New equations and iterative algorithm for blind separation of sources

Rubén Martín-Clemente*, José I. Acha

* Corresponding author. Dept. de Ingenieria Electronica, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain. Tel.: +34-95-448-7335. E-mail address: ruben@us.es (R. Martín-Clemente).

Abstract

This paper addresses the blind separation of non-gaussian sources for instantaneous mixtures by using higher-order statistics. It is proven that the estimation of the separating matrix can be achieved by solving linear equations repeatedly. The convergence of the algorithm is proven analytically. A study of the complexity of the algorithm is also presented. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Blind source separation; Independent component analysis; Higher-order statistics

1. Introduction

Since its appearance in the late 1980s, blind source separation (BSS) has become one of the most active areas of research in signal processing. BSS consists of recovering a set of unobserved signals (called sources) from a set of measured signals (called observations), each of which is assumed to be a linear combination of the sources. The term “Blind” stresses the fact that, by hypothesis, neither the source signals nor the mixing structure are assumed to be known a priori. The key point that makes the separation possible in most of the existing approaches is the assumption of mutual independence between the sources.

The BSS model is attractive in a great number of situations, e.g., when modeling the transfer function between the sources and sensors is difficult due to the lack of information. Applications naturally arise in diverse fields, including communications, array processing, speech processing (e.g., several competing speakers have to be extracted from multiple microphone measurements), etc.

In the past ten years, numerous solutions to this problem have been proposed, starting from the seminal work of Jutten and Herault [16]. In recent times, separation criteria that are based on entropy and mutual information have attracted a great deal of attention [1,2]. The properties of the vectorial spaces of sources and mixtures also lead to geometrical methods [22,23]. Other approaches are concerned with batch (off-line) algorithms based on higher-order statistics [8,9,13] (for further reading, see [6,11,12] and the references therein).

The paper is organized as follows: some background on the source separation problem is given in Section 1. Section 2 presents conditions for source separation, which lead to new equations for BSS. In
Section 3, a batch algorithm is presented. Section 4 analyzes the computational complexity of the algorithm. Links with other methods are presented in Section 5. Finally, numerical experiments illustrating the effectiveness of the approach are given in Section 6.

1.1. Problem statement and notation

The instantaneous, linear, time-invariant BSS model is represented by the equation

$$\mathbf{u}(t) = \mathbf{M}\mathbf{s}(t),$$

(1)

where \(\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T\) is a vector of \(N\) sources, \(\mathbf{M}\) is an unknown \(N \times N\) mixing matrix, which contains the mixture coefficients and vector \(\mathbf{u}(t)\) collects the \(N\) observations, being the only data available. Here, as in the following, \((\cdot)^T\) represents the transpose operator. The aim of BSS is to recover the original sources from the given observations. The following basic assumptions are made throughout the paper:

(A.1) The mixing matrix is invertible.

(A.1) The sources are zero-mean, stationary and statistically independent. In addition, at most one source is Gaussian distributed.

The assumption of independence between the sources is physically plausible when they have different origins. Moreover, the identifiability of the mixing matrix requires at most one source being Gaussian, as shown in [3,9] after a theorem independently devised by Darmois and Skitovich.

Note that the scale of the sources cannot be determined since differences between their powers can be included in the mixing matrix. Accordingly, we make the third assumption, which is not essential but simplifies our analysis:

(A.3) The sources are unit-variance signals.

Most of the existing approaches to BSS simplify the problem by means of the so-called whitening of the original observations \([8,7,9,13]\), i.e., the observed vector \(\mathbf{u}(t)\) is linearly transformed into a vector

$$\mathbf{x}(t) = \mathbf{W}\mathbf{u}(t) = \mathbf{W}\mathbf{M}\mathbf{s}(t) = \mathbf{A}\mathbf{s}(t)$$

(2)

which is spatially white, i.e., the signals in \(\mathbf{x}(t)\) are uncorrelated and have unit variance so that \(E[\mathbf{x}(t)\mathbf{x}^T(t)] = \mathbf{I}\), where \(E[\cdot]\) denotes the expectation operator and \(\mathbf{I}\) is the identity matrix. As a consequence, \(\mathbf{A} = (a_{ij})\) must necessarily be an orthogonal matrix, i.e., it verifies \(\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A} = \mathbf{I}\). The separation is achieved by determining an orthogonal \(N \times N\) separating matrix \(\mathbf{B} = (b_{ij})\) so that

$$\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t) = \mathbf{G}\mathbf{s}(t)$$

(3)

is an estimate of the source signals, i.e., the global transfer matrix \(\mathbf{G} = \mathbf{B}\mathbf{A} = (g_{ij})\) has one and only one non-zero coefficient per column and row, which is equal to \(\pm 1\). In this case, \(\mathbf{G}\) is called a generalized permutation matrix.

Whitening may not lead to statistically efficient BSS algorithms, as discussed in [4,6]. Nevertheless, it reduces the problem of estimating the inverse of an arbitrary mixing matrix to the simpler problem of finding a well-conditioned orthogonal separating matrix. In practice, spatial whiteness can be easily accomplished by classical principal component analysis (PCA) [9,24] and will not be discussed here.

2. Separation criterion: equations for BSS

We present a new set of equations in the coefficients of the separating matrix. These equations are based on some derivatives of the output cross-cumulants. As an example, we shall discuss briefly the two-source case. Some comments and discussion complete the section.

