Scientia Iranica, Vol. 9, No. 4, pp 409–418

SCIENTIA IRANICA

Research Note

Using Multivariate Score Functions in Source Separation: Application to Post Non-Linear Mixtures

M. Babaie-Zadeh[1,2], C. Jutten[1] and K. Nayebi*

In this paper, Joint Score Function (JSF) and Marginal Score Function (MSF) are first defined. It is then pointed out that their difference (SFD) can be treated as the stochastic gradient of mutual information and, hence, can be used in minimizing the mutual information with gradient-based methods. An estimator for SFD is then presented, based on nonlinear regression by means of spline smoothing. It is shown that SFD can be used to obtain a new non parametric algorithm for source separation in Post Non-Linear (PNL) mixtures. The method is very general and can be extended to convolutive mixtures, which is currently being studied.

INTRODUCTION

Blind Source Separation (BSS) or Independent Component Analysis (ICA) is a basic problem in signal processing, which has been intensively considered during the last decade. The goal of BSS is to separate the mixture of a number of independent signals when there is neither information about the source signals nor about the mixing process (hence the term Blind). This problem arises in many different applications, for example, in removing the effects of blinking from the brain EEG signal [1], in separating artifacts from the ECG signal [2], or in enhancing noisy speech signals to improve the quality of speech recognition systems [3]. The problem was first introduced by J. Hérault and C. Jutten [4] for linear mixtures and research has been continued by many others [5-16].

In the linear case, the mixture is assumed to be of the form:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1}$$

where $\mathbf{s} = (s_1, \ldots, s_N)^T$ is the vector of unobserved source signals, which are assumed to be zero mean and independent signals, $\mathbf{x} = (x_1, \ldots, x_N)^T$ is the observation vector and $\mathbf{A}$ is the unknown mixing matrix. Note that in Equation 1, the number of sensors is assumed to be equal to the number of sources. To find the original sources, from observation vector $\mathbf{x}$ only and without any other prior knowledge regarding the mixture or the source signals (except the source independence), one must estimate a matrix $\mathbf{B}$, such that:

$$\mathbf{y} = \mathbf{B}\mathbf{x} = (\mathbf{B}\mathbf{A})\mathbf{s}, \tag{2}$$

has mutually independent components.

In fact, in this linear case, it has been shown [5] that if matrix $\mathbf{A}$ is nonsingular and the sources are zero mean independent signals with, at most, one Gaussian source, the components of $\mathbf{y}$ are independent if, and only if, $\mathbf{BA} = \mathbf{DP}$, where $\mathbf{P}$ and $\mathbf{D}$ denote a permutation and full rank diagonal matrix, respectively (i.e., the sources are recovered up to a scale and a permutation indeterminacy). This property is a direct consequence of the Darmois-Skitovic theorem [17-19]. Therefore, it is said that the linear mixtures are separable and $\mathbf{B}$ is called a separating matrix.

One generalization to this problem concerns separation of nonlinear mixtures, in which observations are a nonlinear transform of the unobserved sources: $\mathbf{x} = F(\mathbf{s})$. For separating the sources, one must estimate an inverse function G such that $\mathbf{y} = H(\mathbf{s}) = G[F(\mathbf{s})]$ would be an estimate of the sources. Unfortunately, it can be seen that, in general, nonlinear mixtures are not separable [14,20], i.e., the independence of the components of $\mathbf{y}$ does not insure that $y_i = f_i(s_{\sigma(i)})$, where $\sigma(\cdot)$ is a permutation.

1. *Institut National Polytechnique de Grenoble (INPG), Laboratoire des Images et des Signaux (LIS), Grenoble, France.*
2. *Department of Electrical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*
*. *Corresponding Author, Department of Electrical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*

However, among the nonlinear systems, there is a special important class, called Post Non-Linear (PNL) mixture [21,14], which is separable. Figure 1 shows this type of mixture, where the observations are the signals e_i, satisfying:

$$e_i = f_i\Big(\sum_{j=1}^{N} a_{ij}s_j\Big) \quad , \quad i = 1, \ldots, N. \tag{3}$$

As can be seen in Figure 1, this model corresponds to the case where the mixing itself is linear, while the sensors introduce nonlinear distortions. Also, it must be noted that the model has been restricted to a case where the number of sensors is equal to the number of sources and there is no (additive) noise in the system.

