New Convergence and Performance Measures for Blind Source Separation Algorithms

Amr Goneid
Computer Science & Engineering Dept., the American University in Cairo, Cairo, Egypt

Abeer Kamel and Ibrahim Farag
Faculty of Computers and Information, Cairo University, Cairo, Egypt

Abstract

Neural learning algorithms developed for blind separation of mixed source signals give rise to a Global Separating-Mixing (GSM) matrix that can be used to measure the performance of the unmixing system. In the case of the instantaneous linear noiseless mixing model, we consider the GSM as a transformation operator and show that it is equivalent to a combined stretching and rotation in the signal space. The extent of rotation is obtained using a polar decomposition method and can be taken as a measure of convergence to the problem solution. We also propose a new performance index (E3) that can be used to measure the performance of algorithms used in blind separation problems. The index E3 is more precise than the commonly used E1 and E2 indices and is normalized to the interval [0,1]. Experimentations using artificially generated supergaussian Laplacian signals have been performed using a fast ICA algorithm and considering a wide range of the number of mixed sources. Using the proposed E3 measure, we present experimental results on the dependence of algorithm performance on the number of mixed signals.

1. Introduction

Independent Component Analysis (ICA) has become an increasingly important tool for analyzing large data sets in search for patterns. ICA has been applied in a wide range of problems [1]. In particular, this method has demonstrated to be successful in various speech recognition problems [2], three dimensional (3-D) object recognition [3], natural images [4], unsupervised classification [5], bioinformatics [6], texture segmentation [7], electroencephalograms (EEG) [8], functional Magnetic Resonance Imaging (fMRI) [9] and face recognition [10]. In a large number of problems of such nature, the observed signals may be considered as results of linearly mixed instantaneous source signals. There is no prior knowledge about the linear generative model or the source signals except that they are statistically independent.

Given the signals $x_i(t)$ in discrete time $t$ from the sensor $(i),\ i = 1, ..., m$, the simplest case for ICA is the instantaneous linear noiseless mixing model given by:

$$X = A S$$

In this model, $X = [x_1(t), ..., x_m(t)]^T$ is a random vector of observations, $S = [s_1(t), ..., s_m(t)]^T$ is a random vector of hidden sources with mutually independent components, and $A$ is a nonsingular mixing matrix. Given $X$, the basic problem is to find an estimate $Y$ of $S$ and the
mixing matrix $A$. Normally, this model may be considered highly under-constrained since both $A$ and $S$ are unknown. For this reason, this type of problems is also called Blind Source Separation or BSS [1]. However, under certain statistical conditions, ICA can provide unique solutions.

If we define $W = A^{-1}$ as an unmixing matrix, ICA aims to estimate the unmixing matrix $W$ and thus to recover each hidden source using $S_k = W_k X$, where $W_k$ is the $k^{th}$ row of $W$.

In practice, an estimate of the unmixing matrix $W$ is not exactly the inverse of the mixing matrix $A$, and the matrix of estimated sources $Y = WX$ is only an approximation to $S$ [11]. In other words, BSS aims to find the matrix $W$ such that:

$$Y = WX = WA S \approx S$$

The linear mapping $W$ is such that the unmixed signals $Y$ are statistically independent. However, the sources are recovered only up to scaling and permutation.

The matrix $G = WA$ is usually called the Global Transfer Function or Global Separating-Mixing (GSM) Matrix. It can be used to estimate the performance of the unmixing system since its deviation from the identity matrix $I$ is a measure of the error in achieving complete separation of sources.

In the present work, we consider $G$ as a transformation operator and show that it is equivalent to a combined stretching and rotation in the signal space. The extent of rotation can therefore be taken as a measure of convergence to the problem solution. We also propose a new performance index that can be used to measure the performance of algorithms used in BSS problems.

The present paper is organized as follows: section 2 gives a brief view of the fundamental properties of the unmixing matrix and the basic features of ICA methods; section 3 introduces basic neural algorithms for obtaining estimates of the unmixing matrix. In section 4 we investigate the measures of convergence and performance of the algorithms and propose a new performance index. Finally in section 5 we present experimental results on the performance of a fast ICA algorithm using the newly proposed index.

