Introduction

Overview : What is Blind Source Separation (BSS)?

Blind Source Separation is a set of various methods in multivariate data analysis used to recompose unknown original sources from known data. These methods utilize information from the known observed sources with consideration of a set of unknown weights, which are then applied to the known observations to recover the original sources desired.

In other words, there are always two unknowns and one known (each source is its own vector within the matrix).

On a fundamental level, BSS attempts to solve a system of linear equations in which:

$$x = A \cdot s$$

Where the solution to this system will be in the form:

$$s = A^{-1} \cdot x$$

However, these s_i are not known values but instead a vector of distinct source signals, defined by their relationship to an also unknown operator matrix A. The x_i are values that can be directly observed from the signal/data without decomposition.

In general the basic components are:

- N unknown sources s_j (eg. the original sources to recover)
- One unknown operator **A** (an "un-mixing" or "de-mixing" matrix)
- P observed signals x_i with some relation:

$$x = A(s)$$

With this, the chosen BSS method takes the P known observations, performs a separation technique (also known as un-mixing or de-mixing) and outputs the unknown N sources that have been obscured by some unobserved process (i.e. the data collection process, random noise or limits of a set of data in most cases).

These y_k outputs should effectively estimate the original sources that make up the s vector(s).

BSS possesses a large range of applications but can effectively uncover factors in data that are previously unknown or underlying the known data through this demixing of source signals.

[@puigt2011]

ICA Generally

Definition: Independent Component Analysis (ICA)

Independent Component Analysis (ICA) provides a specific, efficient methodology to the general BSS framework described above and has been proven a useful application in various settings with complex data. This process can be referred to a generative model, as ICA determines how the observed data are constructed through the mixing process. [pg. 151, @Oja2001]

While ICA follows the same basic matrix model as described above, it can also often also be written as:

$$x = \sum_{i=1}^{n} a_i s_i$$

Assumptions of ICA Model

The two key characteristics and model requirements differentiate ICA from standard BSS and other methods are the following:

1. Assumption of statistical independence amongst the unknown source components

By definition, no component provides information on any of the other components.

[p 3, @Oja2001]

2. Assumption of Non-Gaussianity:

- These independent components are also assumed to have strictly Non-Gaussian distributions (or at most one independent component with a Gaussian distribution)
- Distinguishes from the similar structure method of Factor Analysis, which instead requires Gaussianity.

[p. 2, @Oja2000]

Note-

• The unknown mixing matrix will be denoted as a square for ease of calculation (not always the case)

In this scenario, the number of IC will equal the number of observed mixtures

[pg. 153, @Oja2001]

Ambigities of ICA Model

Because of the two unknown factors (source signals and mixing matrix) :

• No reliable method of calculating variance of ICs exists.

Solution : Set the RV to have unit variance or $E(s_i^2) = 1$

• No method of determining the order of ICs

[pg. 154, @Oja2001]

Primary Features of ICA (from Assumptions)

Independence

Since independence is a fundamental condition of ICA for the construction of ICs , it is important to define it conceptually for clarity.

Looking at two random variables (RV) denoted y_1 and y_2 in general, they will be independent if and only if their joint probability density function (pdf) can be rewritten as:

$$p(y_1, y_2) = p_1(y_1)p_2(y_2)$$

This can be extended to the expectation of two functions applied to these RV's and is denoted:

$$E(h_1(y_1)h_2(y_2)) = E(h_1(y_1))E(h_2(y_2))$$

[p. 3-4, @Oja2000]

Note on Correlation and Independence

By definition, two RV's are uncorrelated if and only if:

$$E(y_1 y_2) - E(y_1)E(y_2) = 0$$

While uncorrelatedness does not necessarily imply independence, independent RV's will always be uncorrelated. For ICA modeling purposes, this property helps simplify the estimation to only uncorrelated values.

[p. 5, @Oja2000]

Non-Gaussianity and Measurement

Though the Central Limit Theorem asserts that the sum of two or more independent RV's will tend towards a Gaussian distribution, the process of un-mixing actually produces non-Gaussian signals.

[@puigt2011]

ICA requires non-Gaussianity for the distributions of all source signals, as accurate estimation of the mixing matrix A is not even possible without meeting this condition.