2.1. Differentiating the output cumulants

Thanks to the multi-linearity property of the cumulants, the output cross-cumulant \(\text{cum}_3(y_i(t), y_j(t)) = \text{cum}_3(y_i(t), y_i(t), y_j(t))\) can be expanded as follows:

$$\text{cum}_3(y_i(t), y_j(t)) = \sum_{p=1}^{N} g_{ijp} g_{jip} \kappa_p,$$

(4)

where \(\kappa_p\) stands for the kurtosis (fourth-order cumulant) of the \(p\)th source. Comon [9, Theorem 11] showed that the sources are separated if and only if the \(N\) components of \(\mathbf{y}(t)\) are pairwise independent. When this occurs, the cross-cumulants of any order between different outputs \(y_i(t)\) and \(y_j(t)\) are zero (see [20, p. 13]). This property is the starting point for several approaches. In practice, though, the problem is that (4) depends on as high as fourth-order powers of the coefficients of \(\mathbf{B}\), through the relation \(\mathbf{G} = \mathbf{B}\mathbf{A}\).
We propose to differentiate the cumulants, which enable us to reduce the higher power to lower powers. Consider the following second-order derivatives (notice that time index \( t \) is omitted in the following):

\[
\frac{1}{6} \frac{\partial^2 \text{cum}_{ij}(y_i, y_j)}{\partial b_{ij}^2} = \sum_{p=1}^{N} g_p g_j a_{jp}^2 \kappa_p \quad (i \neq j).
\]

(5)

In view of (5), one can state the following basic claims: by setting (5) to zero for all \( i \neq j \), we obtain a set of simultaneous equations which are satisfied when \( G \) is a generalized permutation matrix. This is easily proven by direct substitution in (5). Moreover, if \( \forall p \ (1 \leq p \leq N) \), none of the matrices \( A_p(A) \text{def} \ \text{diag}(a_{p1}^2 \kappa_1, \ldots, a_{pN}^2 \kappa_N) \) have repeated diagonal entries, then spurious roots are not possible. This is proven and discussed in Appendix A.

Nevertheless, such equations are only implicit since neither \( G \) nor the kurtoses of the sources are known. In order to express (5) in terms of the coefficients of matrix \( B \) and the joint cumulants of the observed signals, we use the multi-linearity property of the cumulants and obtain

\[
\psi_{ij}(x, B) \overset{\text{def}}{=} \sum_{p=1}^{N} g_p g_j a_{jp}^2 \kappa_p \\
= \sum_{n=1}^{N} \sum_{m=1}^{N} b_{n} b_{m} \text{cum}(x_n, x_m, x_j, x_j).
\]

(6)

2.2. The equations in matrix form

From (6), the \( N \times N \) matrix whose \((i, j)\)-component is equal to \( \psi_{ij}(x, B) \) can be expanded as

\[
J(x, B) \overset{\text{def}}{=} E[\text{cum}(x_1^3, \ldots, x_N^3)] - B \text{B}^T - 2B \text{diag}(b_{11}, \ldots, b_{NN}).
\]

(7)

According to the claims in the preceding section, if \( B \) is a separating matrix (i.e., \( G = BA \) is a generalized permutation matrix), then \( J(x, B) \) is diagonal,

\[
J(x, B) = \text{diag}(\tilde{\psi}_{11}, \ldots, \tilde{\psi}_{NN}),
\]

(8)

where \( \tilde{\psi}_{ii} \) is the value of \( \psi_{ii}(x, B) \) at separation and must be linked to \( B \) through (6). Then, using (7) and the fact that \( BB^T = I \), premultiplying both sides of Eq. (8) by matrix \( B \) and transposing the results, we obtain

\[
E[\text{cum}(x_1^3, \ldots, x_N^3)Bx^T]
= -(I + \text{diag}(\tilde{\psi}_{11}, \ldots, \tilde{\psi}_{NN}))B \\
= 2 \text{diag}(b_{11}, \ldots, b_{NN}).
\]

(9)

Now, let \( b_i^T \) denote the \( i \)th row of matrix \( B \) and \( e_i \) denote the \( i \)th canonical vector, i.e., the \( i \)th column of the identity matrix \( I \), where both \( b \) and \( e \) are column vectors. Then, it is readily found that (9) can be written row by row as

\[
P_i b_i = 2 b_i e_i
\]

for \( i = 1, \ldots, N \), where

\[
P_i = E[x x^T] - \alpha_i I
\]

being

\[
\alpha_i = 1 + \tilde{\psi}_{ii}.
\]

(11b)

Finally, by defining

\[
v_i = b_j/(2 b_i)
\]

we obtain that \( v_i \) satisfies

\[
P_i v_i = e_i
\]

(13)

for \( i = 1, \ldots, N \) at separation.

2.3. Properties and example

Observe that a true separating matrix \( B \) is just obtained by transferring the columns from matrix \( A \) to the rows in matrix \( B \). In view of the preceding results, one may wonder: (i) if each row of the true separating matrices corresponds to a vector \( v_i \) that, for some index \( i \), verifies (13) and (ii) if Eq. (13) is free of spurious non-separating solutions for any value of index \( i \). In answer to both questions, we can state the following properties:

**Property 1.** \( \forall i, j \ (1 \leq i, j \leq N) \), if \( b_j/(2 b_i) \overset{\text{def}}{=} v_i = A e_j/(2 a_{ij}) \), i.e., \( v_i \) is equal to the \( j \)th column of the mixing matrix \( A \) divided by \( 2 a_{ij} \) \((a_{ij} \neq 0) \), then it fulfills Eq. (13), i.e., \( P_i v_i = e_i \).

**Property 2.** For a given index \( i \), Eq. (13) may have spurious non-separating solutions that turn \( P_i \) into a
singular matrix through (11b). Consequently, they can be easily detected and eliminated.

