HAL open science

Blind separation of bilinear mixtures using mutual information minimization

Fatemeh Mokhtari, Massoud Babaie-Zadeh, Christian Jutten

To cite this version:

Fatemeh Mokhtari, Massoud Babaie-Zadeh, Christian Jutten. Blind separation of bilinear mixtures using mutual information minimization. MLSP 2009 - IEEE 19th International Workshop on Machine Learning for Signal Processing, Sep 2009, Grenoble, France. 6 p. hal-00424182

HAL Id: hal-00424182

https://hal.science/hal-00424182v1

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

Christian Jutten

GIPSA-lab, Grenoble, France
and Institut Universitaire de France

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 require 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]:

*This work has been partially supported by Iran National Science Foundation (INSF) under contract number 86/994, by Iran Telecommunications Research Center (ITRC), and also by ISMO and French embassy in Iran in the framework of a Gundi-Shapour collaboration program.

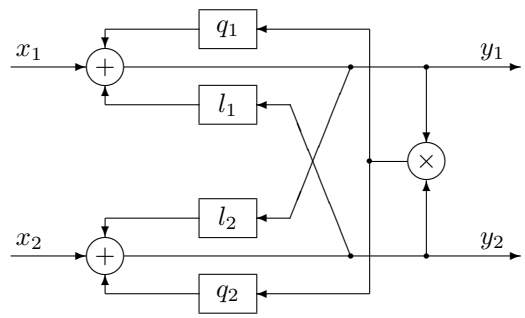

Fig. 1. Recurrent structure proposed by Hosseini and Deville [5, 14] for inverting bilinear mixtures.

$$\begin{cases} x_1 = s_1 - l_1 s_2 - q_1 s_1 s_2 \\ x_2 = s_2 - l_2 s_1 - q_2 s_1 s_2 \end{cases} \tag{1}$$

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:

$$\begin{cases} y_1^{(m)}(\cdot) = x_1(\cdot) + l_1\, y_2^{(m\text{-}1)}(\cdot) + q_1\, y_1^{(m\text{-}1)}(\cdot)\, y_2^{(m\text{-}1)}(\cdot) \\ y_2^{(m)}(\cdot) = x_2(\cdot) + l_2\, y_1^{(m\text{-}1)}(\cdot) + q_2\, y_1^{(m\text{-}1)}(\cdot)\, y_2^{(m\text{-}1)}(\cdot) \end{cases} \tag{2}$$

Stability is an important issue in iterative structures. Local stability of this model at the separating point $(y_1, y_2) =$

978-1-4244-4948-4/09/$25.00 © 2009 IEEE

(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 $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ has statistically independent components if and only if

$$p_{\mathbf{y}}(\mathbf{y}) = \prod_{i=1}^{N} p_{y_i}(y_i) \tag{3}$$

where $p_{\mathbf{y}}(\mathbf{y})$ is the joint probability density function (PDF) of vector $\boldsymbol{y}$ 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 $p_{\mathbf{y}}(\mathbf{y})$ and $\prod_{i=1}^{N} p_{y_i}(y_i)$:

$$\begin{aligned} I(\mathbf{y}) &= D(p_{\mathbf{y}}(\mathbf{y}) \parallel \prod_{i=1}^{N} p_{y_i}(y_i)) \\ &= \int_{\mathbf{y}} p_{\mathbf{y}}(\mathbf{y}) \ln \frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^{N} p_{y_i}(y_i)} \mathbf{dy} \end{aligned} \tag{4}$$

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:*

$$\psi_y(y) = -\frac{d}{dy} \ln p_y(y) = -\frac{p'_y(y)}{p_y(y)}, \tag{5}$$

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

Let $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ 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 $\boldsymbol{y}$ is the vector whose components are the score functions of the components of $\boldsymbol{y}$:*

$$\boldsymbol{\psi}_{\boldsymbol{y}}(\boldsymbol{y}) = (\psi_1(y_1), \ldots, \psi_N(y_N))^T, \tag{6}$$

where

$$\psi_i(y_i) = -\frac{d}{dy_i} \ln p_{y_i}(y_i) = -\frac{p'_{y_i}(y_i)}{p_{y_i}(y_i)}. \tag{7}$$

