Published in Neurocomputing (Elsevier), vol. 69, pp. 1415-1424, August 2006.

Quasi-optimal EASI algorithm based on the Score Function Difference (SFD)

Samareh Samadi [a], Massoud Babaie-Zadeh [a,1] and Christian Jutten [b,1]

[a] *Advanced Communications Research Institute (ACRI), Electrical Engineering Department, Sharif University of Technology, Tehran, Iran*

[b] *Laboratory of images and signals (CNRS UMR 5083, INPG, UJF), Grenoble, France*

Abstract

Equivariant Adaptive Separation via Independence (EASI) is one of the most successful algorithms for Blind Source Separation (BSS). However, the user has to choose nonlinearities, and usually simple (but non optimal) cubic polynomials are applied. In this paper, the optimal choice of these nonlinearities is addressed. We show that this optimal nonlinearity is the output Score Function Difference (SFD). Contrary to simple nonlinearities usually used in EASI (such as cubic polynomials), the optimal choice is neither component-wise nor fixed: it is a multivariate function which depends on the output distributions. Finally, we derive three adaptive algorithms for estimating the SFD and achieving "quasi-optimal" EASI algorithms, whose separation performance is much better than "standard" EASI and which especially converges for any sources.

1 Introduction

Blind Source Separation (BSS) is an ongoing research topic, which has been considered extensively since mid 80's [1,2]. The goal of BSS is to recover unobserved independent mixed signals from mixtures of them, assuming there is information neither about the original signals, nor about the mixing matrix.

[1] This work has been partially funded by Sharif University of Technology, by French Embassy in Tehran, and by Center for International Research and Collaboration (ISMO).

BSS has received attention because of its theoretical interest and of its potential applications in signal processing such as in speech recognition systems, telecommunications and medical signal processing.

The simplest BSS model is the linear instantaneous model, in which the mixture is supposed to be of the form $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{s} = (s_1, \ldots, s_n)^T$ is the source vector with independent components, $\mathbf{x} = (x_1, \ldots, x_n)^T$ is the observation vector, and $\mathbf{A}$ is the (constant) mixing matrix which is supposed to be an unknown full rank matrix. The basic problem of BSS is to estimate the source components s_i from the observations x_i or, equivalently, to estimate the separating matrix $\mathbf{B}$, which leads to independent estimated sources via $\mathbf{y} = \mathbf{B}\mathbf{x}$. A well-known restriction of this model is that we can only estimate non-Gaussian independent components (in fact, at most one independent component can be Gaussian) [3]. Moreover, the BSS solutions are not unique: neither the energies nor the signs of the independent components can be estimated. In fact, any constant multiplying an independent component could be cancelled by dividing the corresponding column of the mixing matrix $\mathbf{A}$ by the same constant and thus leads to the same observations. Note that no order is defined between independent components [3].

The problem of BSS has been first introduced by Ans, Hérault and Jutten [4] for linear instantaneous mixtures. Then, many researchers have been attracted by the subject, and many other works appeared (see [1,2] for a review and many references).

The BSS algorithms can be divided into two categories: batch algorithms and adaptive algorithms. In batch methods, all the observation samples are first recorded, and then the separation algorithm uses all the recorded data. Conversely, adaptive algorithms update the estimation of the separating system after receiving each new sample (observation), and produce the output immediately. Consequently, adaptive methods are well-suited to real-time applications and allows to track the solution when the mixtures is slowly varying, for instance when the sources are moving.

Among the adaptive BSS methods, the Equivariant Adaptive Separation via Independence (EASI) algorithm [5] has a particular place, due to its equivariant performance, that is, its performance does not depend on the mixing matrix (and consequently on the "hardness" of the mixture). The EASI algorithm consists of two stages: the first stage is a whitening stage, which only provides decorrelated signals; the second stage, modeled by an orthogonal matrix, uses high order statistics for completing the separation. The EASI algorithm is related to a serial updating idea, which requires the computation of the so-called natural or relative gradient [5,6].

The standard EASI algorithm requires nonlinearities whose choice is let to the

user. Usually cubic polynomials are used, but it can be noted that the nonlinear function influences the algorithm stability. In this paper, we propose an optimal choice of these nonlinearities. We show that the optimal nonlinear functions depends on the output distributions and that even simple estimation of this optimal nonlinearities results in a much better separation performance than the usual choice. In the following, the EASI algorithms with standard (and fixed) functions (with optimal functions, respectively) will be called standard (optimal, respectively) EASI algorithms.

The paper is organized as follows. In section 2 we review the standard EASI algorithm. Section 3 reviews the essential materials of the optimal EASI algorithm, based on mutual information minimization. The adaptive implementation of the algorithm is developed in Section 4.1. In Section 4.2, we explain why proposed algorithm can be viewed as an optimal version of EASI . In Section 5 three "quasi-optimal" EASI algorithms based on the adaptive estimation of SFD are proposed and their performances are compared with performance of "standard" EASI.

2 The standard EASI algorithm

In linear instantaneous mixtures $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{s} = (s_1, \ldots, s_N)^T$ denotes the source vector, $\mathbf{x} = (x_1, \ldots, x_N)^T$ is the observation vector, and $\mathbf{A}$ is the regular mixing matrix. Then the objective of a source separation algorithm is estimating a separating matrix $\mathbf{B}$ such that the outputs $\mathbf{y} = \mathbf{B}\mathbf{x}$ be the same as source signals $\mathbf{s}$. It is well-known [3] that, if there is at most one Gaussian source, then the independence of the outputs insures separation of the sources up to a scale and a permutation indeterminacy.

The EASI algorithm [5] is achieved by minimizing a contrast function [2] $\phi(\mathbf{B}) = E\{\mathbf{f}(\mathbf{y})\}$ with respect to $\mathbf{B}$. This leads to the following serial updating algorithm:

$$\mathbf{B}_{n+1} = \Big(\mathbf{I} - \mu \nabla\phi(\mathbf{B}_n)\Big)\mathbf{B}_n \tag{1}$$

where $\nabla\phi(\mathbf{B})$ denotes the relative (or natural) gradient [5,6]:

$$\nabla\phi(\mathbf{B}) = \nabla E\left\{\mathbf{f}(\mathbf{y})\right\} = E\left\{\mathbf{f}'(\mathbf{y})\mathbf{y}^T\right\} \tag{2}$$

Consequently, the stochastic version of (1) becomes:

$$\mathbf{B}_{n+1} = \left(\mathbf{I} - \mu\, \mathbf{g}(\mathbf{y})\mathbf{y}^T\right)\mathbf{B}_n \tag{3}$$

where $\mathbf{g} \triangleq \mathbf{f}'$.

[2] See [3] for the definition of contrast functions.

Implementing EASI requires then to chose a "component-wise" and "fixed" $\mathbf{g}(\cdot)$ (*e.g.* cubic polynomials). Moreover, for avoiding trivial solution $\mathbf{y} = \mathbf{0}$, a normalization term $(\mathbf{I} - \mathbf{y}\mathbf{y}^T)$ is added in equation (3). This makes the final EASI equation more complicated than (3).