2. Basic features of ICA methods

In contrast to correlation-based transformations such as Principal Component Analysis (PCA), ICA reduces the statistical dependencies of the signals, attempting to make the signals as independent as possible. This requires that not only the components of $Y$ are uncorrelated but also the components of $g(Y)$ should be uncorrelated, where $g$ is a suitable non-linear function. ICA methods estimate the unmixing matrix $W$ exploiting the non-Gaussianity of the signals. Non-Gaussianity is actually of paramount importance in ICA estimation, as without such signal property, the estimation is not possible at all. Therefore, it is not surprising that non-Gaussianity could be used as a leading principle in ICA estimation. In this case, we seek an estimate of $W$ that maximizes the non-Gaussianity of $Y$.

Such estimation of $W$ has several requirements and constraints. The first of these is that the number of observed signals cannot be fewer than the number of source signals. Therefore, we will assume that they are always equal to the same number $m$. Then, we also have to require that the sources are mutually statistically independent and non-Gaussian with the possible exception of at most one Gaussian source [12]. Also, the estimation of $A = W^{-1}$ and $S$ is up to a scale factor $C$ that could be either positive or negative. This requirement follows from the fact that if $C = diag (c_1, c_2, ..., c_m)$, $S' = CS$ and $A' = A C^{-1}$ then $A' S' = A C^{-1} C S = A I S = A S = X$. Hence, the mixed signals are invariant under scaling. Because of the rescaling property, we always assume that the source signals have unit variance so that $E(S S^T) = I$. 


In addition to such scale invariance, the estimation is up to a permutation of the order of the independent components and it is possible that they are not recovered in any particular order. In other words, perfect separation will result in \( G = W A \) to become a permutation matrix.

This leads to a measure of performance of the ICA methods defined as [13, 14]:

\[
E_1 = \sum_{i=1}^{m} \left( \sum_{j=1}^{m} \left| g_{ij} \right| \right) - 1 + \sum_{j=1}^{m} \left( \sum_{i=1}^{m} \left| g_{ij} \right| \right) - 1
\]

where \( g_{ij} \) is the \( ij \)th element of the matrix \( G \). When perfect signal separation is achieved, the matrix \( G \) becomes a permutation matrix where each row and each column has only one element equal to 1 and the rest are zeros. In this case, the performance index \( E_1 \) takes its minimum value of zero. In practice, it is a very small number.

A second index \( E_2 \) is sometimes used, and it is the same as \( E_1 \) except that the absolute values are replaced by squares. In both cases the higher the value the greater the deviation from perfect separation.

3. Neural learning algorithms for the unmixing matrix

There exist a variety of neural algorithms that attempt to estimate the unmixing matrix through some process of unsupervised learning. Of these, two types of algorithms have been quite popular due to their high performance and computational simplicity. The first is Gradient Algorithms and the second is the Fixed-point Algorithms. Learning rules for the unmixing matrix can be derived by maximizing negentropy or by minimizing mutual information, which is also equivalent to the maximization of likelihood (for an extensive review, see [1]). In case of negentropy maximization, and for some non-quadratic function \( F \), an approximate expression for the negentropy is given by [1]:

\[
J(y) \approx \left( E\{F(y)\} - E\{F(v)\} \right)^2
\]

where \( v \) is a Gaussian variable. Forms of the non-linearity \( F \) which have proved very useful are:

\[
F(y) = \frac{1}{a} \log \cosh ay, \quad \text{or} \quad F(y) = -\exp(-\frac{y^2}{2})
\]

It is quit often in several ICA algorithms to preprocess the mixed signals by whitening or sphering using a linear transform (PCA like): \( Z = VX \) where \( V \) is a whitening matrix such that \( E\{ZZ^T\} = I \). The resulting unmixing matrix \( W \) can be orthogonalized to improve convergence. The total unmixing matrix is then \( W \leftarrow WV \) [15].

3.1 Gradient algorithm

Various gradient-based learning rules have been derived [14,16,17]. Here, we briefly outline a version of this methodology based on maximizing negentropy.

Using whitened data \( Z \) and taking one component \( w \) (a vector representing one row from \( W \)), we seek a transformation \( w^Tz \) that maximizes negentropy \( z \) is one component of \( Z \). Taking into consideration the normalization \( E\{(w^Tz)^2\} = \|w\|^2 = 1 \), and setting \( y = w^Tz \), the gradient of the approximation of negentropy is

\[
\frac{\partial J(y)}{\partial w} = \frac{\partial}{\partial w} \left[ E\{F(w^Tz)\} - E\{F(v)\} \right]^2 = 2 \left[ E\{F(w^Tz)\} - E\{F(v)\} \right] E\{z\} f(w^Tz)
\]

Let \( \gamma = E\{F(w^Tz)\} - E\{F(v)\} \)

