4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan

MINIMIZATION-PROJECTION (MP) APPROACH FOR BLIND SOURCE SEPARATION IN DIFFERENT MIXING MODELS

Massoud Babaie-Zadeh[1,2], *Christian Jutten*[1], *Kambiz Nayebi*[2]

[1]Institut National Polytechnique de Grenoble (INPG), Laboratoire des images et des signaux (LIS), Grenoble, France

[2]Electrical engineering department, Sharif University of Technology, Tehran, Iran
`mbzadeh@yahoo.com,Christian.Jutten@inpg.fr,knayebi@sina.sharif.ac.ir`

ABSTRACT

In this paper, a new approach for blind source separation is presented. This approach is based on minimization of the mutual information of the outputs using a nonparametric "gradient" of mutual information, followed by a projection on the parametric model of the separation structure. It is applicable to different mixing system, linear as well as nonlinear, and the algorithms derived from this approach are very fast and efficient.

1. INTRODUCTION

Blind Source Separation (BSS) or Independent Component Analysis (ICA) is a relatively new subject in signal processing which has been started in the mid 80's [1, 2] (see also [3] for historical notes). Main insights in this domain has been then done in Signal Processing [4, 5, 6], and Neural Networks [7, 8, 9, 10] communities. The problem consists in retrieving unobserved independent mixed signals from mixtures of them, assuming there is neither information about the original source signals, nor about the mixing system (hence the term *Blind*).

To state the problem, suppose that N observed signals $x_1(t), \dots, x_N(t)$ are given, which are assumed to be the mixtures of N independent source signals $s_1(t), \dots, s_N(t)$ (here, the number of sources is assumed to be equal to the number of observations). The observation vector $\mathbf{x}(t) \triangleq (x_1(t), \dots, x_N(t))^T$ is related to the source vector $\mathbf{s}(t) \triangleq (s_1(t), \dots, s_N(t))^T$ by the mixing system $\mathcal{F}$, that is $\mathbf{x}(t) = \mathcal{F}\big(\mathbf{s}(t)\big)$. The goal of BSS is to construct a *separating system* $\mathcal{G}$ (Fig. 1) in order to isolate in each component of the output vector $\mathbf{y}(t) \triangleq \mathcal{G}\big(\mathbf{x}(t)\big)$ the image of one source:

$$y_i(t) = h_i\left(s_{\sigma(i)}(t)\right) \ , \quad i = 1, \dots, N \tag{1}$$

This work has been partially funded by the European project Blind Source Separation and applications (BLISS, IST 1999-13077) and by the French project Statistiques Avancées et Signal (SASI).

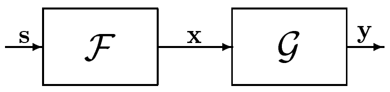

Fig. 1. Mixing and separating systems.

where σ is a permutation, and h_i stands for any invertible mapping.

Since there is no information about the source signals but their statistical independence, one can think about designing the system $\mathcal{G}$ to produce statistical independent outputs. Now, the question of *separability* arises: "Does the independence of the outputs (ICA) insure the blind separation of the sources (BSS)"? In general cases, the answer of this question is negative, and there exist nonlinear mappings which mix the source signals and preserve their statistical independence [11, 12].

However, if a special model is assumed for the mixing system $\mathcal{F}$, and if the mappings $\mathcal{G}$ has to belong to a suited parametric family $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$, then a separability property may exist for the global (parametric) system $\mathcal{G} \circ \mathcal{F}$, *i.e.* ICA coincides with BSS. For example, if $\mathcal{F}$ is assumed to be linear and instantaneous (*i.e.* it is a mixing matrix), and if $\mathcal{G}$ forced to be a linear instantaneous mapping (*i.e.* a separating matrix $\mathbf{B}$), then this mixing-separating model is separable [4]. In other words, if $\mathbf{B}$ is calculated to produce independent outputs, then the source signals will be retrieved up to a permutation and a scale indeterminacy.