3 Mutual Information and its "gradient"

3.1 Mutual Information

Mutual information [7] of the random vector $\mathbf{y} = (y_1, \ldots, y_n)^T$ is one of the widely-used criteria for measuring the independence of random variables y_i. Mutual information (MI) of $\mathbf{y}$ is defined as:

$$\begin{aligned} I(\mathbf{y}) &= D\left(p_{\mathbf{y}}(\mathbf{y}) \parallel \prod_i p_{y_i}(y_i)\right) \\ &= \int_{\mathbf{y}} p_{\mathbf{y}}(\mathbf{y}) \ln \frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_i p_{y_i}(y_i)}\, d\mathbf{y} \\ &= E\left\{\ln \frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_i p_{y_i}(y_i)}\right\} \end{aligned} \tag{4}$$

where $p_{\mathbf{y}}$ and p_{y_i} stands for the Probability Density Function (PDF) of $\mathbf{y}$ and y_i, respectively, and D denotes the Kullback-Leibler divergence. $I(\mathbf{y})$ can also be expressed as [7]:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{y}), \tag{5}$$

where $H(y_i)$ and $H(\mathbf{y})$ denote the marginal and joint differential entropies, respectively.

Mutual information is always non-negative, and is zero if and only if the y_i's are statistically independent [7]. Consequently, the parameters of the separating system may be computed in such a way that the mutual information of the outputs is minimized. This approach has been shown [8] to be asymptotically equivalent to a Maximum Likelihood (ML) estimation of the source signals.

For minimizing the MI, gradient based algorithms can be used. For instantaneous linear mixtures, the gradient of the output mutual information (MI) with respect to the parameters of the separating system is usually computed

from the simplified form of $I(\mathbf{y})$ [3]:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{B}|. \tag{6}$$

Consequently, since $H(\mathbf{x})$ does not depend on the separating system, minimizing $I(\mathbf{y})$ (with respect of the parameters of the separating system) is theoretically equivalent to minimizing $J(\mathbf{y}) = I(\mathbf{y}) + H(\mathbf{x}) = \sum_i H(y_i) - \log|\det \mathbf{B}|$. It is much simpler to use $J(\mathbf{y})$ than $I(\mathbf{y})$ since $J(\mathbf{y})$ does not require the estimation of joint probability density functions. However, as explained in [9], the gradient of $I(\mathbf{y})$ leads to unbiased gradient estimation, contrary to $J(\mathbf{y})$. Moreover, it is much simpler to extend the methods based on minimizing $I(\mathbf{y})$ to more complicated mixtures (*e.g.* convolutive mixtures) than the methods based on minimizing $J(\mathbf{y})$.

For this reason, we suggest to compute the gradient of the complete mutual information (5). Instead of computing the gradient of the MI with respect to the parameters of the separating structure, one can derive the differential of MI, *i.e.* its variation according to a small deviation of its argument (as a non-parametric differential for MI). Such an expression has been recently proposed [10] and requires the definition of multivariate score functions.

3.2 Multivariate Score Functions

In statistics, the score function of a random variable y is the function $\psi_y(y)$ defined as $\psi_y(y) = -p'_y(y)/p_y(y)$, where $p_y(y)$ is the probability density function (PDF) of y. For an N-dimensional random vector $\mathbf{y} = (y_1, \ldots, y_N)^T$, two types of score functions are defined in [10]:

Definition 1 (MSF) *The Marginal Score Function (MSF) of* $\mathbf{y}$ *is the vector of score functions of its components, i.e.:*

$$\boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) = (\psi_1(y_1), \ldots, \psi_N(y_N))^T \tag{7}$$

where:

$$\psi_i(\mathbf{y}) = -\frac{d}{dy_i} \ln p_{y_i}(y_i) = -\frac{p'_{y_i}(y_i)}{p_{y_i}(y_i)} \tag{8}$$

and $p_{y_i}(y_i)$ *is the marginal PDF of* y_i*.*

[3] The result remains true for instantaneous nonlinear mixtures, replacing $\mathbf{B}$ in (6) by the Jacobian matrix of the nonlinear separating system.

Definition 2 (JSF) *The Joint Score Function (JSF) of* $\mathbf{y}$ *is the vector function* $\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y})$*, such that its i-th component is:*

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

where $p_{\mathbf{y}}(\mathbf{y})$ *denotes the joint PDF of* $\mathbf{y}$*.*

Definition 3 (SFD) *The Score Function Difference (SFD) of* $\mathbf{y}$ *is the difference between its MSF and JSF, i.e.:*

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

A few useful properties of SFD have been shown in [11], and are listed below without proof.

Property 1 *The components of a random vector* $\mathbf{y} = (y_1, \ldots, y_N)^T$ *are independent if and only if* $\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y}) \equiv \mathbf{0}$*, that is:*

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

The above property shows that SFD contains all the information about the independence of the components of a random vector.

Property 2 *For a random vector* $\mathbf{y} = (y_1, \ldots, y_N)^T$ *we have:*

$$\beta_i(\mathbf{y}) = \frac{\partial}{\partial y_i} \ln p(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_N | y_i) \tag{12}$$

where $\beta_i(\mathbf{y})$ *denotes the i-th component of the SFD of* $\mathbf{y}$*.*

Property 3 *Let* $\mathbf{y}$ *be a random vector with a density* $p_{\mathbf{y}}$ *and a JSF* $\boldsymbol{\varphi}_{\mathbf{y}}$*. Moreover, let* $f(\mathbf{y})$ *be a multivariate function with continuous partial derivatives and:*

$$\lim_{y_i \to \pm\infty} \int_{y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_N} f(\mathbf{y}) p_{\mathbf{y}}(\mathbf{y})\, dy_1 \cdots dy_{i-1}\, dy_{i+1} \cdots dy_N = 0 \tag{13}$$

Then we have:

$$E\left\{f(\mathbf{y})\varphi_i(\mathbf{y})\right\} = E\left\{\frac{\partial f}{\partial y_i}(\mathbf{y})\right\} \tag{14}$$

The previous property is, in fact, a generalization of a similar property for the score function of a scalar random variable [12,13]. Note that the condition (13) is not restrictive for usual sources: for most physical signals, $p_{\mathbf{y}}(\mathbf{y})$ decreases rapidly when $\|\mathbf{y}\|$ goes to infinity. Especially, (13) holds for bounded signals.

