Blind separation of bilinear mixtures using mutual information minimization
Fatemeh Mokhtari, Massoud Babaie-Zadeh, Christian Jutten

To cite this version:

HAL Id: hal-00424182
https://hal.archives-ouvertes.fr/hal-00424182
Submitted on 14 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
BLIND SEPARATION OF BILINEAR MIXTURES USING MUTUAL INFORMATION MINIMIZATION

Fatemeh Mokhtari, Massoud Babaie-Zadeh

Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran

ABSTRACT

In this paper an approach for blind source separation in bilinear (or linear quadratic) mixtures is presented. The proposed algorithm employs the same recurrent structure as [Hosseini and Deville, 2003] for separating these mixtures. However, instead of maximal likelihood, our algorithm is based on minimizing the mutual information of the outputs for recovering the independent components. Simulation results show the efficiency of the proposed algorithm.

1. INTRODUCTION

Blind separation of several sources from their linear mixture is a well-known problem, for which many solutions have already been suggested [1, 2]. Separation of nonlinear mixtures, too, has already been addressed by a few authors [3, 4, 5, 6, 7, 8, 9, 10]. In many applications, it is unreasonable to assume that the mixtures are linear because the underlying natural processes are inherently nonlinear. The most important issue to be considered in this case is the separability of the mixtures. Due to very large indeterminacies for nonlinear mixtures, statistical independence of the output is not sufficient to achieve source separation [4, 8]. Hence indeterminacy reduction might be considered as an objective. Generally speaking, constraints such as additional prior information on sources or mixture model make the problem well-posed [3, 4, 11]. As an example, Post Non-Linear (PNL) mixtures have already been shown to be separable [8, 12].

Bilinear (linear quadratic) mixture is one of the nonlinear models with applications such as show-through cancellation in scanned documents [13]. Hosseini et al. in [5] have proposed an approach for source separation which might be extended to higher order polynomial models. This approach suggests a recurrent separating structure that does not re-

 requir the inverse of the mixing model to be known. Consider bilinear mixtures of two independent random sources, namely $s_1$ and $s_2$, as [5]:

$$\begin{cases} x_1 = s_1 - l_1 s_2 - q_1 s_1 s_2 \\ x_2 = s_2 - l_2 s_1 - q_2 s_2 s_1 \end{cases}$$

(1)

where $x_i, i = 1, 2$ are the observations and $l_i$ and $q_i, i = 1, 2$ represent the linear and the quadratic contribution of the sources in the mixtures, respectively. The main objective is to estimate $s_1$ and $s_2$ up to a permutation and a scaling factor [5]. A direct separating structure to recover $s_1$ and $s_2$ for known coefficients $l_1, l_2, q_1$ and $q_2$ has been derived in [14]. Moreover, it is shown that this structure is the inverse of the mixing model if its Jacobian has the same sign for all signal values. Since generalization of this structure for applying to arbitrary polynomial models does not seem possible, a recurrent separating structure, illustrated in Fig. 1, has been proposed in [5]. The computation of structure output, in the case that parameters are exactly known, requires the iteration:

$$\begin{cases} y_1^{(m)}(\cdot) = x_1(\cdot) + l_1 y_2^{(m-1)}(\cdot) + q_1 y_1^{(m-1)}(\cdot) y_2^{(m-1)}(\cdot) \\ y_2^{(m)}(\cdot) = x_2(\cdot) + l_2 y_1^{(m-1)}(\cdot) + q_2 y_1^{(m-1)}(\cdot) y_2^{(m-1)}(\cdot) \end{cases}$$

(2)

Stability is an important issue in iterative structures. Local stability of this model at the separating point $(y_1, y_2) =
\((s_1, s_2)\) has been then studied in [5], based on results on dynamic systems, where it is well known that the model is locally stable if and only if the absolute values of the two eigenvalues of the Jacobian matrix are smaller than one.

To estimate the parameters of the mixing structure, \((l_1, l_2, q_1, q_2)\), in [14] a Maximum Likelihood (ML) approach has been developed. In this paper we propose an approach for estimating these parameters, using Mutual Information (MI) of outputs as the independence criterion.

It is worthy of emphasizing that our approach, as well as the previous approaches [5, 14] is only based on calculation of the parameters which result in as independent as possible outputs in the recurrent structure of Fig. 1. However, as mentioned at the beginning of this section, separability of a mixing-separating structure means that the output independence guarantees source separation for that structure, and up to our best knowledge, this separability for bilinear mixing model followed by the separating structure of Fig. 1, has not yet been shown in the literature and is still an open question.

This paper is organized as follows. Section 2 provides some preliminaries about independence and score functions. The gradient of output mutual information with respect to the parameters of the separating structure is computed then in Section 3. Section 4 presents the final source separation algorithm. Finally, experimental results are given in Section 5.