To show this, assume that the known mixing matrix A is orthogonal (eg. $A^T A = AA^T = I$) and that all of the s_i are Gaussian.

Then, x_1 and x_2 have joint density:

$$p(x_1, x_2) = \frac{1}{2\pi} exp(\frac{x_1^2 + x_2^2}{2})$$

The symmetry of this joint density function makes it impossible to know the directional signals of the columns of matrix A, and therefore, the matrix cannot be estimated, rendering ICA ineffective in the Gaussian case (although one Gaussian IC is technically allowed).

[p. 5, @Oja2000]

Kurtosis as a Measure of Non-gaussianity

Kurtosis is defined as the 4th order cumulant:

$$kurt(y) = E(y^4) - 3(E(y^2))^2$$

Since y= 1, it can be simplified to:

$$kurt(y) = E(y^4) - 3$$

Which is equivalent to the 4th moment of y

In a Gaussian Distribution, the 4th moment equals $3(E(y^2))^2$

[p. 6, @Oja2000]

Thus,

$$kurt(y) = 3(E(y^2))^2 - 3(E(y^2))^2 = 0$$

All Gaussian distributions have kurtosis equal to 0, and therefore, it is expected that most non-Gaussian distributions will have a nonzero Kurtosis with some rare exceptions. Beyond this, Kurtosis essentially measures the deviation of RV from Gaussianity. [@puigt2011]

It's use in ICA is popular due to the ease of estimation as well as its linearity property. This holds two properties, assuming that x_1 and x_2 are independent:

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2)$$
$$kurt(\alpha x_1) = \alpha^4 kurt(x_1)$$

Issues with Kurtosis - Highly sensitive to outliers as just a few extreme values can dramatically impact its calculation. Therefore, its application to skewed data or data with outliers that strong influence may not be accurate.

[p. 7,@Oja2000]

Given this and the preference for another method in R packages to be utilized later, kurtosis will not be discussed much further in this paper, and more robust methods of measuring non-Gaussianity will be explored instead.

Negentropy

Since Kurtosis cannot always successfully determine Non-gaussianity, the use of negentropy as a alternative method of estimation and is typically preferred.

Negentropy derives from the theoretical "differential" entropy or level of randomness of a given variable. In general, Gaussian variables tend to possess the largest entropy given control for an equal variance condition. [8, @Oja2000]

Theoretical entropy is defined for discrete and continuous respectively as:

$$H(Y) = -\sum_{i=1} P(Y = a_i) log P(Y = a_i)$$

$$H(\mathbf{y}) = -\int f(\mathbf{y}) log f(\mathbf{y}) dy$$

To obtain the desired negative differential entropy, the calculation modifies to:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

Critically in the above equation, $\mathbf{y}_{gauss}$ is the Gaussian RV of with identical covariance as $\mathbf{y}$.

It is always:

• Non-negative and will be zero if and only if $\mathbf{y}$ is Gaussian.

• Invariant for invertible linear transformations.

In most cases, negentropy tends to be the best performing estimator of non-gaussianity but can be difficult or expensive to compute. However, the ease of computation has improved over time with technology, and thus the fastICA package relies on the estimator in its calculation.

[p. 8, @Oja2000]

Minimization of Mutual Information

Though negentropy will be primarily used in subsequent examples and applications in this paper through fastICA, other forms of ICA estimation are still important to note for thoroughness purposes. A third commonly used estimator is the minimization of mutual information technique.

Mutual Information - The MI between m scalar variables $y_i, i = 1, ..m$ is indicated by:

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

This can be rewritten as the following to descibe its relationship to negentropy:

$$I(y_1, y_2, ..., y_m) = C - \sum_{i=1} J(y_i)$$

MI measures the dependence between RV's and will be non-negative and zero if independent. Critically, this allows it to be used as a criteria for ICA transformation by minimizing mutual information of transformed components. ICA can be viewed as the minimization of mutual information.

[p. 9-10, @Oja2000]

Maximum Likelihood Estimation (MLE)

MLE serves as a similar estimation approach to the mutual information method for the ICA model.

Particularly, the likelihood of the noise-free model can be formulated then used to find an estimate with MLE.