Corollary 1 *For any bounded random vector* $\mathbf{y}$*:*

$$E\left\{\varphi_i(\mathbf{y})\, y_j\right\} = \begin{cases} 1 & ;\text{ if } i = j \\ 0 & ;\text{ if } i \neq j \end{cases} \tag{15}$$

or equivalently:

$$E\left\{\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y})\, \mathbf{y}^T\right\} = \mathbf{I} \tag{16}$$

where $\mathbf{I}$ *denotes the identity matrix.*

Corollary 2 *Suppose we would like to estimate* $\varphi_i(\mathbf{y})$ *by a parametric function* $f(\mathbf{y};\mathbf{w})$*, where* $\mathbf{w} = (w_1, \ldots, w_K)^T$ *denotes the parameter vector, then:*

$$\underset{\mathbf{w}}{\operatorname{argmin}}\, E\left\{\left(\varphi_i(\mathbf{y}) - f(\mathbf{y};\mathbf{w})\right)^2\right\} = \underset{\mathbf{w}}{\operatorname{argmin}}\left\{E\left\{f^2(\mathbf{y};\mathbf{w})\right\} - 2E\left\{\frac{\partial f}{\partial y_i}(\mathbf{y},\mathbf{w})\right\}\right\} \tag{17}$$

This corollary shows a nice property of JSF: even without knowledge about $\varphi_i(\mathbf{y})$, we can design a Minimum Mean Square Error (MMSE) estimator of the JSF.

Property 4 *For a random vector* $\mathbf{y} = (y_1, \ldots, y_N)^T$ *we have:*

$$\psi_i(y) = E\left\{\varphi_i(\mathbf{y}) \mid y_i = y\right\} \tag{18}$$

where φ_i *and* ψ_i *denote the* i*-th component of JSF and MSF of* $\mathbf{y}$*, respectively.*

For clarifying the above property, consider φ_i as a function [4] of y_i, denoted by $\varphi_i(y_i)$. If y_i is independent of the other variables, then $\varphi_i(y_i) = \psi_i(y_i)$. However, if the other variables depend on y_i, $\varphi_i(y_i)$ is no longer equal to $\psi_i(y_i)$, but the above property claims that its "mean" will be still equal to $\psi_i(y_i)$ [11]. In other words, the statistical dependence can introduce some fluctuations in $\varphi_i(y_i)$, but only around its constant "mean".

[4] Strictly speaking, it is a "relation" and not a "function", because for each value of y_i we have several values for φ_i.

Therefore, we can conclude that *the SFD is, in fact, a measure of the variations of JSF* (around its smoothed value).

Property 5 *Let* $\mathbf{y} = \mathbf{B}\mathbf{x}$*, where* $\mathbf{y}$ *and* $\mathbf{x}$ *are random vectors and* $\mathbf{B}$ *is a non-singular square matrix. Then:*

$$\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) = \mathbf{B}^{-T}\boldsymbol{\varphi}_{\mathbf{x}}(\mathbf{x}). \tag{19}$$

3.3 Differential of the mutual information

The "differential" of the mutual information is given by the following theorem [10].

Theorem 1 *Let* $\boldsymbol{\Delta}$ *be a 'small' random vector, with the same dimension as* $\mathbf{x}$*. Then:*

$$I(\mathbf{x}+\boldsymbol{\Delta}) - I(\mathbf{x}) = E\left\{\boldsymbol{\Delta}^T \boldsymbol{\beta}_{\mathbf{x}}(\mathbf{x})\right\} + o(\boldsymbol{\Delta}) \tag{20}$$

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

This theorem points out that SFD can be called the "stochastic gradient" of mutual information [10].

Remark. Equation (20) may be stated in the following form (which is similar to what is done in [14]):

$$I(\mathbf{x}+\mathcal{E}\mathbf{y}) - I(\mathbf{x}) = E\left\{(\mathcal{E}\mathbf{y})^T \boldsymbol{\beta}_{\mathbf{x}}(\mathbf{x})\right\} + o(\mathcal{E}) \tag{21}$$

where $\mathbf{x}$ and $\mathbf{y}$ are bounded random vectors, $\mathcal{E}$ is a matrix with small entries, and $o(\mathcal{E})$ stands for a term that converges to zero faster than $\|\mathcal{E}\|$. This equation is mathematically more sophisticated, because in (20) the term 'small random vector' is somewhat ad-hoc. Conversely, (20) is simpler, and easier to be used in developing gradient based algorithms for optimizing a mutual information.

4 BSS via Mutual Information Minimization

4.1 Optimal EASI

In the separating system $\mathbf{y} = \mathbf{B}\mathbf{x}$, minimizing $I(\mathbf{y})$ with respect to $\mathbf{B}$ (where I stands for mutual information), can be done using the steepest descent

algorithm:

$$\mathbf{B}_{n+1} = \mathbf{B}_n - \mu \left.\frac{\partial I}{\partial \mathbf{B}}\right|_{\mathbf{B}=\mathbf{B}_n} \tag{22}$$

where μ is a small positive constant. However, to design an equivariant algorithm [5], that is, an algorithm whose separation performance does not depend on the conditioning of the mixing matrix, one must use the serial (multiplicative) updating rule:

$$\mathbf{B}_{n+1} = \left(\mathbf{I} - \mu \left[\nabla_{\mathbf{B}} I\right]_{\mathbf{B}=\mathbf{B}_n}\right) \mathbf{B}_n \tag{23}$$

where $\mathbf{I}$ denotes the identity matrix, and $\nabla_{\mathbf{B}} I \triangleq \frac{\partial I}{\partial \mathbf{B}} \mathbf{B}^T$ is the relative (or natural) gradient [5,6] of $I(\mathbf{y})$ with respect to $\mathbf{B}$.

Using theorem 1, $\nabla_{\mathbf{B}} I$ can be easily obtained [10]:

$$\nabla_{\mathbf{B}} I = E\left\{\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^T\right\}. \tag{24}$$

Dropping the expectation operation, the stochastic version of (23) is obtained:

$$\mathbf{B}_{n+1} = \left(\mathbf{I} - \mu\, \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^T\right) \mathbf{B}_n \tag{25}$$

For developing the above algorithm in adaptive form, an adaptive estimation of SFD is required, which will be discussed in Section 5.

4.2 From standard to optimal EASI

A comparison between equations (3) and (25) shows that the algorithm (25) is in fact a special case of EASI (3), in which the nonlinearity $\mathbf{g}(\mathbf{y})$ has been chosen to be equal the SFD of the outputs, *i.e.* $\mathbf{g}(\mathbf{y}) = \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$.

Recall now that the mutual information is an optimal criterion for source separation, in the sense that it asymptotically results in an Maximum Likelihood (ML) estimation of the source signals [8]. Consequently, *the optimal choice (in the ML sense) of the nonlinearity* $\mathbf{g}(\cdot)$ *in EASI is the SFD of the outputs.* Contrary to the "standard" EASI, where $\mathbf{g}(\cdot)$ is a "component-wise" and "fixed" function, here $\mathbf{g}(\mathbf{y}) = \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$ is a multivariate function and depends on the distribution of $\mathbf{y}$.

Moreover, in the "standard" EASI, one must take into account the necessity of existence of a pre-whitening stage, and implement it in the algorithm. This makes the final equation of standard EASI [5] more complicated than (3). On the contrary, using (25), no pre-whitening is required.

However, the above advantages are obtained at the expense of higher computational load: a multivariate nonlinear function (the output SFD) has to be adaptively estimated from output samples.