Proof is given in Appendix B. In practice, the rows of the separating matrix can be obtained by solving Eq. (13) for \( i = 1, \ldots, N \). An instructive example is that of solving the equations in the two-source case. By setting \( i = 1 \), Eq. (13) reads

\[
(m_{40} - x_1) v_{11} + m_{31} v_{12} = 1, \tag{14a}
\]

\[
m_{31} v_{11} + (m_{22} - x_1) v_{12} = 0, \tag{14b}
\]

where \( m_{pq} \equiv E[x_1^p x_2^q] \) and \( v_{1k} \) stands for the \( k \)th component of vector \( v_1 \). If the sources were already separated (i.e., \( a_{12} = a_{21} = 0 \) or \( a_{11} = a_{22} = 0 \)), then \( m_{31} = 0 \) and Eq. (14) would give \( v_{12} = 0 \), the correct solution. Otherwise, Eq. (14a) leads to

\[
x_1 = m_{40} - 2 + 2 m_{31} v_{12}, \tag{15}
\]

where we have used \( v_{11} = \frac{1}{2} \), as follows from (12). Using (15) in (14b) we obtain a quadratic equation:

\[
\frac{m_{31}}{2} + (m_{22} - m_{40} + 2) v_{12} - 2 m_{31} v_{12}^2 = 0 \tag{16}
\]

which is solved by

\[
\begin{align*}
  v_{12}^a &= \frac{a_{21}}{2a_{11}}, \\
  v_{12}^b &= \frac{a_{22}}{2a_{12}}
\end{align*}
\]

as can be shown after some algebra. It is readily verified that both solutions lead to a separating vector \( b_1 = v_1/\|v_1\|_2 \) (i.e., one source is obtained as \( b_1^T x \)).

3. An algorithm for BSS

In this section, we concentrate on a block algorithm aimed at estimating the sources.

3.1. Description of the algorithm

Instead of solving the nonlinear equations (13) directly, it is possible to separate the sources through an iterative batch procedure by repetently estimating the parameter \( x_i \) in (11) and solving linear equations, which may be seen as a “trade-off” between more difficult computation and iteration. As a consequence, the algorithm has been named “LinSep” (Linear equations for Separation). To begin with, let us define

\[
\hat{P}_i = E[xx^T] - \hat{z}_i I,
\]

where \( \hat{z}_i \) is the running estimate of the parameter \( x_i \). A valid choice for \( \hat{z}_i \) is based on the stability criteria presented below. The algorithm returns a vector \( y(t) \) containing the separated sources and proceeds as follows:

Algorithm

Step 1. Let \( x \) be the whitened observation vector.

Step 2. For \( i = 1 \) to \( N - 1 \)

1. Obtain \( \hat{z}_i \) from data (see Section 3.3).

2. Estimate matrix \( \hat{P}_i \).

2.3 Solve the linear equations \( \hat{P}_i v_i = e_i \) for vector \( v_i \).

2.4 Let \( b_i = v_i/\|v_i\|_2 \), where \( b_i^T \) stands for the \( i \)th row of \( B \).

Step 3. Orthogonalize the rows of matrix \( B \).

Step 4. Let \( x = Bx \) be the new observation vector.

Step 5. If \( B \) is close enough to a generalized permutation matrix, output \( y = x \) and finish. Otherwise, go back to Step 2.

Notice that Step 3 prevents different rows of \( B \) from converging to the same column of \( A \). It also ensures that \( B \) is an orthogonal matrix, as required. In addition, observe that only \( N - 1 \) loops are required (i.e., \( i = 1, \ldots, N - 1 \)) in Step 2: since \( B \) is orthogonal, the \( N \)th row of \( B \) can be obtained from the others in Step 3.

One possibility for performing the orthogonalization is the well-known Gram–Schmidt algorithm (see [10, p. 231]; [21, p. 229]), which, for \( k > 1 \), performs the substitution \( b_k = b_k - \sum_{p=1}^{k-1} (b_k^T b_p) b_p \) and, then, divides \( b_k \) by \( \|b_k\|_2 \). This procedure is acceptable since it does not change much \( B \) if this matrix is already almost orthogonal.

3.2. Convergence analysis

In the fourth-step of the algorithm, the observation vector \( x \) is updated as follows:

\[
x_i \leftarrow b_i^T x = b_i^T A s,
\]

where \( x_i \) stands for the updated \( i \)th component of the vector and \( b_i^T \) is the \( i \)th row of \( B \). Notice that (18) defines the recursive law of the algorithm, since it explicitly states the relation between the new observation \( x_i \) and the source vector \( s \). In addition, the mixing matrix is updated as well: in view of (18), \( b_i^T A \) is the
ith row of the new mixing matrix. The separation is thus achieved if the updated mixing matrix tends to a generalized permutation matrix. Let \( a_{ij}^{(n)} \) denote the \((i, j)\)-entry of the mixing matrix at the beginning of the \(n\)th iteration of the algorithm. In Appendix C, it is shown that

\[
\frac{a_{ij}^{(n+1)}}{a_{jp}^{(n+1)}} = G_{iq}^{(n)} \frac{a_{iq}^{(n)}}{a_{ip}^{(n)}},
\]

where \( i, p, q \in \{1, \ldots, N\} \) and

\[
G_{qp}^{(n)} = \frac{1 + (a_{ip}^{(n)})^2 \kappa_p - \hat{z}_i^{(n)}}{1 + (a_{iq}^{(n)})^2 \kappa_q - \hat{z}_i^{(n)}},
\]

where \( \hat{z}_i^{(n)} \) is the parameter of the equations at the beginning of the \(n\)th iteration and \( \kappa_j \) is the kurtosis of the \(j\)th source. For future use, let \( \kappa_{x_i}^{(n)} \) be the kurtosis of the \(i\)th observation at the same time, i.e., \( \kappa_{x_i} = E[x_i^4] - 3 \) (the superindexes have been omitted for simplicity). Additionally, from the multi-linearity property, it readily follows that

\[
\kappa_{x_i}^{(n)} = \sum_{l=1}^{N} (a_{il}^{(n)})^4 \kappa_{j_l}.
\]