Definition 3 (JSF) *The joint score function of a vector $\boldsymbol{y}$ is the gradient of $-\ln p_{\boldsymbol{y}}(\boldsymbol{y})$:*

$$\boldsymbol{\varphi}_{\boldsymbol{y}}(\boldsymbol{y}) = (\varphi_1(\boldsymbol{y}), \ldots, \varphi_N(\boldsymbol{y}))^T, \tag{8}$$

where

$$\varphi_i(\boldsymbol{y}) = -\frac{\partial}{\partial y_i} \ln p_{\boldsymbol{y}}(\boldsymbol{y}) = -\frac{\frac{\partial}{\partial y_i} p_{\boldsymbol{y}}(\boldsymbol{y})}{p_{\boldsymbol{y}}(\boldsymbol{y})}. \tag{9}$$

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

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

$$\boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y}) = \boldsymbol{\psi}_{\boldsymbol{y}}(\boldsymbol{y}) - \boldsymbol{\varphi}_{\boldsymbol{y}}(\boldsymbol{y}). \tag{10}$$

SFD of a random vector $\boldsymbol{y}$ contains information about the independence of its components, as implied by the following theorem [15]:

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

$$\boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) = \boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) \tag{11}$$

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

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

$$I(\mathbf{y}+\boldsymbol{\Delta}) - I(\mathbf{y}) = E\{\boldsymbol{\Delta}^T \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\} + o(\boldsymbol{\Delta}) \tag{12}$$

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

Recall that for a multivariate function $f(\boldsymbol{x})$, we have:

$$f(\mathbf{y}+\boldsymbol{\Delta}) - f(\mathbf{y}) = \boldsymbol{\Delta}^T \nabla f(\mathbf{y}) + o(\boldsymbol{\Delta}) \tag{13}$$

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 ϵ_i and η_i are small values. Thus (2) is rewritten as follows:

$$\begin{aligned}\begin{pmatrix} \tilde{y}_1^{(m)} \\ \tilde{y}_2^{(m)} \end{pmatrix} &= \begin{pmatrix} x_1 + \tilde{l}_1\, y_2^{(m-1)} + \tilde{q}_1\, y_1^{(m-1)} y_2^{(m-1)} \\ x_2 + \tilde{l}_2\, y_1^{(m-1)} + \tilde{q}_2\, y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix} \\ &= \begin{pmatrix} y_1^{(m)} \\ y_2^{(m)} \end{pmatrix} + \begin{pmatrix} \epsilon_1 y_2^{(m-1)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \epsilon_2 y_1^{(m-1)} \end{pmatrix} \\ &+ \begin{pmatrix} \eta_1 y_2^{(m-1)} y_2^{(m-1)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \eta_2 y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix}\end{aligned} \tag{14}$$

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:

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

Employing Theorem 2 we have:

$$\begin{aligned}\tilde{I} - I &= E\left\{ \boldsymbol{\beta}_{\mathbf{y}}^T(\mathbf{y}^{(m)}) \begin{pmatrix} \epsilon_1\, y_2^{(m-1)} \\ 0 \end{pmatrix} \right\} \\ &+ E\left\{ \boldsymbol{\beta}_{\mathbf{y}}^T(\mathbf{y}^{(m)}) \begin{pmatrix} 0 \\ \epsilon_2\, y_1^{(m-1)} \end{pmatrix} \right\} \\ &+ E\left\{ \boldsymbol{\beta}_{\mathbf{y}}^T(\mathbf{y}^{(m)}) \begin{pmatrix} \eta_1\, y_1^{(m-1)} y_2^{(m-1)} \\ 0 \end{pmatrix} \right\} \\ &+ E\left\{ \boldsymbol{\beta}_{\mathbf{y}}^T(\mathbf{y}^{(m)}) \begin{pmatrix} 0 \\ \eta_2\, y_1^{(m-1)} y_2^{(m-1)} \end{pmatrix} \right\}\end{aligned} \tag{15}$$

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 $\tilde{I} - I$ to ϵ_1 denotes the partial derivative of I with respect to l_1, and hence:

$$\begin{aligned}\frac{\partial}{\partial l_1} I &= E\left\{ \boldsymbol{\beta}_{\mathbf{y}}^T(\mathbf{y}^{(m)}) \begin{pmatrix} y_2^{(m-1)} \\ 0 \end{pmatrix} \right\} \\ &= E\left\{ \beta_1(\mathbf{y}^{(m)})\, y_2^{(m-1)} \right\}\end{aligned} \tag{16}$$

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

$$\frac{\partial}{\partial l_2} I = E\left\{ \beta_2(\mathbf{y}^{(m)})\, y_1^{(m-1)} \right\} \tag{17}$$

$$\frac{\partial}{\partial q_1} I = E\left\{ \beta_1(\mathbf{y}^{(m)})\, y_1^{(m-1)}\, y_2^{(m-1)} \right\} \tag{18}$$

$$\frac{\partial}{\partial q_2} I = E\left\{ \beta_2(\mathbf{y}^{(m)})\, y_1^{(m-1)}\, y_2^{(m-1)} \right\} \tag{19}$$

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

mutual information of output with respect to that parameter. Parameter values are updated by the following rules:

$$\begin{aligned} l_1 &\leftarrow l_1 - \mu_1 \frac{\partial I}{\partial l_1} \\ l_2 &\leftarrow l_2 - \mu_2 \frac{\partial I}{\partial l_2} \\ q_1 &\leftarrow q_1 - \nu_1 \frac{\partial I}{\partial q_1} \\ q_2 &\leftarrow q_2 - \nu_2 \frac{\partial I}{\partial q_2} \end{aligned} \tag{20}$$

where the step-sizes μ_1, μ_2, ν_1 and ν_2 are small positive constants that control the speed of convergence. In the next iteration, these updated parameters are used in the recurrent structure for computing the new outputs for all times. The expectations of equations (16)-(19) are estimated by ensemble averaging. The final algorithm has been shown in Fig. 2. Once the algorithm converges, the independent components at the output of the structure are achieved, because the SFD of the outputs converges to zero.

In the ML estimator proposed in [14], the actual sources are needed to estimate the parameters of the separating structure. However, in blind case both parameters and sources are unknown. As proposed in [14], the outputs of the separating structure, y_i, in an iterative algorithm can be used as the reconstructed sources to update the parameters. Hence, for each updated values of parameters (l_1, l_2, q_1, q_2), the structure of Fig. 1 is iterated until convergence to obtain new outputs, and the parameters are modified based on these convergent outputs. This procedure repeats until convergence of all parameters. But in our algorithm, in each recurrence of the separating structure, both parameters and outputs of the structure are updated.

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: $\mathbf{y} = \mathbf{0}$
- ***Loop***
  1. *Computation of structure outputs for all times*
     $y_1(\cdot) \leftarrow x_1(\cdot) + l_1\, y_2(\cdot) + q_1\, y_1(\cdot)\, y_2(\cdot)$
     $y_2(\cdot) \leftarrow x_2(\cdot) + l_2\, y_1(\cdot) + q_2\, y_1(\cdot)\, y_2(\cdot)$
  2. *Estimation of SFD*
     $\boldsymbol{\beta}(n) = \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y}(n))$
  3. *Gradient calculation*
     $\frac{\partial}{\partial l_1} I = \frac{1}{N} \sum_{n=0}^{N} \beta_1(n)\, y_2(n)$
     $\frac{\partial}{\partial l_2} I = \frac{1}{N} \sum_{n=0}^{N} \beta_2(n)\, y_1(n)$
     $\frac{\partial}{\partial q_1} I = \frac{1}{N} \sum_{n=0}^{N} \beta_1(n)\, y_1(n)\, y_2(n)$
     $\frac{\partial}{\partial q_2} I = \frac{1}{N} \sum_{n=0}^{N} \beta_2(n)\, y_1(n)\, y_2(n)$
  4. *Update of parameters*:
     $l_1 \leftarrow l_1 - \mu_1 \frac{\partial I}{\partial l_1}$
     $l_2 \leftarrow l_2 - \mu_2 \frac{\partial I}{\partial l_2}$
     $q_1 \leftarrow q_1 - \nu_1 \frac{\partial I}{\partial q_1}$
     $q_2 \leftarrow q_2 - \nu_2 \frac{\partial I}{\partial q_2}$
- ***Repeat until convergence***

Fig. 2. The final pseudo-code of the proposed algorithm

5. EXPERIMENTAL RESULTS

In order to compare the performance of the proposed algorithm with the ML estimator of [14], the experiment performed in [14] is re-done. The model parameters are: $l_1 = -0.2$, $l_2 = 0.2$, $q_1 = -0.8$ and $q_2 = 0.8$ and the source values are in the range $[-0.5, 0.5]$. In this experiment, the step-sizes are set to: $\mu_1 = \mu_2 = 0.2$ and $\nu_1 = \nu_2 = 0.5$. The Simulation is performed using 1000 samples from two independent sources, uniformly distributed over $[-0.5, 0.5]$. The experiment has been repeated 100 times to cover different realization of the sources. Output Signal to Noise Ratio (SNR) is used as the performance measure of the algorithm, which is defined as follows:

$$\text{SNR} = 0.5 \sum_{i=1}^{2} 10 \log_{10} \frac{E[s_i^2]}{E[(y_i - s_i)^2]} \tag{21}$$

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 MATLAB7 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 $f_s(s) = 5 \exp(-10|s|)$. Table 1 summarizes the simulation results for all three cases and provides a comparison between the performance of the proposed algorithm and that of the ML estimator. According to the

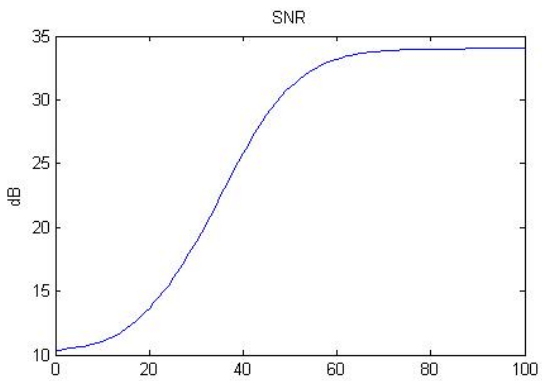

Fig. 3. Averaged SNR (over 100 runs of the algorithm) versus iteration.