Then \( \Delta w \propto \gamma E\{z f(w^Tz)\} \)
with the normalization \( w \leftarrow w / \| w \| \)

The non-linearity function \( f \) which is the derivative of \( F \), can be taken as one of the following:

\[
f(y) = \tanh(ay), \quad \text{or} \quad f(y) = y \exp\left(-\frac{y^2}{2}\right) \quad \text{or} \quad f(y) = y^3
\]

It is to be noted that the above algorithm is very useful when one targets on-line BSS. In this case, the learning (adaptation) parameter \( \gamma \) can be used to update \( W \) using batches of on-line data, as e.g. in [17]

### 3.2 Fixed-point (FastICA) algorithm

The FastICA algorithm [18] is based on a fixed-point iteration method to maximize the negentropy \( J(w^T z) \) using a Newton iteration method. This approach proved to be much more superior to gradient-based algorithms with much faster convergence. A variant of the algorithm that estimates several or all components in parallel uses symmetric orthogonalization to prevent the different \( w \) vectors to converge to the same maxima. This process is achieved for the matrix \( W = (w_1, w_2, \ldots w_m)^T \) in every iteration by setting

\[
W \leftarrow (W W^T)^{-1/2} W
\]

The basic steps of the algorithm are as follows:

- Normalize the signals \( X \) to have zero means and unit variances.
- Whiten data using e.g. PCA estimation to get the data matrix \( Z = VX \), where \( V \) is the whitening matrix.
- Choose random initial orthonormal vectors \( w_i \) to form initial \( W \).
- Iterate:
  - Do symmetric orthogonalization of \( W \) as described above.
  - Test for convergence. If converged, break.
  - Set \( W_{old} \leftarrow W \)
  - For each component \( w_i \) of \( W \) set
    \[
    w_i \leftarrow E\{ z f(w_i^T z) \} - E\{ z f'(w_i^T z) \} w_i
    \]

- At convergence, the total unmixing matrix is obtained as \( W \leftarrow W V \).

The functions \( f' \) are the derivatives of \( f \) given before. For example for the \( \tanh \) function we have \( f'(y) = a [1 - \tanh^2(a y)] \), etc.

For the test of convergence, we require that the old and new \( W \) point in the same direction, i.e.,

\[
| \text{diag}(W^T W_{old}) | \approx 1.
\]

Unlike the gradient-based algorithm, the FastICA does not need adaptation, except perhaps in fine tuning using the non-linearity \( f(y) \). This feature makes the algorithm more effective in off-line applications.
4. Convergence and performance measures

In the present work, we re-examine two important measures needed for the convergence and the performance of adaptive or iterative algorithms for obtaining the unmixing matrix. We introduce new measures that, in our view, would enhance the process of convergence to the separation solution.

4.1 Convergence measure

The measure of convergence used in the FastICA algorithm is that in a given iteration, the old and new $W$ are found to point in the same direction, i.e.,

$$1 - |\text{diag}(W^T W_{\text{old}})| < \varepsilon$$

Since in a given iteration, the GSM $G = W A_o$, where $A_o$ is a constant initial value for the mixing matrix, the above criterion also holds for $G$. Notice that the departure of $G$ from the identity matrix is a combined effect of rescaling, and permutation and rotation (with sign ambiguity).

We propose to separate these factors using a polar decomposition of the $G$ matrix in the form:

$$G = Q P$$

where $P$ is a positive definite matrix with $\det(P) = |\det(G)|$ and $Q$ is an orthogonal matrix with $Q^T = Q^T$, $Q^T Q = Q Q^T = I$ and $|\det(Q)| = +1$

The method for computing the polar decomposition is to use a Singular Value Decomposition (SVD) of $G$ in the form $G = U S V^T$ so that $P = V S V^T$ and $Q = U V^T$.

We may identify $P$ as a stretching matrix, whereas $Q$ acts as a rotational matrix. Therefore, the polar representation decomposes $G$ into separate stretching and rotation effects. The cosine of the rotation angle is to be found on the diagonal of $Q$. Hence, we may introduce a convergence criterion as:

$$\Delta |\text{diag}(Q)|_{\min} < \varepsilon$$

where $\varepsilon$ is a threshold value. The above condition also applies for $W$.