Post Non-Linear (PNL) mixtures [11] are special nonlinear mixtures in which a linear instantaneous mixture is followed by component-wise and invertible nonlinearities (Fig. 2). This corresponds to the case where the mixture is itself linear, but the sensors have nonlinear effects (such as saturation). Although nonlinear system are generally non separable, it has been proved that the mixing-separating system as shown in Fig. 2, is separable [11, 13].

In this paper, we propose a general approach for source

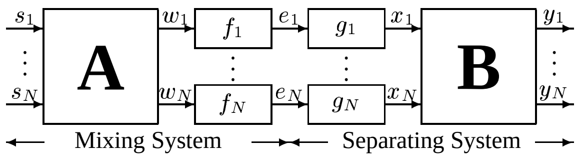

Fig. 2. Mixing-separating system for PNL mixtures.

separation, based on minimizing the mutual information of the outputs. In Section 2 some preliminary issues are considered, and a nonparametric "gradient" for mutual information is presented. The basic idea of the approach is stated in Section 3. Sections 4 and 5 explain how this approach can be used for separating linear and PNL mixtures. Finally, Section 6 contains experimental results.

2. PRELIMINARY ISSUES

2.1. Independence Criterion

For determining the parameters of the separating system which produce independent outputs, a criterion for measuring the statistical independence of the outputs is needed. Recall that the random variables $y_1, \ldots, y_N$ are independent if and only if:

$$p_{\mathbf{y}}(\mathbf{y}) = \prod_{i=1}^{N} p_{y_i}(y_i). \tag{2}$$

A convenient independence measure is mutual information of y_i's, denoted by $I(\mathbf{y})$, which is nothing but the Kullback-Leibler divergence between $p_{\mathbf{y}}(\mathbf{y})$ and $\prod_{i=1}^{N} p_{y_i}(y_i)$:

$$\begin{aligned} I(\mathbf{y}) &= D(p_{\mathbf{y}}(\mathbf{y}) \parallel \prod_{i=1}^{N} p_{y_i}(y_i)) \\ &= \int_{\mathbf{y}} p_{\mathbf{y}}(\mathbf{y}) \ln \frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^{N} p_{y_i}(y_i)} \mathbf{dy} \end{aligned} \tag{3}$$

This function is always non negative, and vanishes if and only if the y_i's are independent. Consequently, a source separation algorithm can be designed based on the minimization of the mutual information of the outputs. Recently (see section 2.3), a nonparametric "gradient" for the mutual information has been proposed [14]. This gradient, will help us in minimizing a mutual information. Stating this gradient requires the definition of multivariate score functions.

2.2. Multivariate Score Functions

In this section, we recall the definitions of multivariate score functions [15]. For the scalar case, we know the following definition from the statistics literature:

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

$$\psi_y(y) = -\frac{d}{dy} \ln p_y(y) = -\frac{p'_y(y)}{p_y(y)} \tag{4}$$

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

But for a random vector $\mathbf{y} = (y_1, \ldots, y_N)^T$ we define two different kinds of score functions. Let $p_{\mathbf{y}}(\mathbf{y})$ and $p_{y_i}(y_i)$ denote the joint and marginal PDFs, respectively.

Definition 2 (MSF) *The marginal score function (MSF) of $\mathbf{y}$ is the vector whose i-th component is the score function of the i-th random variable, i.e.:*

$$\boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) = (\psi_1(y_1), \ldots, \psi_N(y_N))^T \tag{5}$$

where:

$$\psi_i(y_i) = -\frac{d}{dy_i} \ln p_{y_i}(y_i) = -\frac{p'_{y_i}(y_i)}{p_{y_i}(y_i)}. \tag{6}$$