If we consider the contribution of the \(q\)th source to the \(i\)th observation, then a natural measure of the separation performance is the intersymbol interference (ISI), defined by

\[
\text{ISI}_q(n) = \frac{\sum_p (a_{ip}^{(n)})^2 - (a_{iq}^{(n)})^2}{(a_{iq}^{(n)})^2} = \frac{1 - (a_{iq}^{(n)})^2}{(a_{iq}^{(n)})^2},
\]

where the last equality follows from the fact that \( A \) is an orthogonal matrix. Clearly, \( \text{ISI}_q(n) \geq 0 \) \( \forall n \) where equality only occurs at separation. If \( |a_{iq}^{(n)}| \) were increased from iteration to iteration, i.e., \( |a_{iq}^{(n+1)}| > |a_{iq}^{(n)}| \) for all \( n \), then \( \text{ISI}_q(n) \) would be monotonically decreasing, i.e., \( \text{ISI}_q(n+1) < \text{ISI}_q(n) \). To this end, two cases are treated separately below:

1. **The Case in which \( \kappa_{x_i}^{(n)} > 0 \):**

   Since \( \kappa_{x_i}^{(n)} > 0 \), it follows from (21) that there must be at least one source whose kurtosis is non-negative. If \( q \) is an index which verifies

\[
1 + (a_{iq}^{(n)})^2 \kappa_q > 1 + (a_{ip}^{(n)})^2 \kappa_p,
\]

for all \( q \neq p \) where \( \kappa_q \geq 0 \) then, by taking

\[
\hat{z}_i^{(n+1)} > 1 + (a_{iq}^{(n)})^2 \kappa_q,
\]

i.e., \( \hat{z}_i^{(n)} \) positive and large enough, it follows that

\[
G_{qp}^{(n)} > 1 \Rightarrow \left| \frac{a_{iq}^{(n+1)}}{a_{ip}^{(n+1)}} \right| > \left| \frac{a_{iq}^{(n)}}{a_{ip}^{(n)}} \right| \Rightarrow \left| a_{iq}^{(n+1)} \right| > \left| a_{iq}^{(n)} \right|,
\]

where implication (i) follows from (19) and implication (ii) is deduced from the fact that \( \sum_j |a_{ij}^{(n+1)}|^2 = \sum_j |a_{ij}^{(n)}|^2 = 1 \).

In the \((n+1)\)th iteration, the inequalities

\[
\kappa_{x_i}^{(n+1)} > 0 \quad \text{and} \quad 1 + (a_{ip}^{(n+1)})^2 \kappa_q > 1 + (a_{ip}^{(n+1)})^2 \kappa_p
\]

still hold. This is deduced from (21), (22) and the last part of (24). Hence, the same argumentation can be repeated: by taking \( \hat{z}_i^{(n+1)} > 1 + (a_{ip}^{(n+1)})^2 \kappa_p \), it follows that \( |a_{iq}^{(n+2)}| > |a_{iq}^{(n+1)}| \) and so on.

**In conclusion**, if \( \hat{z}_i^{(n)} \) is positive and large enough, then \( |a_{iq}^{(n+1)}| > |a_{iq}^{(n)}| \) for all \( n \), as required. However, observe that \( G_{qp}^{(n)} \) would be close to one and the convergence would be slow if \( \hat{z}_i^{(n)} \) were too large.

2. **The Case in which \( \kappa_{x_i}^{(n)} < 0 \):**

   If \( q \) is an index which verifies

\[
1 + (a_{iq}^{(n)})^2 \kappa_q < 1 + (a_{ip}^{(n)})^2 \kappa_p
\]

for all \( q \neq j \), where \( \kappa_q < 0 \), then by taking \( \hat{z}_i^{(n+1)} < 1 + (a_{iq}^{(n)})^2 \kappa_q \) it again follows that \( |a_{iq}^{(n+1)}| < |a_{iq}^{(n)}| \) \( \forall n \). The proof is similar as that of the previous case.

**3.3. Choice of the parameter \( \hat{z}_i \)**

Using (19) in \( \sum_p (a_{ip}^{(n+1)})^2 = 1 \), we obtain that

\[
(a_{iq}^{(n+1)})^2 = \frac{(a_{iq}^{(n)})^2}{(a_{ip}^{(n)})^2 + \sum_{p \neq q} (a_{ip}^{(n)})^2 G_{qp}^{(n)}},
\]

We can assume without any loss of generality that \( G_{qp}^{(n)} > 1 \) \( \forall p, q, n \), as guaranteed by the arguments in the preceding section. Let \( G_q^{(n)} = \max_{p, p \neq q} G_{qp}^{(n)} \) and \( g_q^{(n)} = \min_{p, p \neq q} G_{qp}^{(n)} \). Then, it is readily obtained from
(26) that
\[
\frac{G_q^{(n)}(a_{iq}^{(n)})^2}{(G_q^{(n)} - 1)(a_{iq}^{(n)})^2 + 1} \geq (a_{iq}^{(n+1)})^2 \geq \frac{g_q^{(n)}(a_{iq}^{(n)})^2}{(g_q^{(n)} - 1)(a_{iq}^{(n)})^2 + 1}
\]  \hspace{1cm} (27a)

and
\[
\frac{\text{ISI}_q(n)}{G_q^{(n)}} \leq \frac{\text{ISI}_q(n + 1)}{G_q^{(n)}} \leq \frac{\text{ISI}_q(n)}{G_q^{(n)}}.
\]  \hspace{1cm} (27b)

In view of (27), the performance of the iteration depends on the value of $g_q^{(n)}$, if $g_q^{(n)} \gg 1$, which implies that $G_q^{(n)} \gg 1$ as well, the inequalities in (27a) and (27b) assert that $(a_{iq}^{(n+1)}) \equiv 1$ (while all other components in the same row and column vanish, as a consequence of the orthogonality of the mixing matrix) and $\text{ISI}_q(n + 1) \equiv 0$, as desired.