4.3 Normalization of output variances

There is no scale constraint in the algorithm (25). Consequently, due to the typical scale indeterminacy of BSS, the algorithm does not converge to a unique solution. For overcoming this problem, one can enforce the algorithm to converge to unit variance outputs. In this purpose, we can use some properties of score functions. From the property 3, we can deduce that the diagonal elements of $E\{\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^T\}$ are zero. If we replace the i-th diagonal element of $\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^T$ by $1 - y_i^2$, we force the separating system to provide unit variance outputs. This idea is similar to what is done in [15].

5 Adaptive SFD Estimation

In Section 4.1, we showed that the optimal choice for the nonlinearity $\mathbf{g}(\cdot)$ in EASI is the SFD of the outputs, but this function is usually not known and it must be estimated from the data. Hence, due to this estimation, the final algorithm will be no longer "optimal", but "quasi-optimal". Depending on the accuracy of the SFD estimation, the "quasi-optimal" EASI algorithm achieves various performances. However, in this section, we will show that even relatively simple estimation of SFD in (25) improves the separation performance of "standard" EASI in which $\mathbf{g}(\cdot)$ is a cubic polynomial. Of course, since EASI is an adaptive BSS algorithm, for implementing (25) adaptively, we need an adaptive estimation of SFD.

In this Section, we consider polynomial models for JSF and MSF, with three adaptive estimation algorithms of their parameters.

5.1 SFD estimation by steepest descent

For estimating SFD, one can estimate MSF and JSF independently and then computing the difference. The MSF is simply obtained by estimating the score functions of its components. For any function f with continuous first derivative and random variable y such that $\lim_{u\to\pm\infty} f(u)p_y(u) = 0$ (where $p_y(\cdot)$ is the PDF of y), we have [12,15]:

$$E\left\{f(y)\psi_y(y)\right\} = E\left\{f'(y)\right\} \tag{26}$$

where ψ_y is the score function of the random variable y. Now, let the score function ψ_y be modeled as a linear combination of some basis functions $h_1(y)$, $h_2(y)$, ..., $h_L(y)$:

$$\hat{\psi}_y(y, \mathbf{v}) = \sum_{i=1}^{L} v_i h_i(y) = \mathbf{h}(y)^T \mathbf{v} \tag{27}$$

where $\mathbf{h}(y) \triangleq (h_1(y), \ldots, h_L(y))^T$ and $\mathbf{v} \triangleq (v_1, \ldots, v_L)^T$ is the parameter vector. For computing $\mathbf{v}$, we minimize the mean square error:

$$\mathcal{E}(\hat{\psi}_y(y, \mathbf{v})) \triangleq E\left\{\left(\psi_y(y) - \hat{\psi}_y(y, \mathbf{v})\right)^2\right\}. \tag{28}$$

Expanding the above expression and using (26), one deduces that minimizing $\mathcal{E}$ is equivalent to minimizing:

$$\xi(\hat{\psi}_y(y, \mathbf{v})) \triangleq \frac{1}{2} E\left\{\hat{\psi}_y(y, \mathbf{v})^2\right\} - E\left\{\frac{\partial}{\partial y}\hat{\psi}_y(y)\right\}. \tag{29}$$

For minimizing $\xi(\hat{\psi}_y(y, \mathbf{v}))$ with respect to $\mathbf{v}$, one can use a gradient descent algorithm by dropping the expectation operation:

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \mu \left.\frac{\partial \xi(\hat{\psi}_y(y_n, \mathbf{v}))}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{v}_n} \tag{30}$$

where:

$$\left.\frac{\partial \xi(\hat{\psi}_y(y_n, \mathbf{v}))}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{v}_n} = \mathbf{h}(y_n)\mathbf{h}(y_n)^T \mathbf{v}_n - \left.\frac{\partial \mathbf{h}(y)}{\partial y}\right|_{y=y_n} \tag{31}$$

This method can be easily generalized for estimating JSF. Let $\varphi_i(\mathbf{y})$ be the i-th component of JSF, and denote $\hat{\varphi}_i(\mathbf{y})$ its estimation based on the linear model:

$$\hat{\varphi}_i(\mathbf{y}, \mathbf{w}) = \sum_{i=1}^{L} w_i k_i(\mathbf{y}) = \mathbf{k}(\mathbf{y})^T \mathbf{w} \tag{32}$$

where $k_1(\mathbf{y}), ..., k_L(\mathbf{y})$ are (multivariate) basis functions, and $\mathbf{w}$ is the parameter vector which is computed for minimizing the mean square error:

$$\mathcal{E}(\hat{\varphi}_i(\mathbf{y}, \mathbf{w})) = E\left\{(\varphi_i(\mathbf{y}) - \hat{\varphi}_i(\mathbf{y}, \mathbf{w}))^2\right\}. \tag{33}$$

Applying Property 4 [10], *i.e.* $E\{f(\mathbf{y})\varphi_i(\mathbf{y})\} = E\left\{\frac{\partial}{\partial y_i} f(\mathbf{y})\right\}$ to (33), one proves that the mean square error JSF estimate can be obtained by minimizing:

$$\xi(\hat{\varphi}_i(\mathbf{y}, \mathbf{w})) = \frac{1}{2} E\left\{\hat{\varphi}_i(\mathbf{y})^2\right\} - E\left\{\frac{\partial}{\partial y_i}\hat{\varphi}_i(\mathbf{y}, \mathbf{w})\right\}. \tag{34}$$

Computing the gradient of (34) and dropping the expectation operation, leads to equations similar to (30) and (31):

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \left. \frac{\partial \xi(\hat{\varphi}_i(\mathbf{y}_n, \mathbf{w}))}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_n} \tag{35}$$

where:

$$\left. \frac{\partial \xi(\hat{\varphi}_i(\mathbf{y}_n, \mathbf{w}))}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_n} = \mathbf{k}(\mathbf{y}_n)\mathbf{k}(\mathbf{y}_n)^T \mathbf{w}_n - \left. \frac{\partial \mathbf{k}(\mathbf{y})}{\partial y_i} \right|_{\mathbf{y}=\mathbf{y}_n} \tag{36}$$

Finally, SFD is estimated by calculating the difference of the estimated MSF and JSF.

Polynomial basis functions. As a simple choice for basis functions in (27) and (32), we use a 3rd-order polynomial model for MSF and JSF. In this model, the MSF ($\hat{\psi}_i(y_i, \mathbf{v})$) uses the basis functions:

$$h_1(y) = 1,\ h_2(y) = y,\ h_3(y) = y^2,\ h_4(y) = y^3 \tag{37}$$

and the JSF ($\varphi_i(\mathbf{y})$) is estimated using the basis functions:

$$\begin{aligned} &k_1(y_1, y_2) = 1, \\ &k_2(y_1, y_2) = y_1,\ k_3(y_1, y_2) = y_1^2,\ k_4(y_1, y_2) = y_1^3 \\ &k_5(y_1, y_2) = y_2,\ k_6(y_1, y_2) = y_2^2,\ k_7(y_1, y_2) = y_2^3 \end{aligned}$$

Using these polynomial estimations of MSF and JSF, the final "quasi-optimal" EASI algorithm (using steepest descent estimation of SFD) is summarized in Fig. 1.

Experiment. As an experiment, two zero mean and unit variance independent sources, with normal and uniform distributions are mixed by:

$$\mathbf{A} = \begin{pmatrix} 1 & 0.7 \\ 0.5 & 1 \end{pmatrix}. \tag{38}$$

We compare the separation result of "quasi-optimal" EASI of Fig. 1 with "standard" EASI. In both algorithms, the step sizes (of both SFD estimation and separation algorithms) are 0.001. In "standard" EASI, the componentwise nonlinear function $g_i(y_i) = y_i|y_i|^2$ has been used.

Figure 2 shows the averaged output signal to noise ratios (SNR) over 50 runs

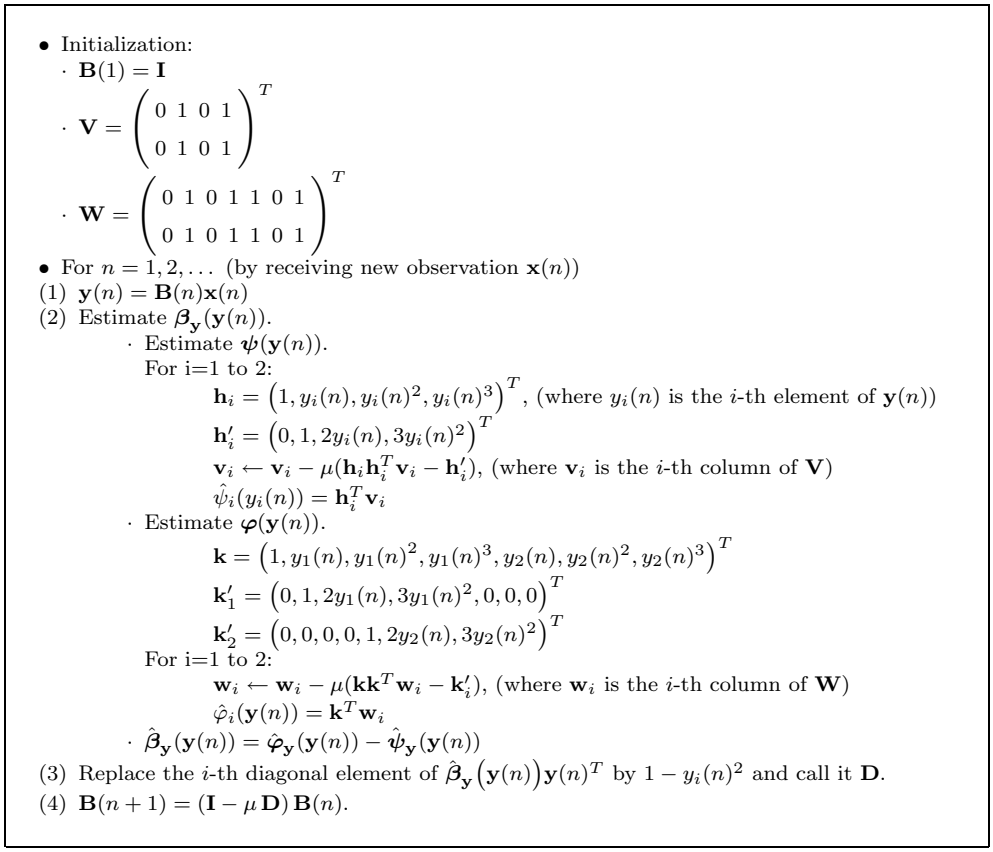

Fig. 1. Quasi-optimal EASI algorithm using steepest descent SFD estimation. In the algorithm, $\mathbf{k}'_i$ denotes the partial derivative $\partial\mathbf{k}/\partial y_i$, $y_i(n)$ the i-th component of $\mathbf{y}$ (*i.e.* the i-th output) at discrete time n, and $\mathbf{B}(n)$ the separation matrix at discrete time n.

of the algorithms. SNR is defined as:

$$\mathrm{SNR} = 10\log_{10}\frac{E\left\{s^2\right\}}{E\left\{(y-s)^2\right\}} \tag{39}$$

where y is the output corresponding to the source s. In this experiment, the score function difference, which is estimated by using very simple basis functions for MSF and JSF estimations, do not provide very good separation performance.

This poor performance of the algorithm can be justified as follows. In fact, in the algorithm, we have two different iterative algorithm working concurrently. The first is (25), in which $\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$ is adaptively estimated using the second iterative algorithm, that is, equations (27), (30), (32) and (35). However, after each iteration of the first iterative algorithm (*i.e.* after each modification of $\mathbf{B}$), since the $\mathbf{y}$ is changed, the previous estimation of $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$ is no longer valid. This results in the poor performance of the proposed algorithm.

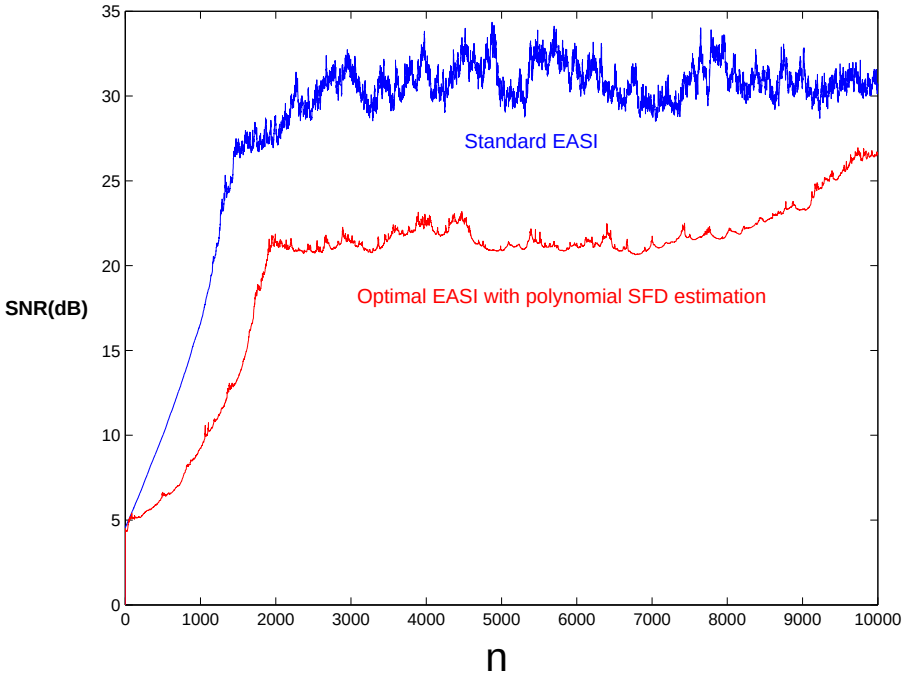

Fig. 2. Output SNRs versus iteration for standard EASI and the quasi-optimal EASI.

To overcome the above problem, two different solutions may be thought. The first solution is to modify the old estimation of $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$ after each modification of the separating matrix $\mathbf{B}$. The second solution is to use a faster converging algorithm for estimating $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$, *e.g.* using Newton like methods instead of steepest descent method used in (30) and (35). In fact, since at each iteration of (25), $\mathbf{B}$ is changed just a little bit, the fast convergence of the second iterative algorithm (for estimating $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$) may solve the problem.

In the next two subsections, we are going to test these two ideas. Although the first approach proposed in the above paragraph may seem better, it is not practically easy to modify the estimation of $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$ after changing $\mathbf{B}$. In fact, we will see that the second approach results in a better performance.

5.2 Improved steepest descent gradient method

To improve the old estimation of $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$ after each modification of $\mathbf{B}$, Property 5 can be used. Let $\hat{\boldsymbol{\varphi}}_{\mathbf{y}}(\cdot)$ be the estimation of $\boldsymbol{\varphi}_{\mathbf{y}}(\cdot)$ after receiving $\mathbf{x}_n$ (using equations (32) and (35)) but before updating $\mathbf{B}_n$ to $\mathbf{B}_{n+1}$ (using (25)). From Property 5, after updating $\mathbf{B}_n$ to $\mathbf{B}_{n+1}$, the estimated $\hat{\boldsymbol{\varphi}}_{\mathbf{y}}(\cdot)$ can be modified as:

$$\hat{\boldsymbol{\varphi}}_{\mathbf{y}}(\cdot) \leftarrow \mathbf{B}_{n+1}^{-T}\mathbf{B}_n^T\hat{\boldsymbol{\varphi}}_{\mathbf{y}}(\cdot) \tag{40}$$

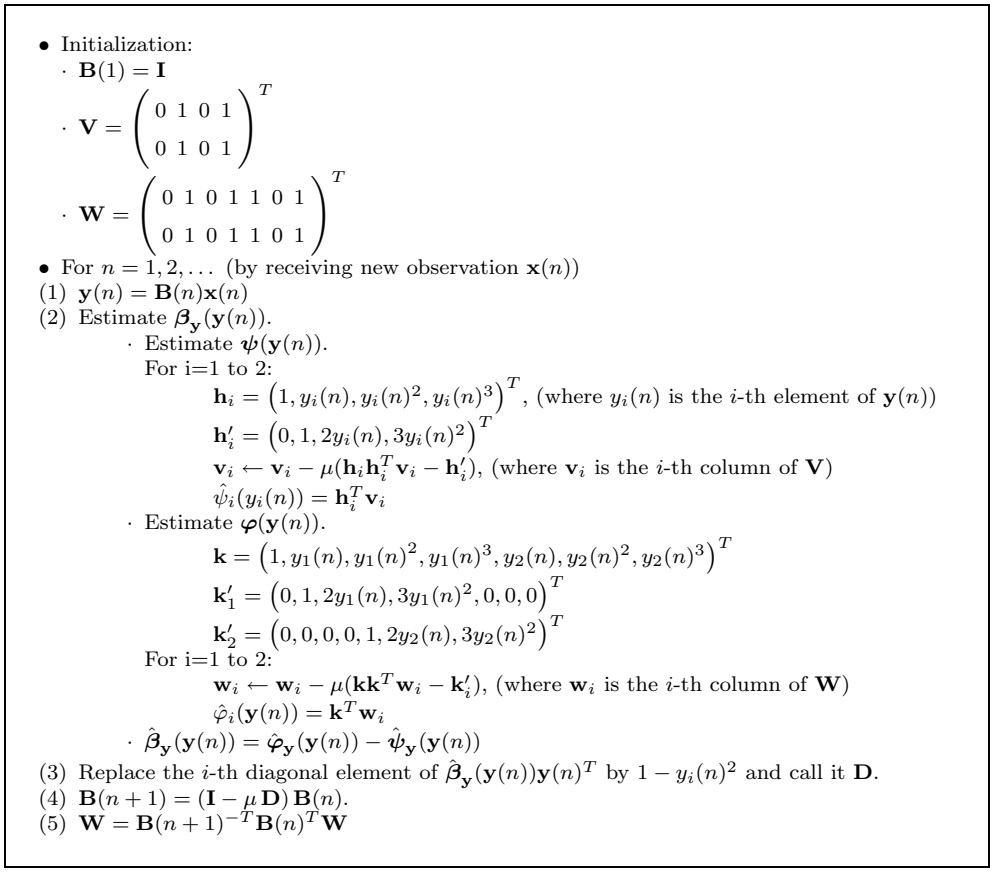

Fig. 3. Quasi-optimal EASI algorithm using modified steepest descent SFD estimation. In the algorithm, $\mathbf{k}'_i$ denotes the partial derivative $\partial\mathbf{k}/\partial y_i$, $y_i(n)$ the i-th component of $\mathbf{y}$ (*i.e.* the i-th output) at discrete time n, and $\mathbf{B}(n)$ the separation matrix at discrete time n.

Using this improvement, the final modified quasi-optimal EASI is shown in Fig. 3. The only difference of this algorithm with the algorithm of Fig. 1 is adding the 5th step.

Experiment. The algorithm of Fig. 3 has been applied using the same signals and mixture as in the first experiment. The results obtained with this new algorithm are shown in Figure 4. It points out that enhancing the JSF estimation accuracy without enhancing the MSF estimation accuracy does not improve the performance. Of course, because independence is achieved when the score function difference vanishes, it requires an accurate estimation of SFD, *i.e.* of both MSF and JSF. Moreover, since there is no simple relation between MSF of observations and outputs (like Property 5 for JSF), improving the estimation of MSF after updating $\mathbf{B}$ is not easy.

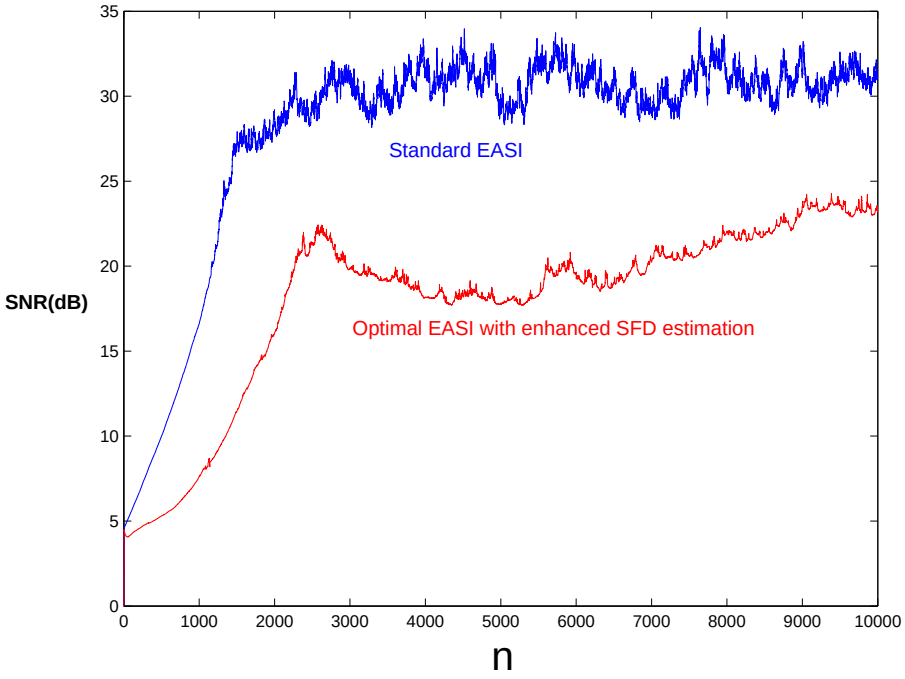

Fig. 4. Output SNRs versus iteration for standard EASI and quasi-optimal EASI with enhanced SFD estimation.

5.3 Newton's method

The second idea mentioned at the end of Section 5.1 was speeding up the convergence of the iterative algorithm of estimation of $\boldsymbol{\beta}_{\mathbf{y}}(\cdot)$. To do this, we propose to use the Newton's method for minimizing the cost functions $\xi(\hat{\psi}(y,\mathbf{v}))$ and $\xi(\hat{\varphi}_i(\mathbf{y},\mathbf{w}))$ in equations (29) and (33).

The Newton's method for minimizing $\xi(\hat{\psi}(y,\mathbf{v}))$ in (29) is written as:

$$\mathbf{v} \leftarrow \mathbf{v} - \mu E\left\{\frac{\partial^2 \xi(\hat{\psi}(y,\mathbf{v}))}{\partial \mathbf{v}^2}\right\}^{-1} E\left\{\frac{\partial \xi(\hat{\psi}(y,\mathbf{v}))}{\partial \mathbf{v}}\right\} \tag{41}$$

where:

$$\frac{\partial \xi(\hat{\psi}(y,\mathbf{v}))}{\partial \mathbf{v}} = \mathbf{h}(y)\mathbf{h}(y)^T\mathbf{v} - \frac{\partial \mathbf{h}(y)}{\partial y} \tag{42}$$

and:

$$\frac{\partial^2 \xi(\hat{\psi}(y,\mathbf{v}))}{\partial \mathbf{v}^2} = \mathbf{h}(y)\mathbf{h}(y)^T. \tag{43}$$

Like the steepest descent method, we simply ignore the second expectation operation in (41). However, the first expectation operation cannot be ignored, because the matrix $\mathbf{h}\mathbf{h}^T$ is not full-rank and hence is not invertible. To have

a simple estimation of this expectation, we use:

$$\hat{E}\{\mathbf{h}\mathbf{h}^T\} \leftarrow \alpha\hat{E}\{\mathbf{h}\mathbf{h}^T\} + (1-\alpha)\mathbf{h}\mathbf{h}^T \tag{44}$$

where $0 < \alpha < 1$.

This method can be easily generalized for estimating JSF, by minimizing $\xi(\hat{\varphi}_i(\mathbf{y}, \mathbf{w}))$ (33). Following similar calculation as above, we obtain equations similar to (41), (42) and (43), replacing $\xi(\hat{\psi}(y, \mathbf{v}))$ by $\xi(\hat{\varphi}_i(\mathbf{y}, \mathbf{w}))$, and the basis $\mathbf{h}$ by $\mathbf{k}$.

Finally, SFD is estimated by calculating the difference of the estimated MSF and JSF.

The final algorithm is summarized in Fig. 5. It can be seen that the difference of this algorithm with the algorithm of Fig. 1 is only in its second step (estimation of SFD).

Experiment. We repeat the previous experiments with Newton's method, using same signals and mixtures. The step size of the Newton algorithm is 0.1 and the value of α in (44) is 0.9. Figure 6 shows that the "quasi-optimal" EASI Newton algorithm has a better performance than the "standard" EASI.

6 Conclusion

In this paper, we proved that the optimal non-linearities of EASI algorithm are related to the Score Function Difference (SFD). Although the theoretical algorithm can be seen as an optimal version of the EASI algorithm, its adaptive implementation requires adaptive implementation of score functions, and the practical algorithm is only quasi-optimal. Three adaptive quasi-optimal algorithms for blind separating linear instantaneous mixtures have been proposed, each based on a different adaptive estimation of SFD, modeled by polynomials. The experimental results show that quasi-optimal EASI can achieve better performance than standard EASI, but requires an accurate SFD estimation. However, the method has the great advantage to converge for any sources, contrary to standard EASI whose convergence assumes a condition on the source statistics [5].

We could improve the SFD estimation by estimating first the JSF and then integrate the JSF for computing the MSF. Contrary to the three methods used in this paper, which estimate independently SFD and JSF, this method leads

- Initialization:
  - $\mathbf{B}(1) = \mathbf{I}$
  - $\mathbf{V} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}^T$
  - $\mathbf{W} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}^T$
- For $n = 1, 2, \ldots$ (by receiving new observation $\mathbf{x}(n)$)

(1) $\mathbf{y}(n) = \mathbf{B}(n)\mathbf{x}(n)$

(2) Estimate $\hat{\boldsymbol{\beta}}_{\mathbf{y}}(\mathbf{y}(n))$.

- Estimate $\hat{\boldsymbol{\psi}}(\mathbf{y}(n))$.

  For i=1 to 2:

  $\mathbf{h}_i = \left(1, y_i(n), y_i(n)^2, y_i(n)^3\right)^T$, (where $y_i(n)$ is the i-th element of $\mathbf{y}(n)$)

  $\mathbf{h}'_i = \left(0, 1, 2y_i(n), 3y_i(n)^2\right)^T$

  $\hat{E}\{\mathbf{h}_i\mathbf{h}_i^T\} \leftarrow \alpha\hat{E}\{\mathbf{h}_i\mathbf{h}_i^T\} + (1-\alpha)\mathbf{h}_i\mathbf{h}_i^T$

  $\mathbf{v}_i \leftarrow \mathbf{v}_i - \mu\left(\hat{E}\{\mathbf{h}_i\mathbf{h}_i^T\}\right)^{-1}(\mathbf{h}_i\mathbf{h}_i^T\mathbf{v}_i - \mathbf{h}'_i)$, (where $\mathbf{v}_i$ is the i-th column of $\mathbf{V}$)

  $\hat{\psi}_i(y_i(n)) = \mathbf{h}_i^T\mathbf{v}_i$
- Estimate $\hat{\boldsymbol{\varphi}}(\mathbf{y}(n))$.

  $\mathbf{k} = \left(1, y_1(n), y_1(n)^2, y_1(n)^3, y_2(n), y_2(n)^2, y_2(n)^3\right)^T$

  $\mathbf{k}'_1 = \left(0, 1, 2y_1(n), 3y_1(n)^2, 0, 0, 0\right)^T$

  $\mathbf{k}'_2 = \left(0, 0, 0, 0, 1, 2y_2(n), 3y_2(n)^2\right)^T$

  $\hat{E}\{\mathbf{k}\mathbf{k}^T\} \leftarrow \alpha\hat{E}\{\mathbf{k}\mathbf{k}^T\} + (1-\alpha)\mathbf{k}\mathbf{k}^T$

  For i=1 to 2:

  $\mathbf{w}_i \leftarrow \mathbf{w}_i - \mu\left(\hat{E}\left\{\mathbf{k}\mathbf{k}^T\right\}\right)^{-1}(\mathbf{k}\mathbf{k}^T\mathbf{w}_i - \mathbf{k}'_i)$, (where $\mathbf{w}_i$ is the i-th column of $\mathbf{W}$)

  $\hat{\varphi}_i(\mathbf{y}(n)) = \mathbf{k}^T\mathbf{w}_i$
- $\hat{\boldsymbol{\beta}}_{\mathbf{y}}(\mathbf{y}(n)) = \hat{\boldsymbol{\varphi}}_{\mathbf{y}}(\mathbf{y}(n)) - \hat{\boldsymbol{\psi}}_{\mathbf{y}}(\mathbf{y}(n))$

(3) Replace the i-th diagonal element of $\hat{\boldsymbol{\beta}}_{\mathbf{y}}(\mathbf{y}(n))\mathbf{y}(n)^T$ by $1 - y_i(n)^2$ and call it $\mathbf{D}$.

(4) $\mathbf{B}(n+1) = (\mathbf{I} - \mu\,\mathbf{D})\,\mathbf{B}(n)$.

Fig. 5. Quasi-optimal EASI algorithm using Newton SFD estimation. In the algorithm, $\mathbf{k}'_i$ denotes the partial derivative $\partial\mathbf{k}/\partial y_i$, $y_i(n)$ the i-th component of $\mathbf{y}$ (*i.e.* the i-th output) at discrete time n, and $\mathbf{B}(n)$ the separation matrix at discrete time n.

to unbiased estimations [9]. The batch algorithm, proposed in [16], has to be modified for providing an adaptive version.

Moreover, since SFD has been successfully used in separating convolutive and non-linear mixtures [11,16], it can be conjectured that this method could be generalized for separating more complicated (than linear instantaneous) mixing models. Such a generalization is currently under study.

The main drawback of this method is that it requires the estimation of multivariate score functions (which are related to joint PDFs). This estimation becomes too difficult, and requires a lot of data, when the dimension (*i.e.* number of sources) grows. Practically, this method is suitable only up to 3 or 4 sources. However, it could be overcame, by decomposing the separation matrix $\mathbf{B} = \mathbf{U}\mathbf{W}$, where $\mathbf{W}$ is a whitening matrix and $\mathbf{U}$ is a rotation matrix. Parameterizing $\mathbf{U}$ as a product of Givens's rotation matrices, one should

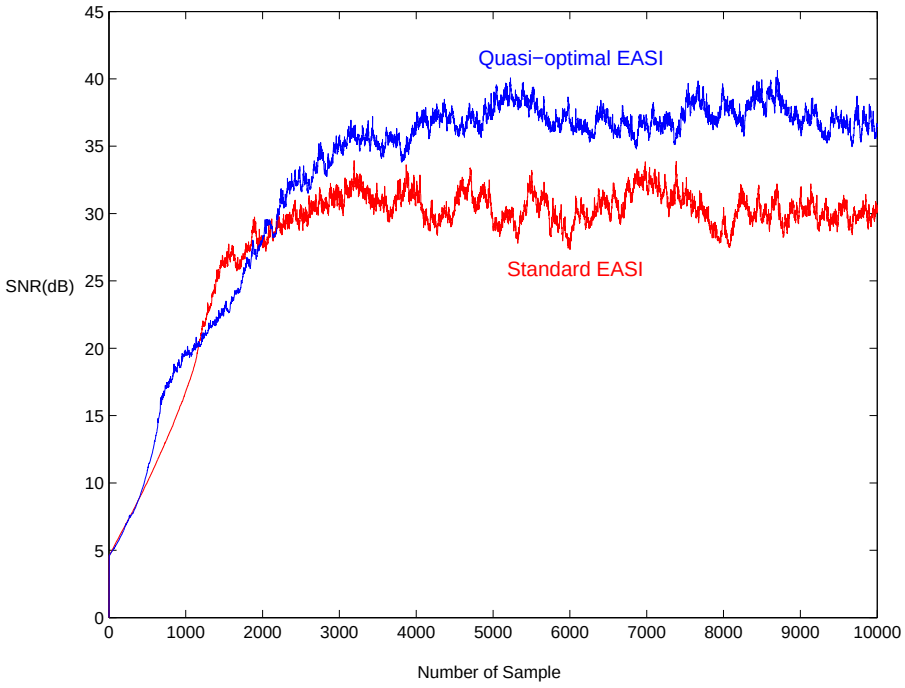

Fig. 6. Output SNRs versus iteration for standard EASI and quasi-optimal EASI with Newton's method for SFD estimation.

estimate the Givens's matrices by pairwise independence, which should only require bivariate SFD estimation. This idea is still under study.

References

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

[2] A. Cichocki, S.-I. Amari, Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications, John Wiley and Sons, 2002.

[3] P. Comon, Independent component analysis, a new concept?, Signal Processing 36 (3) (1994) 287–314.

[4] B. Ans, J. Hérault, C. Jutten, Adaptive neural architectures: Detection of primitives, in: Proceedings of COGNITIVA'85, Paris, France, 1985, pp. 593–597.

[5] J.-F. Cardoso, B. Laheld, Equivariant adaptive source separation, IEEE Trans. on SP 44 (12) (1996) 3017–3030.

[6] S. I. Amari, Natural gradient works efficiently in learning, Neural Computation 10 (1998) 251–276.

[7] T. M. Cover, J. A. Thomas, Elements of Information Theory, Wiley Series in Telecommunications, 1991.

[8] J.-F. Cardoso, Blind signal separation: statistical principles, Proceedings IEEE 9 (1998) 2009–2025.

[9] S. Achard, D. T. Pham, C. Jutten, Criteria for blind source separation in post nonlinear mixtures with mutual information, Signal Processing .

[10] M. Babaie-Zadeh, C. Jutten, K. Nayebi, Differential of mutual information, IEEE Signal Processing Letters 11 (1) (2004) 48–51.

[11] M. Babaie-Zadeh, C. Jutten, A general approach for mutual information minimization and its application to blind source separation, Signal Processing .

[12] D. D. Cox, A penalty method for nonparametric estimation of the logarithmic derivative of a density function, Ann. Instit. Statist. Math. 37 (1985) 271–288.

[13] A. Taleb, C. Jutten, Batch algorithm for source separation in postnonlinear mixtures, in: Proceedings of ICA'99, Aussois, France, 1999, pp. 155–160.

[14] D. T. Pham, Mutual information approach to blind separation of stationary sources, IEEE Transactions on Information Theory 48 (7) (2002) 1–12.

[15] A. Taleb, C. Jutten, Entropy optimization, application to blind source separation, in: Proceedings of ICANN'97, Lausanne, Switzerland, 1997, pp. 529–534.

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