Definition 3 (JSF) *The joint score function (JSF) of $\mathbf{y}$ is the gradient of $-\ln p_{\mathbf{y}}(\mathbf{y})$, i.e.:*

$$\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) = (\varphi_1(\mathbf{y}), \ldots, \varphi_N(\mathbf{y}))^T \tag{7}$$

where:

$$\varphi_i(\mathbf{y}) = -\frac{\partial}{\partial y_i} \ln p_{\mathbf{y}}(\mathbf{y}) = -\frac{\frac{\partial}{\partial y_i} p_{\mathbf{y}}(\mathbf{y})}{p_{\mathbf{y}}(\mathbf{y})} \tag{8}$$

The difference of these two score functions contains information about the independence of the components of $\mathbf{y}$, and hence it worths to give it a formal name:

Definition 4 (SFD) *The score function difference (SFD) of $\mathbf{y}$ is the difference between its MSF and JSF:*

$$\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y}) = \boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) - \boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) \tag{9}$$

The following theorem relates the independence of the components of a random vector $\mathbf{y}$ to its SFD [16].

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

$$\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) = \boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) \tag{10}$$

2.3. "Gradient" of mutual information

The variations of mutual information resulted from a small deviation in its argument (the "differential" of mutual information), is given by the following theorem [14]:

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

$$I(\mathbf{y}+\mathbf{\Delta}) - I(\mathbf{y}) = E\left\{\mathbf{\Delta}^T \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\right\} + o(\mathbf{\Delta}) \qquad (11)$$

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

Note that for any multivariate differentiable function $f(\mathbf{y})$, we have:

$$f(\mathbf{y}+\mathbf{\Delta}) - f(\mathbf{y}) = \mathbf{\Delta}^T \nabla f(\mathbf{y}) + o(\mathbf{\Delta}) \qquad (12)$$

A comparison between (11) and (12) shows that SFD can be called the *stochastic gradient* of the mutual information.

3. MINIMIZATION-PROJECTION APPROACH FOR BLIND SOURCE SEPARATION

3.1. Mathematical foundation

For minimizing a multivariate differentiable function $f(\mathbf{y})$, its gradient may be used through the steepest descent algorithm:

$$\mathbf{y} \leftarrow \mathbf{y} - \mu \nabla f(\mathbf{y}) \qquad (13)$$

Having the gradient of the mutual information (Theorem 2), one can think about a similar approach for its minimization:

$$\mathbf{y} \leftarrow \mathbf{y} - \mu \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y}) \qquad (14)$$

Now, the following theorem is stated:

Theorem 3 *Let $\mathbf{y}_n$ be a random vector and denote its SFD by $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y})$. If:*

$$\mathbf{y}_{n+1} \triangleq \mathbf{y}_n - \mu \boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y}_n) \qquad (15)$$

where μ is a small positive scalar, then $I(\mathbf{y}_{n+1}) \leq I(\mathbf{y}_n)$. Moreover, if $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y})$ is continuous, then the equality holds if and only if the components of $\mathbf{y}_n$ are independent.

This theorem proves that the algorithm (14) makes a reduction in $I(\mathbf{y})$ at each iteration. It proves also that the algorithm converges when the components of $\mathbf{y}$ are independent: *the algorithm has no "local minimum"*.

The proof of the theorem is left to the appendix.

3.2. The approach

As stated in Section 1, in source separation, there is a parametric separating system $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$, and the parameters must be determined to produce independent outputs. Although the algorithm (14) results in independent outputs, it does not insure that the separating system belongs to the family $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$. In other words, by this algorithm, we are directly manipulating the outputs, and after the convergence there may be no particular relationship between the observations ($\mathbf{x}$) and the outputs ($\mathbf{y}$).

To solve this problem, we propose to use a "projection" approach. That is, at each iteration, the mapping $\mathbf{x} \mapsto \mathbf{y}$ is replaced by its projection on the family $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$. In other words, each iteration of the separation algorithm is composed of the following steps:

- Minimization:
  1. $\mathbf{y} \leftarrow \mathbf{y} - \mu \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$.
- Projection:
  2. $\boldsymbol{\theta}_0 = \operatorname{argmin}_{\boldsymbol{\theta}} E\{\|\mathbf{y} - \mathcal{G}(\mathbf{x}; \boldsymbol{\theta})\|^2\}$.
  3. $\mathbf{y} = \mathcal{G}(\mathbf{x}, \boldsymbol{\theta}_0)$.