Log-likelihood for ICA:

Denote:

$$\mathbf{W} = (\mathbf{w}_1, ...\mathbf{w}_n) = \mathbf{A}^{-1}$$

$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} log f_j(\mathbf{w}_i^T \mathbf{x}(t)) + T log|det\mathbf{W}|$$

Further simplified, this becomes:

$$\frac{1}{T} log L(\mathbf{W}) = E\{\sum_{i=1}^{n} log p_i w_i^T x\}) + log|det\mathbf{W}|$$

[p. 10 @Oja2000]

In this, the f_i signify the density functions of the s_i, which would generally be unknown but are known here.

$$\mathbf{x}(t), t = 1, ..., N$$

are the observations of x

$$log|det\mathbf{W}|$$

is linear transformation of an RV and its density

[p. 11 @Oja2000]

Connect to Infomax

A similar principle that has been used to develop ICA algorithms previously is Infomax (eg. the maximization of output entropy).

Given a neural network with :

$$y_i = \phi_i(\mathbf{w}_i^T x) + \mathbf{n}$$

Apply some function *H()* and maximize:

$$H(\mathbf{y}) = H(\phi_1(\mathbf{w}_1^T x), ..., \phi_n(\mathbf{w}_n^T x))$$

[pg. 211,@Oja2001]

Maximization of mutual information will be equal to maximization of output entropy, for which the transformation can be written as:

$$H(\phi_1(\mathbf{w}_1^T x), ..., \phi_n(\mathbf{w}_n^T x)) = H(x) + E\{log|det\frac{\partial F}{\partial W}(x)|\}$$

Where:

$$\mathbf{F}(x) = (\phi_1(\mathbf{b}_1^T x), ..., \phi_n(\mathbf{b}_n^T x))$$

And therefore:

$$E\{log|det\frac{\partial F}{\partial W}(x)|\} = \sum_i E\{log\phi_i'\mathbf{w}_i^T x\} + log|det\mathbf{W}|$$

Thus, output entropy equals likelihood and that infomax and MLE are effectively equal methods.

[pg. 213, @Oja2001]

ICA and Projection Pursuit

It is important to note the relationship between ICA and Projection Pursuit due to some significant similarities in their processes. Particularly, projection pursuit also looks at multidimensional data with the intent to find "interesting" projections. Much of its key applications are in the data exploration phase, but often projection pursuit directions can successfully visualize nuanced distribution of the data.

Beyond this, projection pursuit's goal of finding interesting data projections ties it directly to non-Gaussian distributions and how they can produce powerful visualizations of clustered data. Thus, most projection pursuit projections will be performed by determining the most non-Gaussian, and therefore, ICA components

can be considered projection pursuit *indicies* However, projection pursuit does not follow a set model with rigorous definition like ICA.

If the specified ICA model holds, then non-Gaussainity optimization will generate independent components. If not, the result model can be viewed as a set of projection pursuit directions.

[p 197-198, @Oja2001]

Pre-proccessing in ICA

Centering

Practically speaking, subtracting the expectation/mean vector $\mathbf{m} = E(\mathbf{x})$ will *center* the observations and make x zero-mean.

Centering data simplifies algorithmic processes and calculations.

[p. 12, @Oja2000]

Whitening

As a concept, whitening serves to take the original data and generate vectors with components that are both uncorrelated and equal variance with theoretical notation for the covariance matrix of a whiten vector as stated below:

$$E(\tilde{x}\tilde{x}^T) = \mathbf{I}$$

[p. 12, @Oja2000]

Whitening, also known as *Sphering*, removes the scale and correlation structure from the $\mathbf{X}$ dat matrix. In general, the process has been criticized for manipulating the distribution too far but still remains standard practice for pre-processing data for ICA. [p. 5,@izenman2003]

fastICA and Other Methods

Quick Look at fastICA

The next chapter will dig deeper in demonstrating the power of the fastICA R package in generating ICA models, but the key features should be pointed out before their application to two toy examples later.

Key Details

fastICA takes the data matrix X of the model and attempts to un-mix components. Explicitly, this matrix is a linear combination of matrices S (original source signals) and A (mixing matrix) by generating matrix W that maximized non-Gaussianity, which is $\mathbf{WX = S}$

This non-Gaussian approximation is based on negentropy calculations discussed in prior sections, and it is used over kurtosis because of efficiency according to the authors of the package.