In practice, the following choice performs well, as will be shown via simulations:
\[
\hat{\chi}_i^{(n)} = 1 + \mu \kappa_i^{(n)},
\]  \hspace{1cm} (28)

where $\mu \geq 1$ to ensure that either $\hat{\chi}_i > 1 + a_{ij}^2 \kappa_j \forall j$ in Case 1 or $\hat{\chi}_i < 1 + a_{ij}^2 \kappa_j \forall j$ in Case 2. Typically, simulations suggest the choice $\mu \in [1, 1.5]$. Furthermore, in the general $N$-source case, it is straightforward to show that (15) is generalized as
\[
\chi_i = E[x_i^4] - 2 + 2 \sum_{j \neq i} E[x_i^2 x_j] v_{ij},
\]  \hspace{1cm} (29)

where $v_{ij}$ is the $j$th component of vector $v_i$. Close to separation, the updated observations are almost pairwise independent and, consequently, $E[x_i^4] \approx 0$. Therefore, the true value of $\chi_i$ is
\[
\chi_i \approx E[x_i^4] - 2 = 1 + \kappa_i
\]  \hspace{1cm} (30)

which is precisely the value of (28) when $\mu \rightarrow 1$.

For example, assume a mixture of three sources that are generated as the cube of gaussian variables. The mixing matrix was randomly chosen to be
\[
M = \begin{bmatrix}
1.61 & -1.76 & 0.09 \\
1.05 & 1.68 & 0.65 \\
0.43 & -0.42 & -0.67
\end{bmatrix}.
\]

The statistics were estimated by using 1000 samples of each observation and $\mu$ was set to one in (28). After the first iteration of the algorithm, the global transfer matrix $G = BA$ was equal to
\[
G = \begin{bmatrix}
-0.99 & 0.11 & 0.016 \\
0.10 & 0.99 & -0.29 \\
0.03 & -0.04 & -0.99
\end{bmatrix}
\]

which seems to be sufficient for most practical applications. This is only one favorable example, but it is useful to illustrate the potential performance of the algorithm.

4. Computational complexity

In a batch algorithm, the speed of convergence is not as important as the computational complexity, measured as the number of single floating point operations (flops).

The cost of estimating the matrices $E[xx^T x_i^2]$ is critical in developing a computationally efficient algorithm. Fortunately, it is not necessary to estimate these matrices from the data: using the multi-linearity property, they can be obtained, with little effort, from the corresponding matrices used in the previous iteration of the algorithm. As a significant consequence, we do not need to update the observations in order to estimate these matrices: the fourth step of the algorithm can be replaced with another one in which only the matrices $E[xx^T x_i^2]$ are updated by using the multi-linearity property. This results in a significant computational saving.

Regarding the first iteration, each matrix $\hat{P}_i$ contains about $N^2/2$ different statistics of the observations. To calculate any fourth-order moment from a batch of $T$ samples requires about $4T$ single floating point operations (specifically, $3T$ termwise products plus $T - 1$ additions). Since we have to estimate $N - 1$ matrices, the computational cost of estimating all of them is approximately equal to $N^2/2 \times 4T \times N = 2N^3T$ flops.

To solve a system of $N$ equations by means of the LU decomposition requires about $2N^3/3$ flops. The algorithm solves $N - 1$ systems of equations in each iteration and iterates typically about $N$ times. Consequently, to determine the separating matrix requires about $2N^3/3$ flops. The cost of the whitening
and orthogonalization processes are negligible in comparison.

Since \( T \gg N^2 \) for any reasonable situation, it follows that the main task is the computation of the statistics of the observations. In conclusion, the effort of solving the equations is negligible and in practice does not affect the efficiency of the algorithm, even taking into consideration that we have to iterate several times.

5. Links with other methods

Note that Section 2 introduces a system of second-degree equations for the mixing matrix. Mansour and Jutten [17] also estimated the mixing matrix in the two-source case by rooting a quadratic equation (see also Martin-Clemente and Acha [18]). In their approach, the separating matrix was constrained to have its diagonal entries equal to one. This convention does not lead to equivariant estimates [5] and, consequently, the accuracy of the solution depends greatly on the mixing matrix. We overcome this important drawback by assuming that both the mixing matrix and the separating matrix are orthogonal (well-conditioned) matrices.

Furthermore, an important algorithm known as joint approximate diagonalization of eigenmatrices (JADE) was proposed by Cardoso and Souloumiac [4]. In their approach, the separating matrix is estimated as the matrix which jointly diagonalizes a set of \( N \) different cumulant matrices. The diagonalization of several matrices introduces redundancy and a greater robustness to estimation errors when the corresponding eigenvectors are not well conditioned. It is possible to reformulate the second-degree equations presented in Section 2 as an eigenvector-calculation problem, which can be algebraically related to the theoretical work of Cardoso and Souloumiac. To be specific, (6) comes from a sufficient subset of those equations which are solved by JADE. Consequently: (i) a significant computational cost saving is expected in our approach and (ii) since JADE satisfies more statistical equations, it may also be numerically more robust. The experiments provide support for both intuitions.