4.2 Performance index

The performance index $E_1$ is supposed to measure the sum of absolute deviations of the $G$ matrix elements $g_{ij}$ from a pure permutation matrix where each row and each column has only one 1 (or -1) in it. The performance index $E_2$ is the same except that it measures the sum of Euclidean distances. If $M_i = \max_k |g_{ik}|$ and $M_j = \max_k |g_{kj}|$ then a close examination of $E_1$ shows that it underestimates the sum of absolute deviations in each row sum by $|M_i - 1| / M_i$ and in each column sum by $|M_j - 1| / M_j$. Although the missing quantities become zero for perfect separation, their values are not necessarily zero during the iteration process. Moreover, $E_1$ is not normalized to $\{0, 1\}$ and its value in the worst case (matrix of all 1’s) is $2m$ (m-1), where $m$ is the matrix dimension. Similar arguments apply to $E_2$.

In order to include these contributions, we are proposing a new performance index $E_3$ in the following form:
\[ e_3 = \sum_{i=1}^{m} \left( \sum_{j=1}^{m} |g_{ij} - M_i + |M_i - 1| \right) + \sum_{j=1}^{m} \left( \sum_{i=1}^{m} |g_{ij} - M_j + |M_j - 1| \right) \]

and \[ E_3 = \frac{e_3}{2m(m-1)} \]

where \( E_3 \) is now normalized to \((0,1)\).

We have performed an experiment using 2000 random \( G \) matrices with dimensions ranging between 2x2 and 20x20 to measure the difference between \( E1 \) and our proposed \( E3 \). Figure (1) shows the ratio \( E1/E3 \) as a function of matrix dimension \( m \) as well as the ratio \( E1^*/E3 \) where \( E1^* \) is the \( E1 \) index normalized to \( (0,1) \).

![Figure (1): Comparison between performance measures \( E1 \) and \( E3 \).](image)

It can be seen from the figure that \( E1 \), even after normalization, underestimates the error at small dimensions \( m \) then overestimates the deviations for \( m > 5 \) due to the division by \( M_i \) and \( M_j \).

### 5. Experimental results using artificially generated signals

To test the present convergence and performance measures, we have conducted several experiments using artificially generated signals and a modified implementation of the FastICA algorithm using a MATLAB environment. The reason for using the FastICA algorithm is that it is considerably fast and that it allows computing all of the components in parallel using symmetric orthogonalization.

#### 5.1 Geometrical shapes as sources

To visualize the rotational effect of mixing and unmixing, we have used \( m = 2 \) sources representing some parametric geometrical shapes \( \rho (\theta) \) so that

\[ \mathbf{S}(t) = \begin{cases} s_1(t) = \rho(\theta) \cos(\theta) \\ s_2(t) = \rho(\theta) \sin(\theta) \end{cases} \]

\[ \theta = (2\pi / N)t, \quad t = 0,1,\ldots, N \]

Figure (2) shows two examples with parametric equations

- **Bifolium**: \( \rho(\theta) = \sin(\theta) \cos^2(\theta) \)
- **Rose**: \( \rho(\theta) = \cos(3\theta) \cos(\theta) \)
5.2 Supergaussian random signals

The greater bulk of artificial signals used in the present experimentation has been generated using Laplacian signals with probability density function (PDF) of

\[ p(x) = \frac{1}{2b} \exp\left( -\frac{|x - \mu|}{b} \right) \]

where \( b \) is the scale parameter and \( \mu \) is the location parameter. It should be noted that the Laplacian density is only a special case of the more general Exponential Power Distribution (EPD), also called the Generalized Gaussian Distribution [19].

\[ p(x) = \frac{1}{2b^\alpha \Gamma(1+1/\alpha)} \exp\left( -\frac{|x - \mu|^\alpha}{\alpha b^\alpha} \right) \]

The shape parameter \( \alpha \) determines the shape and hence the kurtosis of the distribution. By changing the shape parameter \( \alpha \), the exponential power distribution can describe both supergaussian (\( 0 < \alpha < 2 \)) and subgaussian (\( \alpha > 2 \)) densities. In particular, Laplacian densities have \( \alpha = 1 \), the normal distribution has \( \alpha = 2 \) and the uniform distribution has \( \alpha \to \infty \).

In the present experiments, random Laplacian variates have been generated using the following relation

\[ x = \mu - b \text{sgn}(r) \ln(1 - |2r|) \]

where \( r \) is a uniform random variate over the interval \{ -1/2 , +1/2 \}.

5.3 Rotational Effects of mixing and unmixing signals

As an illustration of the rotational effects of the mixing and unmixing signals, figure (2) shows the results obtained using the FastICA algorithm for three cases: (A) Two Laplacian signals with \( \mu = 0 \) and different scale parameters (\( b = 0.5 \) and \( b = 2.0 \)). The length of each signal is 5000 samples. (B) Two signals representing bifolium shape, each 1000 points in length (C) Two signals representing a rose shape, each 1000 points in length.
Figure (2): Rotations resulting from mixing and unmixing signals: (A) Two Laplacian signals, (B) Bifolium shape, (C) Rose shape (see text for signal parameters).