Consequently, $I(\mathbf{y})$ is being minimized while staying in the family $\mathbf{y} = \mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$. In the following, this method is called the Minimization-Projection (MP) approach.

The MP approach for BSS is very general, and can be used in different mixing models. The minimization step is the same for all mixtures, but the projection step (step 2) is different: for each mixing model, a different projection problem must be solved, *i.e.* one has to found the projection of a general mapping $\mathbf{x} \mapsto \mathbf{y}$ on the desired family.

In the following sections, it is shown how the MP approach can be used for designing new algorithms for separating linear and PNL mixtures.

4. APPLICATION TO LINEAR MIXTURES

4.1. Calculating the projected mapping

In linear (instantaneous) mixtures, the separating system is of the form $\mathbf{y} = \mathbf{B}\mathbf{x}$. Consequently, finding the projected mapping (step 2 of the general approach) consists in finding the matrix $\mathbf{B}$ which minimizes $E\{\|\mathbf{y} - \mathbf{B}\mathbf{x}\|^2\}$. This is given by the following well-known lemma:

Lemma 1 *The matrix $\mathbf{B}$ which minimizes $E\{\|\mathbf{y} - \mathbf{B}\mathbf{x}\|^2\}$ is:*

$$\mathbf{B}_0 = E\left\{\mathbf{y}\mathbf{x}^T\right\} \left(E\left\{\mathbf{x}\mathbf{x}^T\right\}\right)^{-1} \qquad (16)$$

4.2. The Algorithm

From the above lemma, the MP algorithm for separating linear instantaneous mixtures will be obtained as shown in Fig. 3. Step 3 of the algorithm loop is required to overcome the mean and scaling indeterminacies.

5. APPLICATION TO PNL MIXTURES

5.1. Calculating the projected mapping

For PNL mixtures, the calculation of the projected mapping is more complicated than for linear mixtures. The problem consists in finding the matrix $\mathbf{B}$ and the invertible functions

```
• Initialization: $\mathbf{y} = \mathbf{x}$.
• Loop:
    1. Estimate $\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$, the SFD of $\mathbf{y}$.
    2. $\mathbf{y} \leftarrow \mathbf{y} - \mu\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$.
    3. Remove the mean of each output, and normalize its energy to 1.
    4. $\mathbf{B} = E\left\{\mathbf{y}\mathbf{x}^T\right\}\left(E\left\{\mathbf{x}\mathbf{x}^T\right\}\right)^{-1}$
    5. $\mathbf{y} = \mathbf{B}\mathbf{x}$.
• Repeat until convergence.
```

Fig. 3. MP algorithm for separating linear instantaneous mixtures.

g_i (see Fig. 2) which minimize $E\{\|\mathbf{y} - \mathbf{B}\mathbf{g}(\mathbf{e})\|^2\}$. In this section, we propose an iterative algorithm for finding this projected mapping.

First, we note that the invertibility of g_i's implies their monotonicity. Here, without loss of generality, we assume that these functions are ascending. In other words, we are looking for ascending functions g_i and matrix $\mathbf{B}$ which minimize $E\{\|\mathbf{y} - \mathbf{B}\mathbf{g}(\mathbf{e})\|^2\}$.