Our main contribution is to show further the calculation of the mixing matrix by solving a set of linear equations: indeed, the solution of a set of \( N \) linear equations requires \( 1/N \) times the computational effort needed to diagonalize a \( N \times N \) matrix. Even though we propose to refine the solutions by solving several times the equations, a very efficient and competitive algorithm is obtained, as shown by the simulations.

6. Experiments

In this section, we provide some simulation results on the performance of the proposed algorithm.

6.1. Performance indexes

Moreau and Macchi [19] have proposed a performance index which makes it possible to determine the quality of the source separation algorithms. This index is calculated as

\[
p = \frac{1}{2} \left\{ \sum_{i=1}^{N} \left( \sum_{j=1}^{N} \frac{|g_{ij}|^2}{\max_k |g_{ik}|^2} - 1 \right) + \sum_{j=1}^{N} \left( \sum_{i=1}^{N} \frac{|g_{ij}|^2}{\max_k |g_{kj}|^2} - 1 \right) \right\},
\]

where \( g_{ij} \) is the \((i,j)\)-entry of the global transfer matrix \( G \). The lower the value of the index, the better the performance and the separation is achieved when \( p = 0 \). The main advantage of this index is that it does not require matrix \( G \) to be reordered or normalized.

6.2. Experiment 1

We applied our algorithm to four signals in order to visualize and check our results. Different distributions, uniform, binary (negative kurtosis), exponential and the cube of a gaussian variable (positive kurtosis) were used to generate each source. Only the mixed signals, depicted in Fig. 1, were observed and used to estimate the sources, depicted in Fig. 2 (for clarity, only the first 100 samples were plotted for each signal). Observe that the sources are not zero-mean unit-variance signals. Therefore, our algorithm must subtract the mean and uncorrelate all the observations in a preprocessing step. The mixing matrix was
Fig. 1. Observations.

Fig. 2. Original sources.
randomly chosen as

\[
M = \begin{bmatrix}
-1.475 & 1.432 & -1.050 & -0.246 \\
0.602 & -0.190 & 0.162 & -0.290 \\
0.418 & 1.111 & -0.170 & -2.100 \\
1.315 & 0.190 & -1.990 & 1.349
\end{bmatrix}.
\]

This matrix is almost singular, since it has a very high condition number (623.8).

We used 1000 samples of each observation in order to estimate the matrices \(P_i\). After five iterations, the global transfer matrix was equal to

\[
G = \begin{bmatrix}
-0.0706 & 0.0079 & -0.2565 & -0.1006 \\
-0.0046 & 0.0008 & -0.0196 & -2.0013 \\
-0.1176 & 0.9720 & 0.0064 & 0.0277 \\
3.4534 & 0.0498 & -0.0007 & -0.1629
\end{bmatrix}.
\]

The estimated source signals are shown in Fig. 3. Note that it is impossible for the algorithm to distinguish between \(s_i(t)\) and \(-s_i(t)\). Moreover, the order of the sources has been permuted.

6.3. Experiment 2

For comparison with existing batch algorithms, we used a mixture of ten sources: five of them were uniform variables and the other five were generated as the cube of gaussian variables (5000 samples each). The mixing matrix was randomly chosen and then estimated by means of three different algorithms: LinSep, JADE [8] and FastIca [13]. For this task, standard implementations of JADE and FastIca were obtained from their authors via Internet [14,15]. In our experiments, we have used these programs as they were obtained, without any modification. The following parameters were measured:

- **The computational complexity.** Measured by means of the “flop” command of MATLAB. By convention, the complexity of LinSep is taken as unity. Hence, Table 1 really shows the quotient: flops-of-the-algorithm/flops-of-LinSep.
- **The mean signal-to-noise ratio (SNR) after the separation.**
- **The performance index,** defined above.
The results of 20 independent experiments were averaged and are shown in Table 1. Each entry in the table consists of two numbers: the first one is the mean value of the result, the second one, between brackets, is the standard deviation.

This is not a formal comparison between the algorithms, which is beyond the scope of this paper. However, the experiment shows that solving the BSS problem by means of linear equations leads to competitive algorithms.

It is noteworthy that the estimation of the matrices \( \mathbf{P}_i \) involves about 85% of the computational burden of LinSep.

6.4. Experiment 3

We prepared an experiment to study the convergence properties and speed of the algorithm. Thirty independent uniform sources were first mixed by a randomly chosen mixing matrix and then separated by our algorithm, which used 5000 samples of each observation to estimate the matrices \( \mathbf{P}_i \). The parameter \( \mu \) was set to 1.5. The experiment was repeated 20 times and the averaged results are presented. It is depicted in Fig. 4 the evolution of the performance index during the process. It shows that the algorithm usually needs to iterate of the order of \( N \) times. It is noteworthy that not all the existing algorithms are able to handle such a great a number of sources in practice.

7. Conclusions

The present paper uses the second-order derivatives of some cross-cumulants in order to present a new algorithm for BSS. First, Section 2 presents theoretical results in order to describe the separating matrices. The simplification of the cumulant-based equations by differentiating the cumulants seems to be a general-purpose procedure. We are currently studying its applicability to the problem of separating convolutive mixtures.