The first row of the figure shows the signals/shapes, the second row shows their density plots after mixing by some random mixing matrix, the third row shows intermediate rotations towards convergence and the fourth row shows the density plots of the recovered signals (at...
convergence). The rotational effects are quite clear from these examples. The last row shows the convergence profile where the angle of rotation as deduced from the polar decomposition of the $G$ matrix is shown as a function of iteration number. Convergence is to zero or $90^\circ$ due to permutation of recovered signals. These convergence profiles support our proposal of using the change in cosines of angles from the polar decomposition as a measure of convergence.

Figure (3) shows the convergence profiles of the signals of figure (2) as measured by the performance index $E3$. The figure shows the decrease in $E3$ by further iterations ($E3$ is already normalized to the interval $\{0, 1.0\}$). The figure also shows how fast convergence is in the case of supergaussian signals (signals (A)).

5.4 Effect of number of sources on algorithm performance

The index $E3$ has been used to measure the performance of the FastICA algorithm using the $tanh$ non-linearity in 25 experiments each representing a different number of signals $m$ ($m = 2,4,6 \ldots 50$). In each experiment, $m$ artificially generated Laplacian signals represented the sources. Each source was characterized by a location parameter of $\mu = 0$ and a randomly selected scale parameter $b$ in the range 0.2 to 2.2 with a length of 1000 random samples for each source. Each of the 25 experiments was repeated 20 times with a different random mixing matrix $A_0$ in each time in order to obtain an average of the results.

Figure (4) shows an example of $m = 6$ source signals. Signals like these were input as a source matrix $S$ to a preprocessing stage where it is normalized to unit variance followed by the mixing process where we obtain $X = A_0 S$. The mixed signals are then normalized to zero mean followed by the process of data whitening. Such process considerably improves the convergence and can be summarized in the following steps:

- Obtain the eigenvector matrix $E$ and eigenvalues diagonal matrix $D$ from the covariance matrix $C = \text{cov} (X)$.
- Whiten data to get the data matrix $Z = VX$, where $V$ is the whitening matrix and is computed as $D^{-1/2} E^T$.

Whitened data $Z$ are then input to the FastICA algorithm, and in each iteration ($k$) we do a polar decomposition of the GSM matrix $G_k = W_k A_0$ and compute the performance index $E3_k$. 

![Figure (3): Performance index $E3$ for the signals of figure (2).]
Figure (4): Example of 6 Laplacian source signals.

Figure (5) shows the results for the performance of the algorithm as a function of the number of sources. On the left, the performance index $E_3$ at iteration $k = 5$ is plotted against number of mixed sources. The figure shows that although the algorithm is quite fast, only below 5 mixed sources that the performance index is small enough ($< 0.04$) to indicate convergence after about 5 iterations. When the number of sources is $m > 30$, the error after 5 iterations is still high (about 0.12) and we expect the need for more iterations to reach convergence. This is evident from the figure on the right where the number of iterations needed for convergence at $E_3 = 0.03$ is plotted against the number of mixed sources. For $m < 30$, the number of iterations is probably linear in $m$. Above 30 sources, the profile is more complex and the number of needed iterations increases rapidly with the number of sources.
6. Conclusion

Neural learning algorithms developed for blind separation of mixed source signals give rise to a Global Separating-Mixing (GSM) matrix $G$ that can be used to measure the performance of the unmixing system. Considering $G$ as a transformation that involves both stretching and rotation, and using polar decomposition, it was possible to arrive at a convergence measure for iterative algorithms for blind source separation that depends on the change of angle of rotation. Such rotational convergence can also be measured using the unmixing matrix estimates deduced iteratively by the unmixing algorithm.

A re-examination of the performance measures commonly used at convergence shows that they underestimate the error at a given iteration and that they are not normalized. We have proposed a performance measure ($E3$) that is both normalized and more precise than the commonly used $E1$ and $E2$ measures. Such measure is dependent on the total absolute deviation from perfect separation, and allows for the permutation of the separated sources. The new performance measure proved to be able to express the dependence of convergence profile on the complexity of the input source mixture.

Figure (5): Performance of FastICA algorithm with a $tanh$ non-linearity as measured by the index $E3$. 
References: