i.e. there is no external noise). The model must thus be noise free. Fourth, it is assumed that the data are centered (zero mean). Also, for some algorithms, the data must be pre-processed further; sometimes, the observation vector **X** must be whitened.[3] Fifth, the source signals must not have a Gaussian probability density function (pdf) except for one single source that can be Gaussian.

Statistical independence

Let $x_1, x_2, \ldots, x_m$ be random variables with pdf $f(x_1, x_2, \ldots, x_m)$, then the variables x_i are mutually independent if:

$$f(x_1, x_2, \ldots, x_m) = f_1(x_1) f_2(x_2) \ldots f_m(x_m) \tag{3}$$

that is, if the pdf of the x_i is equal to the multiplication of each marginal pdf of the x_i. Statistical independence is a more severe criterion than uncorrelatedness between two variables. If we take random centered variables, uncorrelatedness is expressed by the following equation:

6. $\mathbf{w}_i = \dfrac{\mathbf{w}_i^+}{\|\mathbf{w}_i^+\|}$ (13d)

7. If not converged, go back to step 2. Else go back to step 1 with i = i + 1 until all components are extracted.

where w_i is a column-vector of the unmixing matrix W, $\mathbf{w}_i^+$ is a temporary variable used to calculate w_i (it is the new w_i before normalization), $\phi'(.)$ is the derivative of $\phi(.)$ and $E(.)$ is the expected value (mean). Once a given w_i has converged, the next one (w_{i+1}) must be made orthogonal to it (and all those previously extracted) with Equations 13c and 13d in order for the new component to be different from it (them). This algorithm has been implemented in the package FastICA.nb.

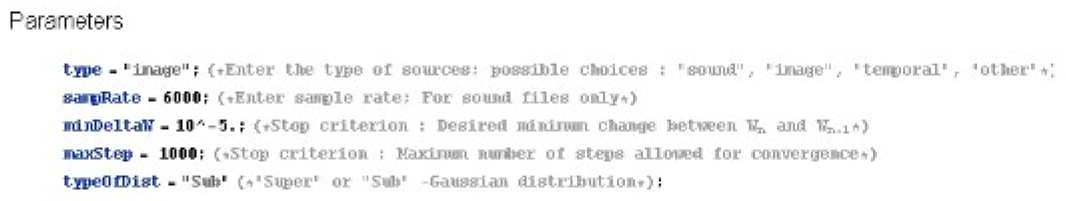
How to use the ICA packages

This section provides a quick overview of the InfoMax ICA package based on the maximum information perspective (InfoMax.nb; Amari et al., 1996), and on the FastICA package, based on the non-Gausianity perspective (FastICA.nb; Hyvärinen & Oja, 2000). Both packages have been implemented using Mathematica 7.0 and contain the same options with the exception of some parameters that are unique to a given algorithm. Each package consists of two main sections: Functions and Main. The Functions section contains the implementation of the algorithm and the necessary accompanying auxiliary functions. This section must be activated before ICA can be performed using the Main section. The Main section is divided into three cells: parameters, sources and ICA. The Parameters cell contains the information about the various parameters that need to be set prior to the analyses.

Parameters

First, the **type** of data must be specified in order to display the results properly (Figure 2). For example, if the data are in a sound file format (.wav), **type** must equal "sound" and **sampleRate** must be set to the correct desired frequency for the software to play the file correctly. Setting **type** to "image" allows for the use of usual image file formats (e.g., .jpg and .bmp). Since the analysis is only performed on greyscale images, any colour images will be automatically converted. If the data are time series, but not sound, then **type** must be set to "temporal" and a graph will depict the data using time as the independent variable. Finally, setting **type** to "other" is used for any data in a text format (e.g., .dat or .txt) (Each mix must be in a different file and arranged in a column). The next two options are about the convergence of the algorithm. First, **minDeltaW** controls the minimum difference between a given approximation W(t) and the next one W(t + 1). The lower the value, the better the estimation of the source will be. However, in some cases, the algorithm may not find a proper solution and, as a precaution, **maxStep** will control the maximum number of allowed steps before it stops searching. Finally, for the InfoMax notebook only, the type of distribution of the sources (**typeOfDist**) must be given in advance for the algorithm to be able to find the correct solution. To this end, there are two possible distributions: sub-Gaussian ("Sub") and super-Gaussian ("Super").

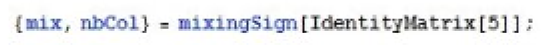
Figure 2. Screen capture featuring an example of the various parameters to be set before performing the analysis.