![Evolution of the Performance Index](image-url)
Secondly, an algorithm is presented in Section 3. Several approaches are possible, although the one presented in the paper, LinSep, is the most accurate of the different algorithms that we have implemented. It repeatedly solves a set of linear equations. However, this has little effect on the complexity of the algorithm. Instead, most of the computational effort is due to the estimation of the statistics of the observations. Hence, it would be interesting to reduce the set of statistics required to separate the sources. The work by Zhu et al. [25] serves as a guideline. Simulations show that this has little effect on the complexity of the algorithm.

Acknowledgements

This work was supported by the CICYT (Spain) under project TIC96-0500-C10-08.

The authors are indebted to the anonymous reviewers whose constructive comments helped us to improve the first version of this paper. We are also very grateful to Prof. C.G. Puntonet (University of Granada) for his fruitful discussions.

Appendix A

Let us reproduce Eq. (6) for convenience of presentation

$$\psi_{ij}(x, B) = \sum_{n=1}^{N} g_{jn} g_{jn} a_{jn}^2 \kappa_n.$$  \hspace{1cm} (A.1)

Observe that (A.1) can be written compactly in a matrix form

$$\psi_{ij}(x, B) = g_j^T A_j(A) g_i,$$  \hspace{1cm} (A.2)

where $g_j^T$ is the $k$th row of the global matrix $G$, i.e., $g_j^T = [g_{k1}, \ldots, g_{kN}]$ and $A_j(A)$ is the diagonal matrix whose $(i, l)$-entry is equal to $a_{jl}^2 \kappa_l$. Moreover, observe that $g_j^T g_i = \delta_{ij}$, where $\delta_{ij}$ stands for the Kronecker delta, since $G$ is an orthogonal matrix.

If $B$ is a separating matrix, then each vector $g_i$ is a different canonical vector. Therefore, it follows from (A.2) that $\psi_{ij}(x, B) = 0$ for all $i, j$ ($i \neq j$).

Now, we prove that if (H.1) $\forall \ p \ (1 \leq p \leq N)$, none of the matrices $A_p(A)$ have repeated diagonal entries then $\psi_{ij}(x, B) = 0$ for all $i \neq j$ only if $B$ is a separating matrix.

Note that hypothesis (H.1) implies that each eigenvalue of $A_j(A)$ is different.

The vector $A_j(A) g_j$ can be expressed as a linear combination of the vectors $g_i$ since the set \{ $g_1, g_2, \ldots, g_N$ \} forms an orthonormal basis of the space. Therefore,

$$A_j(A) g_j = \psi_{jj}(x, B) g_j + \sum_{i \neq j} \psi_{ij}(x, B) g_i,$$  \hspace{1cm} (A.3)

where $\psi_{ij}(x, B) = g_j^T A_j(A) g_i$ as was defined above. Since, $\psi_{ij}(x, B) = 0$ for all $i \neq j$, we obtain that $A_j(A) g_j = \psi_{jj}(x, B) g_j$, which implies that each $g_j$ is an eigenvector of a diagonal matrix whose eigenvalues are not repeated, i.e., each $g_j$ is a canonical vector. In addition, all the vectors $g_j$ $(j = 1, \ldots, N)$ are different, since $G$ is an orthogonal matrix. As a consequence, $B$ is a separating matrix.

Some comments must be made about condition (H.1): it cannot be verified a priori (when $B$ is unknown), only a posteriori (e.g., one may confirm that $P_i$ is invertible, see Property 2 in Section 2.3 and Appendix B). In any case, it is always possible to find an orthogonal matrix $U$ so that each eigenvalue of $A_j(UA)$ is different, under the condition that at most one of the kurtoses of the sources vanishes. Therefore, if (H.1) was not satisfied then, by multiplying $x$ with any orthogonal matrix $U$, the new mixing matrix $UA$ got in this way may satisfy (H.1).

However, condition (H.1) cannot hold if more than one kurtosis of the sources vanishes. This difficulty can be overcome by resorting to higher-than-fourth order cumulants, e.g., the sixth-order cross-cumulant

$$\text{cum}_{ij}(y_i, y_j) = \sum_{n=1}^{N} g_{jn} g_{jn} \gamma_n,$$

where $\gamma_n$ is the sixth-order cumulant of the $q$th source, can be differentiated four times to obtain an expression similar to (5), leading us to generalize the equations, etc. Interestingly, second-order statistics are...
Appendix B

B.1. Proof of Property 1

First, by definition (6):

$$\psi_{ii}(\mathbf{x}, \mathbf{B}) = \sum_{n=1}^{N} g_{in}^2 a_{in}^2 \kappa_n. \quad (B.1)$$

As a consequence, if \( \mathbf{G} \) is a generalized permutation matrix (see Section 1), then

$$\psi_{ii}(\mathbf{x}, \mathbf{B}) \in \{a_{i1}^2 \kappa_1, \ldots, a_{iN}^2 \kappa_N\}. \quad (B.2)$$

Secondly, matrix \( \mathbf{P}_i \) can be rewritten as

$$\mathbf{P}_i = \mathbf{A} \{E[\mathbf{s}s^T x_i^2] - z_i \mathbf{I}\} \mathbf{A}^T. \quad (B.3)$$

This follows from (11a) and the fact that \( \mathbf{AA}^T = \mathbf{I} \).

By definition (11b), \( z_i \) equals \( 1 + \hat{\psi}_{ii} \), where \( \hat{\psi}_{ii} \) can be any of the values of \( \psi_{ii}(\mathbf{x}, \mathbf{B}) \) at separation. Thus, from (B.2),

$$z_i \in S_{x_i} = \{1 + a_{i1}^2 \kappa_1, \ldots, 1 + a_{iN}^2 \kappa_N\}. \quad (B.4)$$

Note that each generalized permutation matrix \( \mathbf{G} \) is associated with one element in \( S_{x_i} \). By using (B.3), Eq. (13) leads to

$$\mathbf{P}_i \mathbf{v}_i = \mathbf{e}_i \Rightarrow \{E[\mathbf{s}s^T x_i^2] - z_i \mathbf{I}\} \mathbf{A}^T \mathbf{v}_i = \mathbf{a}_i, \quad (B.5)$$

where the column vector \( \mathbf{a}_i = \mathbf{A}^T \mathbf{e}_i \) stands for the \( i \)th transposed row of matrix \( \mathbf{A} \). Some algebra shows that

$$E[\mathbf{s}s^T x_i^2] = \mathbf{I} + A_i(\mathbf{A}) + 2 \mathbf{a}_i \mathbf{a}_i^T, \quad (B.6)$$

where \( A_i(\mathbf{A}) = \text{diag}(a_{i1}^2 \kappa_1, \ldots, a_{iN}^2 \kappa_N) \). Substituting (B.6) in the right-hand side of (B.5) gives

$$\{\mathbf{I} + A_i(\mathbf{A}) + 2 \mathbf{a}_i \mathbf{a}_i^T - z_i \mathbf{I}\} \mathbf{A}^T \mathbf{v}_i = \mathbf{a}_i. \quad (B.7)$$

Since \( z_i = 1 + a_{ij}^2 \kappa_j \) \((j \in 1, \ldots, N)\), the \((j, j)\)-entry of matrix \( \mathbf{I} + A_i(\mathbf{A}) - z_i \mathbf{I} \) is null. Thus, the \( j \)th column of the matrix \( E[\mathbf{s}s^T x_i^2] - z_i \mathbf{I} \) equals \( 2a_{ij} \mathbf{a}_i \). Consequently, by substituting \( \mathbf{v}_i = \mathbf{A} \mathbf{e}_j/(2 a_{ij}) \) in (B.7), the equation is satisfied. \( \square \)

B.2. Proof of Property 2

Assume that hypothesis (H.1) is not verified. If \( z_i \) were equal to any of the repeated diagonal entries of \( A_i(\mathbf{A}) \), then several diagonal components of \( \mathbf{I} + A_i(\mathbf{A}) - z_i \mathbf{I} \) would be equal to zero. Therefore, the columns of matrix \( \mathbf{I} + A_i(\mathbf{A}) - z_i \mathbf{I} + 2 \mathbf{a}_i \mathbf{a}_i^T \) could be linearly dependent and matrix \( \mathbf{P}_i \) would be singular (conversely, it can be shown that if (H.1) is satisfied, then \( \mathbf{P}_i \) is invertible at separation). In this case, it can be shown that (B.7) could be solved by some linear combinations of the columns of \( \mathbf{A} \) (i.e., \( \mathbf{v}_i = \mathbf{A} \sum_{j \in S} \hat{\beta}_j \mathbf{e}_j \), where index \( j \) is in \( S \) if \( a_{ij}^2 \kappa_j \) is one of the repeated diagonal entries of \( A_i(\mathbf{A}) \)). \( \square \)

Appendix C

Matrix \( \hat{\mathbf{P}}_i \) can be rewritten as

$$\hat{\mathbf{P}}_i = \mathbf{A} \{E_i + 2 \mathbf{a}_i \mathbf{a}_i^T\} \mathbf{A}^T, \quad (C.1)$$

where \( E_i \) is the diagonal matrix defined as \( E_i \equiv \mathbf{I} + A_i(\mathbf{A}) - z_i \mathbf{I} \), where \( A_i(\mathbf{A}) = \text{diag}(a_{i1}^2 \kappa_1, \ldots, a_{iN}^2 \kappa_N) \). The proof is similar to that which led to (B.7). The case in which \( \hat{z}_i \) equals any diagonal element of \( \mathbf{I} + A_i(\mathbf{A}) \) has been considered in the previous appendix. Hence, we assume that there are no null entries in the diagonal of \( E_i \).

The equation \( \hat{\mathbf{P}}_i \mathbf{v}_i = \mathbf{e}_i \) leads to

$$\mathbf{v}_i = \mathbf{A}(E_i + 2 \mathbf{a}_i \mathbf{a}_i^T)^{-1} \mathbf{a}_i, \quad (C.2)$$

where we have introduced the vector \( \mathbf{a}_i = \mathbf{A}^T \mathbf{e}_i \). Then, by replacing \( (E_i + 2 \mathbf{a}_i \mathbf{a}_i^T)^{-1} \) by

$$E_i^{-1} - 2 \frac{E_i^{-1} \mathbf{a}_i \mathbf{a}_i^T E_i^{-1}}{k}, \quad (C.3)$$

deduced from the Sherman–Morrison–Woodbury formula [10, p. 50], where \( k = 1 + 2a_{ij}^2 E_i^{-1} \mathbf{a}_i \), we
\begin{equation}
    v_i = \frac{A_i}{\|v_i\|_2} \mathbf{a}_i. \quad \text{(C.4)}
\end{equation}

Then, by setting
\begin{equation}
    x_i \leftarrow \frac{a_i^T E^{-1}}{k \|v_i\|_2} \mathbf{s}, \quad \text{(C.5)}
\end{equation}

From (C.5), it follows that \(a_i^T E^{-1}/k \|v_i\|_2\) is the \(i\)th row of the new mixing matrix. The quotient between any of its components gives (19). It can be shown that the Gram–Schmidt algorithm in Step 3 simply sets to zero all components of \(b_i^T A\) which correspond with columns of \(A\) that have been correctly estimated before.

\section*{References}