Specfically for the fastICA algorithm, the $H(\mathbf{y})$ function options are:

•$G(u) = \frac{1}{\alpha} logcosh(\alpha u)$

•$G(u) = -exp(u^2/2)$

[see @fastICA]

fastICA Algorithm

"Basic" - One Component/Unit Alogrithm

1. Choose weight vector $\mathbf{w}$
2. Let $\mathbf{w}^+$ =E{$\mathbf{w}$g($\mathbf{w}$ˆT$\mathbf{x}$)} -E{$\mathbf{w}$g'($\mathbf{w}$ˆT$\mathbf{x}$)}$\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+/||\mathbf{w}^+||$
4. Repeat until converged (eg. old and new values of vector are same direction)

[p 14,@Oja2000]

Several Units

When determining multiple independent components, the vector $\mathbf{w}$ becomes a matrix $\mathbf{W}$ composed of vectors ($\mathbf{w}_1$,. . . ,$\mathbf{w}_n$)

Outputs must be decorrelated in this scenario, else they may converge to the same value. This is done by iteratively subtracting $\mathbf{w}_{p+1}$ with each additional projection, then normalizing that same vector.

1. Let $\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}^T\mathbf{w}_j\mathbf{w}_j$
2. Let $\mathbf{w}_{p+1} = \mathbf{w}_{p+1}/\sqrt{\mathbf{w}_{p+1}^T\mathbf{w}_{p+1}}$

This can also be written as:

1. Let $\mathbf{W} = \mathbf{W}/\sqrt{||\mathbf{W}\mathbf{W}^T||}$
2. Let $\mathbf{W}$ = (3/2)$*$$\mathbf{W}$-(1/2)*$\mathbf{W}\mathbf{W}^T\mathbf{W}$

[@Oja2000,p. 15]

Key Properties of fastICA

1. Cubic/quadratic convergence leads to faster convergences than linear methods
2. No step size parameters
3. No estimate distribution needs to be chosen
4. Can choose suitable non-linearity
5. Components can be done one by one, reducing computational expense
6. Neural algorithm: parallel, distributed, computationally simple, and requires little memory

[p. 16,@Oja2000]

Arguments in the fastICA function

fastICA (R implementation) employs several different technical arguments in its function that must be understood in order to interrupt the results of its output rigorously.

X simply the data matrix of interest

n.comp Number of components to be extracted

alg.typ :

1. if == "parallel" - components are extracted simultaneously(default)

2. if == "deflation" - components are extracted one at a time

fun Function form of G to use (see above)

alpha Constant range[1,2] used in approx. to neg-entropy when fun == "logcosh"

method :

1 if == "R" then computations are done exclusively in R

2. if == "C" then C code is used to perform computations (runs faster)

row.norm logical value indicating whether rows of X should be standardized

maxit Maximum number of iterations

tol A positive scalar giving tolerance at which the un-mix is considered to have converged

verbose level of output

w.init Initialized un-mixing matrix of dim(n.comp,n.comp) if Null matrix of RV's used

JADE and Infomax

Although fastICA has largely become the primary R packages for ICA modeling, JADE and Infomax methods are also available to perform the same task. In subsequent sections, their effectiveness will be evaluated in relation to fastICA.

JADE -

Uses *Joint Approximate Diagonalization of Eigenmatrices* (JADE) derived by Cardoso and Souloumiac in 1993. This approach involves finding a orthogonal rotation matrix R that diagonalizes the cumulant array of source signals. [@icaR]

Particularly, the original Cardoso paper describes the algorithm in 4 steps:

1 Create sample covariance $\hat{R}_x$ and compute whitening matrix $\hat{W}$

2 Create sample of 4th-order cumulants of whitening; compute significant eigenpairs

3 Jointly diagonialize the set with unitary matrix

4 Estimate A with $\hat{A} = \hat{W}\hat{U}$

[p.366,@cardoso93]

Infomax -

Also utilizes the orthogonal rotation matrix R that maximizes the joint entropy of a non-linear function of the estimated source signals and is derived from Bell and Sejnowski's 1995 paper *An information-matiximization approach to blind separation and blind deconvolution.* [@icaR]