Suppose that we know $\mathbf{x}$. Then, the matrix $\mathbf{B}$ which optimally maps $\mathbf{x}$ to $\mathbf{y}$ is obtained from (16). Knowing $\mathbf{B}$, we can compute $\mathbf{x}$ again by $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$. This automatically defines the functions $x_i = g_i(e_i)$. However, these functions are not necessarily ascending. To insure g_i is ascending, we change the order of the values $x_i(1), x_i(2), \ldots, x_i(T)$ (T is the length of data block) in such a way that for any $e_i(k_1) > e_i(k_2)$ we have $x_i(k_1) > x_i(k_2)$. This may be better explained by its MATLAB code:

```
% Outside the main loop:
   [temp, index_i] = sort(e_i);
% Inside the main loop:
   x_i(index_i) = sort(x_i);
```

This time permutation insures that the function $g_i : e_i \mapsto x_i$ is ascending. It must be noted that we are using a steepest descent iterative algorithm and our initial value for g_i is ascending. Moreover, at each iteration, only a rather small modification is done in the values of $x_i(t)$. Consequently, the above time permutation does not result in a huge modification of the estimated g_i; it must be seen as a manner for preventing the algorithm to produce a non-ascending g_i.

After this new estimation of $\mathbf{x}$, we compute a new $\mathbf{B}$ and then the loop is repeated. This results in the iterative algorithm of Fig. 4 for calculating g_i's and $\mathbf{B}$.

In Section 6 it is shown experimentally that this algorithm converges very fast, typically in 2 to 5 iterations.

```
• Initialization: $\mathbf{x} = \mathbf{e}$.
• Loop:
    1. Let $\mathbf{B} = E\left\{\mathbf{y}\mathbf{x}^T\right\}\left(E\left\{\mathbf{x}\mathbf{x}^T\right\}\right)^{-1}$.
    2. Let $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$.
    3. For $i = 1, \ldots, N$, change the order of $x_i(k)$ such that the function $x_i = g_i(e_i)$ become ascending (see the text).
• Repeat until convergence
```

Fig. 4. Finding the projected PNL mapping.

5.2. The Algorithm

Having an algorithm for calculating the projected mapping, the final MP algorithm can be easily developed.

However, some precautions are required. First note that there is indeterminacies on the means and energies of x_i's and y_i's. To overcome these indeterminacies, in each iteration, their means are removed and their energies are normalized (steps 4 and 7 in Fig. 5).

On the other hand, as we are working with a limited data set, the functions g_i must have a sufficient degree of smoothness. Hence, in each iteration the functions g_i are replaced by the smoothing spline which fits on the data set (e_i, x_i). However, since this smoothing is done in each iteration, the smoothing parameter of this spline (as defined in the MATLAB's spline toolbox) must be chosen very close to 1.

Moreover, two modifications must be done in applying the algorithm of Fig. 4. First, instead of initializing by $\mathbf{x} = \mathbf{e}$, we can use the value of $\mathbf{x}$ obtained in the previous iteration of MP algorithm which is a better initial estimation of $\mathbf{x}$. Secondly, we do not wait for the projection algorithm to converge (even, it is possible that it does not converge, when the outputs cannot expressed as $\mathbf{B}\mathbf{g}(\mathbf{e})$). Instead, we simply repeat the loop for some fixed number of iterations, say 5 or 10 times. In fact, even 1 iteration seems sufficient in many cases, because the whole algorithm is itself iterative, and the value of $\mathbf{x}$ in the previous iteration is used as initial value in the current iteration.

The final separating algorithm is shown in Fig. 5. The value of K in this algorithm (number of repetitions of the internal loop), is a small number, *e.g.* 1 to 10.

6. EXPERIMENTAL RESULTS

Experiment 1. This experiment is to show the efficacy of the algorithm of Fig. 4 for finding the projected mapping in PNL mixtures. Here, two zero mean and unit variance

```
• Initialization: $\mathbf{y} = \mathbf{x} = \mathbf{e}$.
• Loop:
    1. Estimate $\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$, the SFD of $\mathbf{y}$.
    2. Modify the outputs by $\mathbf{y} \leftarrow \mathbf{y} - \mu\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$.
    3. For $k = 1, \dots, K$, do:
        (a) Let $\mathbf{B} = E\{\mathbf{y}\mathbf{x}^T\}\left(E\{\mathbf{x}\mathbf{x}^T\}\right)^{-1}$.
        (b) Let $\mathbf{x} = \mathbf{B}^{-1}\mathbf{y}$.
        (c) For $i = 1, \dots, N$, change the order
            of $x_i(k)$ such that the function $x_i = g_i(e_i)$ be ascending.
    4. For $i = 1, \dots, N$, remove the mean of $x_i$
       and normalize its energy.
    5. For $i = 1, \dots, N$, let $g_i$ be the smoothing
       spline which fits on $(e_i, x_i)$.
    6. Let $\mathbf{y} = \mathbf{B}\mathbf{g}(\mathbf{e})$.
    7. For $i = 1, \dots, N$, remove the mean of $y_i$
       and normalize its energy.
• Repeat until convergence
```