2. PRELIMINARIES

A random vector \(y = (y_1, \ldots, y_N)^T\) has statistically independent components if and only if

\[
p_y(y) = \prod_{i=1}^N p_{y_i}(y_i)
\]

where \(p_y(y)\) is the joint probability density function (PDF) of vector \(y\) and \(p_{y_i}(y_i)\) is the marginal PDF of the random variables \(y_i\). Mutual information of \(y_i\)'s might be used as an independence criterion and is defined by the Kullback-Leibler divergence between \(p_y(y)\) and \(\prod_{i=1}^N p_{y_i}(y_i)\):

\[
I(y) = D(p_y(y) \parallel \prod_{i=1}^N p_{y_i}(y_i))
\]

\[
= \int p_y(y) \ln \frac{p_y(y)}{\prod_{i=1}^N p_{y_i}(y_i)} \, dy
\]

This function is always non-negative, and is zero if and only if \(y_i\)'s are independent. Therefore to generate independent components in the output of the separating structure, the mutual information of the outputs can be minimized. The parameters of this structure are \(l_1, l_2, q_1\) and \(q_2\) and should be calculated such that output's MI is minimized. For applying the steepest descent gradient algorithm to this minimization problem, the gradient of the outputs' mutual information with respect to these parameters has to be calculated. For doing this, we use a general approach for minimizing mutual information which has been studied in [6, 9, 15]. This approach is based on Score Function Difference (SFD) as a non-parametric “gradient” for mutual information. Here, we review the main definition and results, which requires first, the definition of joint and marginal score function of a random vector. First, recall the definitions of score functions and score function difference.

**Definition 1 (Score Function)** The score function of a scalar random variable \(y\) is the opposite of the log derivative of its density:

\[
\psi_y(y) = -\frac{d}{dy} \ln p_y(y) = -\frac{p'_y(y)}{p_y(y)},
\]

where \(p_y(y)\) denotes the probability density function (PDF) of \(y\).

Let \(y = (y_1, \ldots, y_N)^T\) be a random vector. Then two different score functions may be defined [6, 9, 15]: Marginal Score Function (MSF) and Joint Score Function (JSF).

**Definition 2 (MSF)** The marginal score function of a vector \(y\) is the vector whose components are the score functions of the components of \(y\):

\[
\psi_y(y) = (\psi_1(y_1), \ldots, \psi_N(y_N))^T,
\]

where

\[
\psi_i(y_i) = -\frac{d}{dy_i} \ln p_{y_i}(y_i) = -\frac{p'_{y_i}(y_i)}{p_{y_i}(y_i)}.
\]

**Definition 3 (JSF)** The joint score function of a vector \(y\) is the gradient of \(-\ln p_y(y)\):

\[
\varphi_y(y) = (\varphi_1(y_1), \ldots, \varphi_N(y_N))^T,
\]

where

\[
\varphi_i(y_i) = -\frac{\partial}{\partial y_i} \ln p_y(y) = -\frac{\partial}{\partial y_i} \ln \frac{p_y(y)}{\prod_{i=1}^N p_{y_i}(y_i)}.
\]

The difference between these two score functions is defined in [6, 9, 15] as the Score Function Difference (SFD) of \(y\):

**Definition 4 (SFD)** The Score Function Difference (SFD) of a vector \(y\) is the difference between its JSF and MSF:

\[
\beta_y(y) = \psi_y(y) - \varphi_y(y).
\]
SFD of a random vector $\mathbf{y}$ contains information about the independence of its components, as implied by the following theorem [15]:

**Theorem 1** The components of a random vector $\mathbf{y}$ are independent, if and only if its SFD is zero, i.e.

$$\check{\varphi}_y(\mathbf{y}) = \varphi_y(\mathbf{y})$$  \hspace{1cm} (11)

The “gradient” of the mutual information, needed for our algorithm, is a result of the following theorem [15]:

**Theorem 2** Let $\Delta$ be a ‘small’ random vector with the same dimension of the random vector $\mathbf{y}$. Then:

$$I(\mathbf{y} + \Delta) - I(\mathbf{y}) = E\{\Delta^T \beta_y(\mathbf{y})\} + o(\Delta)$$  \hspace{1cm} (12)

where $o(\Delta)$ denotes higher order terms.

Recall that for a multivariate function $f(x)$, we have:

$$f(\mathbf{y} + \Delta) - f(\mathbf{y}) = \Delta^T \nabla f(\mathbf{y}) + o(\Delta)$$  \hspace{1cm} (13)

Comparing (12) and (13), [15] proposes that SFD can be seen as a non-parametric “gradient” for mutual information. Then, [6, 9] states that (12) provides a general approach for solving mutual information minimization problems. The idea of this general approach is that using (12), one can calculate the deviation resulted in the mutual information of the outputs of a parametric system resulted from a small deviation in its parameters. Finally, this results in the calculation of the gradient of the outputs mutual information with respect to the parameters of the system. [9] has then used this approach for blind source separation of linear instantaneous, convolutive, Post Non-Linear (PNL) and Convolutive PNL (CPNL) mixtures.

In the next section, we will show how this approach can be used for separating bilinear mixtures.

### 3. GRADIENT COMPUTATION

Assume that at the $(m)^{th}$ iteration of the recurrent structure of Fig. 1, we apply a small variation in the parameters of the separating structure. More precisely, let $\tilde{l}_i = l_i + \epsilon_i$ and $\tilde{q}_i = q_i + \eta_i$ for $i = 1, 2$, where $\epsilon_i$ and $\eta_i$ are small values. Thus (2) is rewritten as follows:

$$\begin{pmatrix} y_1^{(m)} \\ y_2^{(m)} \end{pmatrix} = \begin{pmatrix} x_1 + \tilde{l}_1 y_2^{(m-1)} + \tilde{q}_1 y_1^{(m-1)} y_2^{(m-1)} \\ x_2 + \tilde{l}_2 y_1^{(m-1)} + \tilde{q}_2 y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix}$$

$$= \begin{pmatrix} y_1^{(m)} \\ y_2^{(m)} \end{pmatrix} + \begin{pmatrix} \epsilon_1 y_2^{(m-1)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \epsilon_2 y_1^{(m-1)} \end{pmatrix} + \begin{pmatrix} \eta_1 y_2^{(m-1)} y_1^{(m-1)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \eta_2 y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix}$$  \hspace{1cm} (14)

where $y_1^{(m)}$ and $y_2^{(m)}$ are the outputs of the structure at the $(m)^{th}$ iteration. The first term in the right side of equation (14) is the actual output, while the other terms show how the variation of each parameter affects the outputs. Hence by applying Theorem 2, the gradient of mutual information with respect to each parameter is obtained. Let’s assume the following notation for simplicity:

$$I = I(\mathbf{y}^{(m)}), \quad \bar{I} = I(\tilde{\mathbf{y}}^{(m)})$$

Employing Theorem 2 we have:

$$\bar{I} - I = E\left\{ \beta_1^T (\mathbf{y}^{(m)}) \begin{pmatrix} \epsilon_1 y_2^{(m-1)} \\ 0 \end{pmatrix} \right\} + E\left\{ \beta_2^T (\mathbf{y}^{(m)}) \begin{pmatrix} 0 \\ \epsilon_2 y_1^{(m-1)} \end{pmatrix} \right\} + E\left\{ \begin{pmatrix} \eta_1 y_2^{(m-1)} y_1^{(m-1)} \\ 0 \end{pmatrix} \right\} + E\left\{ \begin{pmatrix} 0 \\ \eta_2 y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix} \right\}$$  \hspace{1cm} (15)

Let see how (15) can result in the calculation of the derivative with respect to $l_1$. The first term of (15) corresponds to the effect of $l_1$, therefore it may be used in calculating the derivative with respect to $l_1$. The ratio of $\bar{I} - I$ to $\epsilon_1$ denotes the partial derivative of $I$ with respect to $l_1$, and hence:

$$\frac{\partial}{\partial l_1} I = E\left\{ \beta_1^T (\mathbf{y}^{(m)}) \begin{pmatrix} \eta_1 y_2^{(m-1)} y_1^{(m-1)} \\ 0 \end{pmatrix} \right\}$$

$$= E\left\{ \beta_1 (\mathbf{y}^{(m)}) y_2^{(m-1)} \right\}$$  \hspace{1cm} (16)

where $\beta_i(\mathbf{y}^{(m)}), i = 1, 2$, denotes the $i^{th}$ component of $\beta_i(\mathbf{y}^{(m)})$. The gradient with respect to other variables is calculated in a similar manner, which gives:

$$\frac{\partial}{\partial q_1} I = E\left\{ \beta_1 (\mathbf{y}^{(m)}) y_1^{(m-1)} y_2^{(m-1)} \right\}$$  \hspace{1cm} (17)

$$\frac{\partial}{\partial q_2} I = E\left\{ \beta_2 (\mathbf{y}^{(m)}) y_1^{(m-1)} y_2^{(m-1)} \right\}$$  \hspace{1cm} (18)

$$\frac{\partial}{\partial l_2} I = E\left\{ \beta_2 (\mathbf{y}^{(m)}) y_1^{(m-1)} y_2^{(m-1)} \right\}$$  \hspace{1cm} (19)