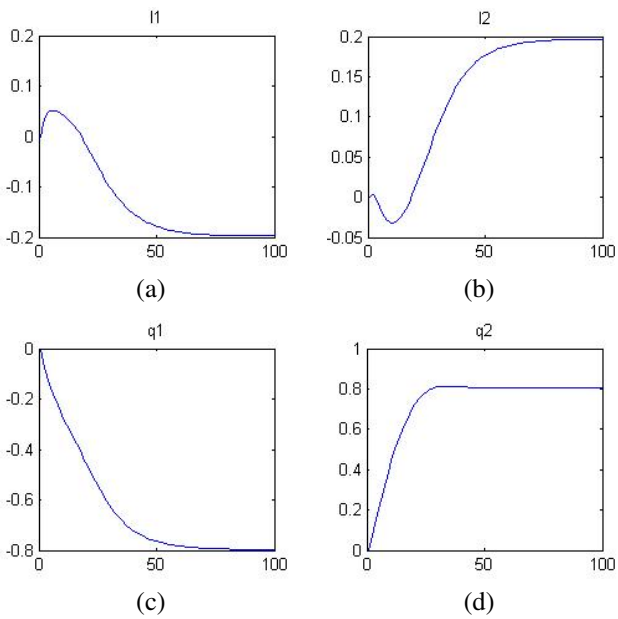

Fig. 4. Estimated parameters (averaged over 100 runs) versus iteration, (a) l_1, (b) l_2, (c) q_1, (d) q_2.

obtained results, the suggested approach demonstrates a superior functionality.

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

different combination of sources	ML estimator		MI minimizer	
	Mean	Std	Mean	Std
s_1 and s_2 uniform	28.0	4.2	34	4.2
s_1 uniform, s_2 Laplacian	27.8	3.8	30.8	4.5
s_1 Laplacian, s_2 Laplacian	26.8	3.1	28.2	2.9

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

[1] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[2] Andrzej Cichocki and Shun-ichi Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, John Wiley and sons, 2002.

[3] J. Eriksson and V. Koivunen, "Blind identifiability of a class of nonlinear instantaneous ICA models," in *EUSIPCO*, Toulouse (France), September 2002, pp. 7–10.

[4] C. Jutten, M. Babaie-Zadeh, and S. Hosseini, "Three easy ways for separating nonlinear mixtures?," *Signal Processing*, vol. 84, no. 2, pp. 217–229, February 2004.

[5] S. Hosseini and Y. Deville, "Blind separation of linear-quadratic mixtures of real sources using a recurrent structure," in *IWANN*, Mao, Menorca, Spain, June 2003, vol. 2, pp. 241–248.

[6] M. Babaie-Zadeh, *On blind source separation in convolutive and nonlinear mixtures*, Ph.D. thesis, INP Grenoble, 2002.

[7] A. Taleb and C. Jutten, "Non-linear source separation: the post non-linear mixtures," in *Proceedings of ESANN'97*, Bruges, Belgium, April 1997, pp. 279–284.

[8] A. Taleb and C. Jutten, "Source separation in post nonlinear mixtures," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2807–2820, 1999.

[9] M. Babaie-Zadeh and C. Jutten, "A general approach for mutual information minimization and its application to blind source separation," *Signal Processing*, vol. 85, no. 5, pp. 975–995, May 2005.

[10] L. T. Duarte and C. Jutten, "A mutual information minimization approach for a class of nonlinear recurrent separating systems," in *IEEE International Workshop on Machine Learning for Signal Processing*, Aug 2007, pp. 122–127.

[11] A. Taleb, "A generic framework for blind source separation in structured nonlinear models," *IEEE Trans. Signal processing*, vol. 50, no. 8, pp. 1819–1830, 2002.

[12] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "A geometric approach for separating Post Non-Linear mixtures," in *EUSIPCO*, Toulouse, France, September 2002, vol. II, pp. 11–14.

[13] F. Merrikh-Bayat, M. Babaie-Zadeh1, and C. Jutten, "A nonlinear blind source separation solution for removing the show-through effect in the scanned documents," in *Proc. European Signal Processing Conf. EUSIPCO'08*, Lausanne, Suisse, September 2008.

[14] S. Hosseini and Y. Deville, "Blind maximum likelihood separation of a linear-quadratic mixture," in *5th Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA'04)*, Granada, Spain, September 2004, pp. 694–701.

[15] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Differential of mutual information function," *IEEE Signal Processing Letters*, vol. 11, no. 1, pp. 48–51, January 2004.

[16] D. T. Pham, "Estimation de la fonction score conditionnelle et l'entropie conditionnelle," Tech. Rep., 2002, (in French).