Fig. 5. Projection algorithm for separating PNL mixtures.

sources are mixed by the mixing system:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \tag{17}$$

$$f_1(x) = f_2(x) = 0.1x + \tanh(2x) \tag{18}$$

and then we let $\mathbf{y} = \mathbf{s}$. Now, we apply the algorithm of Fig. 4 for obtaining g_i's and $\mathbf{B}$. We also define the estimation error by $E\{\|\mathbf{y} - \mathbf{B}\mathbf{g}(\mathbf{e})\|^2\}$. Evidently, the optimal error is zero and is obtained for $g_i = f_i^{-1}$ and $\mathbf{B} = \mathbf{A}^{-1}$. Figure 6 shows the variations of this error term versus iterations (one iteration corresponds to computation over $N = 1000$ samples). This experiment shows that the algorithm of Fig. 6 converges very fast, essentially after 2 to 5 iterations.

Experiment 2. In this experiment, the efficacy of the MP algorithm for separating PNL mixtures is verified. Here, we use two uniform random sources with zero means and unit variances. The mixing system is composed of:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \tag{19}$$

$$f_1(x) = f_2(x) = 0.1x + \tanh(2x) \tag{20}$$

The parameters of the separating system are: $\mu = 0.1$, 1000 samples data block, $K = 5$, and the smoothing parameter used for smoothing g_i's is 0.999. For estimating the SFD, we have used a method proposed by D. T. Pham for estimating conditional score functions [17]. Figure 7 shows the

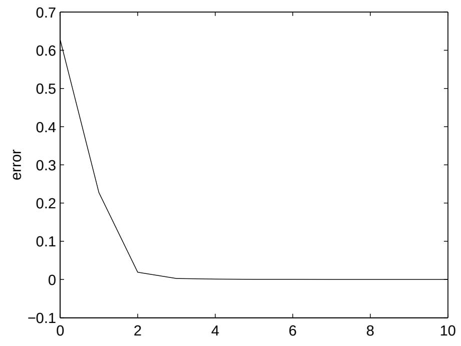

Fig. 6. Estimation error in finding best PNL mapping which maps $\mathbf{x}$ to $\mathbf{s}$.

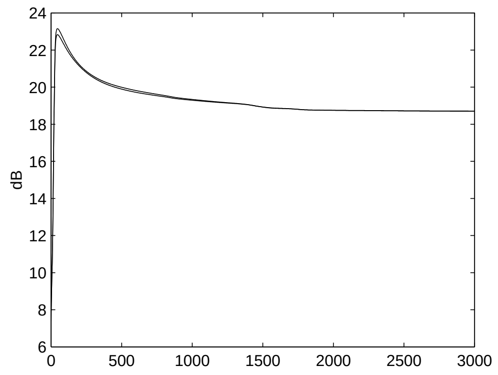

Fig. 7. Averaged output SNR's versus iterations in separating PNL mixtures.

averaged output Signal to Noise Ratios (SNR's) versus iteration, taken over 100 runs of the algorithm. This quantity (in dB) is defined by (assuming there is no permutation):

$$\mathrm{SNR}_i = 10 \log_{10} E\left\{\frac{s_i^2}{(y_i - s_i)^2}\right\} \tag{21}$$

7. CONCLUSION

In this paper, we proposed a Minimization-Projection (MP) approach for blind source separation. This approach can be used in different mixing systems, linear or nonlinear. It is based on minimization of the mutual information of the outputs using a nonparametric gradient of mutual information (Theorem 2).