### 4. SOURCE SEPARATION ALGORITHM

To minimize the mutual information of the outputs, we apply a steepest descent algorithm using the gradient calculated in the previous section. In this procedure, the parameters are initially set to zero. In each iteration, outputs are computed using (2) and each parameter is adjusted by forcing it to move in the negative direction of the gradient of
which is defined as follows:

\[
\text{SNR} = 0.5 \sum_{i=1}^{2} 10 \log_{10} \frac{E[s_i^2]}{E[(y_i - s_i)^2]}
\]

The simulation results, depicted in Fig. 3 and Fig. 4, illustrate the efficiency of the proposed algorithm in source separation and parameter estimation. For estimation of SFD, a method proposed by Pham in [6, 16] is employed for performing simulations.

We use the CPU time as a measure of complexity. Although it is not an exact measure, it gives a rough estimation of the complexity, for comparing proposed algorithm and ML estimator. Our simulations are performed in MATLAB environment using an AMD Athlon 4000+, 2.1GHz processor with 896MB of memory, and under Microsoft Windows XP operating system. The time required for running 100 iterations of updating parameters for the proposed algorithm is approximately 1.75 seconds and for ML estimator is 7.2 seconds.

The performance of the proposed algorithm for separation of other combinations of sources is further investigated by repeating the above simulation for two other distributions of sources: i) One source is uniform as before, the other is Laplacian, ii) both sources are Laplacian. For Laplacian sources, the pdf is \( f_s(s) = 5 \exp(-10|s|) \).

**Algorithm parameters:** \( \mu_1, \mu_2, \nu_1, \nu_2 \)

**Recurrent structure parameters:** \( l_1, l_2, q_1, q_2 \)

**Input:** two mixtures \( x_1 \) and \( x_2 \)

**Initialization**

Let: \( l_1 = l_2 = q_1 = q_2 = 0 \)

Let: \( y = 0 \)

**Loop**

1. **Computation of structure outputs for all times**

   \[
   y_1(t) = x_1(t) + l_1 y_2(t) + q_1 y_1(t) y_2(t)
   \]

   \[
   y_2(t) = x_2(t) + l_2 y_1(t) + q_2 y_1(t) y_2(t)
   \]

2. **Estimation of SFD**

   \[
   \beta(n) = \beta(y(n))
   \]

3. **Gradient calculation**

   \[
   \frac{\partial l}{\partial y_1} I = \frac{1}{N} \sum_{n=0}^{N} \beta_1(n) y_2(n)
   \]

   \[
   \frac{\partial l}{\partial y_2} I = \frac{1}{N} \sum_{n=0}^{N} \beta_2(n) y_1(n)
   \]

   \[
   \frac{\partial q_1}{\partial y_1} I = \frac{1}{N} \sum_{n=0}^{N} \beta_1(n) y_1(n) y_2(n)
   \]

   \[
   \frac{\partial q_2}{\partial y_2} I = \frac{1}{N} \sum_{n=0}^{N} \beta_2(n) y_1(n) y_2(n)
   \]

4. **Update of parameters:**

   \[
   l_1 \leftarrow l_1 - \mu_1 \frac{\partial l}{\partial y_1}
   \]

   \[
   l_2 \leftarrow l_2 - \mu_2 \frac{\partial l}{\partial y_2}
   \]

   \[
   q_1 \leftarrow q_1 - \nu_1 \frac{\partial q_1}{\partial y_1}
   \]

   \[
   q_2 \leftarrow q_2 - \nu_2 \frac{\partial q_2}{\partial y_2}
   \]

**Repeat until convergence**

**Fig. 2.** The final pseudo-code of the proposed algorithm
obtained results, the suggested approach demonstrates a superior functionality.

Table 1. Mean and standard deviation of output SNR (in dB)

<table>
<thead>
<tr>
<th>different combination of sources</th>
<th>ML estimator</th>
<th>MI minimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
</tr>
<tr>
<td>$s_1$ and $s_2$ uniform</td>
<td>28.0</td>
<td>4.2</td>
</tr>
<tr>
<td>$s_1$ uniform, $s_2$ Laplacian</td>
<td>27.8</td>
<td>3.8</td>
</tr>
<tr>
<td>$s_1$ Laplacian, $s_2$ Laplacian</td>
<td>26.8</td>
<td>3.1</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this paper the problem of blind source separation in bilinear (linear quadratic) mixtures was addressed. The proposed algorithm takes advantage from a previously designed structure, brought together with a new idea for parameters estimation based on mutual information minimization. The simulation results emphasize on the functionality of the proposed method.

7. REFERENCES