Parameters

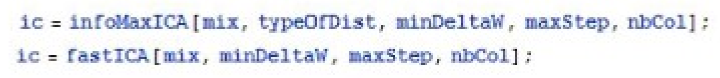
```
type = "image"; (*Enter the type of sources: possible choices : "sound", "image", "temporal", "other"*)
sampRate = 6000; (*Enter sample rate: For sound files only*)
minDeltaW = 10^-5.; (*Stop criterion : Desired minimum change between Wn and Wn+1*)
maxStep = 1000; (*Stop criterion : Maximum number of steps allowed for convergence*)
typeOfDist = "Sub" (*"Super" or "Sub" -Gaussian distribution*);
```

Figure 3. Example of five mixed signals to be loaded.



```
{mix, nbCol} = mixingSign[IdentityMatrix[5]];
```

Figure 4. Examples of "infoMaxICA" and "fastICA" functions to perform ICA.

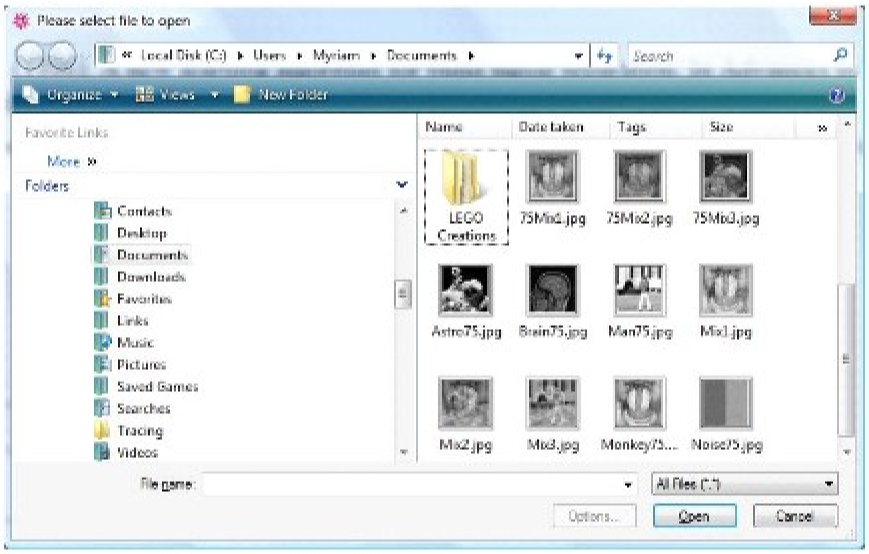


```
ic = infoMaxICA[mix, typeOfDist, minDeltaW, maxStep, nbCol];
ic = fastICA[mix, minDeltaW, maxStep, nbCol];
```

Figure 5. Syntax used to load three mixed sources (a) from a file selection window (b).

a)
```
{mix, nbCol} = mixingSign[IdentityMatrix[3]];
```

b)



Sources

The second cell must be activated to load the mixes. Two options are offered: mixed or unmixed sources. Mixed sources are obviously the ones that are most commonly encountered. In this case, the function mixingSign[ ] will need IdentiyMatrix[*m*] as an argument; where *m* is the number of sources (Figure 3).

If the sources are not mixes (e.g. to use the packages for illustration purposes), then the notebook will generate a random mixing matrix or alternatively the user can provide one. Finally, once activated, a window will appear requesting the location of each file. Once loaded, the sources will be displayed accompanied by correlation matrices.

Performing ICA

Finally, to perform the ICA, the function infoMaxICA[ ] or fastICA[ ] must be activated (Figure 4). Once the analysis is completed, the notebook will display the extracted sources as well as the correlation matrix of the extracted sources.

Example

In this example, Infomax and FastICA algorithms are used to extract the components from three mixes of images (provided in the supplemental materials). Also, for comparison, Principal Component Analysis (PCA) will be performed on the same mixes.

Infomax and FastICA

After the "Functions" section has been activated, the parameters were set as follows:

- type = "image"

- sampRate non-applicable in this case

- minDeltaW = 10^-5

- maxStep = 2500 for Infomax

- maxStep = 100 for FastICA (i.e. 100 for each component)

- For InfoMax, typeOfDist = "Sub" and "Super". Since no information about the underlying distribution was available, both types were tried.

Once the parameters are set, three "image" mixed sources were loaded. To that end, IdentityMatrix[3] was used as an argument for the function mixingSign[ ] (Figure 5).

Once the images are loaded, the notebook illustrates the loaded data (Figure 6). In this example, since the signals are already mixes, both the original and mixed signals are the same.

The ICA is then performed (Figure 7). The output of the