Moreover, as it has been shown in Theorem 3, the minimization part of this approach has no local minimum, and in which the outputs are directly manipulated regardless of the structure of the separating system. Consequently, it can be conjectured that it is less probable that this approach traps in a local minimum compared to the usual approach of applying a steepest descent algorithm on each parameter of the separating system.

However, this approach requires the estimation of a multidimensional PDF, whose dimension is equal to the source number. It then demands large samples and becomes very expensive and time-consuming when the number of sources is large. Because of this drawback, the application of this approach seems to be restricted to small number of sources, at most 3 or 4.

A. APPENDIX

Proof of Theorem 3. From Theorem 2, for a small μ we have (up to first order terms):

$$I(\mathbf{y}_{n+1}) - I(\mathbf{y}_n) = -\mu E\left\{\|\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y}_n)\|^2\right\} \qquad (22)$$

As $E\left\{\|\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y}_n)\|^2\right\} \geq 0$, then $I(\mathbf{y}_{n+1}) \leq I(\mathbf{y}_n)$, which proves the first part of the theorem.

Now, we prove that the equality holds if and only if the components of $\mathbf{y}_n$ are independent. If the components of $\mathbf{y}_n$ are independent, then from Theorem 1, $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y}) \equiv 0$. Consequently $\mathbf{y}_{n+1} = \mathbf{y}_n$, and hence $I(\mathbf{y}_{n+1}) = I(\mathbf{y}_n) = 0$.

Conversely, suppose that $I(\mathbf{y}_{n+1}) = I(\mathbf{y}_n)$, we have to prove that the components of $\mathbf{y}_n$ are independent. This can be easily proved by contradiction. If the components of $\mathbf{y}_n$ are not independent, then from Theorem 1, $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y})$ cannot be zero everywhere. From continuity $E\left\{\|\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y}_n)\|^2\right\} > 0$. Consequently, from (22) we must have $I(\mathbf{y}_{n+1}) < I(\mathbf{y}_n)$ which is a contradiction. ▲

Remark. $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y})$ does not need to be continuous everywhere. It is sufficient that there exists one region on which $\boldsymbol{\beta}_{\mathbf{y}_n}(\mathbf{y})$ is non-zero and continuous, which holds for any "usual" random vector.

B. REFERENCES

[1] J. Hérault, C. Jutten, and B. Ans, "Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé," in *Actes du Xeme colloque GRETSI*, Nice, France, 20-24 Mai 1985, pp. 1017–1022, (in French).

[2] J. Hérault and C. Jutten, "Space or time adaptive signal processing by neural networks models," in *Intern. Conf. on Neural Networks for Computing*, Snowbird (Utah, USA), 1986, pp. 206–211.

[3] C. Jutten and A. Taleb, "Source separation: From dusk till dawn," in *ICA2000*, Helsinki, Finland, June 2000, pp. 15–26.

[4] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[5] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. on SP*, vol. 44, no. 12, pp. 3017–3030, December 1996.

[6] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings IEEE*, vol. 9, pp. 2009–2025, 1998.

[7] A. Cichocki, R. Unbehauen, and E. Rummert, "Robust learning algorithm for blind separation of signals," *Electronics Letters*, vol. 30, no. 17, pp. 1386–1387, 1994.

[8] T. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comutation*, vol. 7, no. 6, pp. 1004–1034, 1995.

[9] S. I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, 1998.

[10] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiely & Sons, 2001.

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

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

[13] 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.

[14] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Differential of mutual information function," *IEEE Signal Processing Letters*, 2002, submitted.

[15] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Blind separating Convolutive Post-Nonlinear mixtures," in *ICA2001*, San Diego, California, December 2001, pp. 138–143.

[16] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Separating convolutive mixtures by mutual information minimization," in *Proceedings of IWANN'2001*, Granada, Spain, Juin 2001, pp. 834–842.

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