To separate these mixtures, one must first compensate for the nonlinearities by functions g_i, and then separate the resulting (linear) mixture. For PNL mixtures, it has been shown [14] that under mild conditions (mainly, differentiability and invertibility of f_i's) the output signals (y_i's) are independent if, and only if, the functions $g_i \circ f_i$ are linear and $\mathbf{B}$ is the separating matrix (i.e., $\mathbf{BA} = \mathbf{DP}$, where $\mathbf{P}$ and $\mathbf{D}$ are a permutation and a full rank diagonal matrix, respectively).

Separating algorithms for PNL mixtures were first presented by. Taleb et al. [10,21-23] and have since been studied by some other researchers (e.g., see [24,25], chapter 6 of [26], chapter 4 of [27] and chapter 17 of [28]). In almost all of these methods, the separation criterion is the mutual information of the outputs (see next section). In other words, the parameters of separator systems are determined to minimize $I(\mathbf{y})$. One key relation in these algorithms is the multiplicative Relation 1, which leads to the following equation:

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{|\det \mathbf{B}|}, \tag{4}$$

and, hence:

$$H(\mathbf{y}) = H(\mathbf{x}) + \ln|\det \mathbf{B}|\,, \tag{5}$$

where H denotes Shannon entropy. However, for convolutive mixtures, where the components of the mixing matrix are linear filters instead of scalars, there is not a multiplicative relation like Equation 1 and, hence, there is no simple equation as in Equations 4 and 5. As a result, it is extremely difficult to generalize these methods to the convolutive case and therefore, is an important drawback to previously known methods. As far as recorded, no such mixture has yet been addressed in the literature.

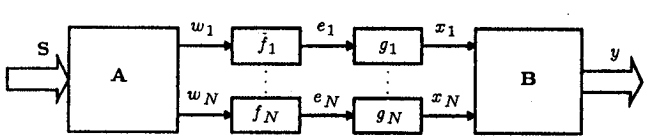

Figure 1. Mixing-separating system.

In this paper, like previous methods, mutual information of outputs has been used as the separation criterion. However, instead of working with entropies, mutual information is dealt with directly. To do this, two functions called Joint Score Function (JSF) and Marginal Score Function (MSF) are first defined. Then, it is shown that their difference, Score Function Difference (SFD), can be seen as the gradient of mutual information. Having this result, this gradient can be dealt with directly to minimize the mutual information of the outputs. However, for using SFD, it must first be estimated from the data and, hence, some parts of the paper have been devoted to developing an estimator for SFD. As the multiplicative Relation 1 has not been used in these developments, the method is very general and can be extended for separating other forms of mixtures. Currently, work is being carried out in continuing its generalization to include convolutive PNL mixtures.

This paper is organized as follows. First, the concept of multivariate score functions are introduced, then the mutual information and its gradient are presented. This gradient is then used for developing estimation equations and the new non-parametric algorithm for separating PNL mixtures. Finally, a few experiments are provided.

JOINT SCORE FUNCTIONS

Here, the concepts of Joint Score Function (JSF), Marginal Score Function (MSF) and Score Function Difference (SFD) are introduced. First, the following definition is recalled from the statistics theory.

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

$$\psi_x(x) = \frac{d}{dx}\ln p_x(x) = \frac{p'_x(x)}{p_x(x)}, \tag{6}$$

where $p_x(x)$ denotes the Probability Density Function (PDF) of x.

Now, let $\mathbf{x} = (x_1, \ldots, x_N)^T$ be an N-dimensional random vector. In this paper, two different forms of score functions are defined.

Definition 2 (MSF)
Marginal Score Function (MSF) of $\mathbf{x}$ is a vector containing the score functions of its components, i.e.:

$$\psi_{\mathbf{x}}(\mathbf{x}) = (\psi_1(x_1), \ldots, \psi_N(x_N))^T, \tag{7}$$

where the ith component of $\psi_{\mathbf{x}}(\mathbf{x})$ is:

$$\psi_i(x_i) = \frac{d}{dx_i} \ln p_{x_i}(x_i), \tag{8}$$

and $p_{x_i}(x_i)$ denotes the PDF of the random variable x_i.

Definition 3 (JSF)
Joint Score Function (JSF) of $\mathbf{x}$ is the vector function $\varphi_{\mathbf{x}}(\mathbf{x})$, such that its ith component is:

$$\varphi_i(\mathbf{x}) = \frac{\partial}{\partial x_i} \ln p_{\mathbf{x}}(\mathbf{x}) = \frac{\frac{\partial}{\partial x_i} p_{\mathbf{x}}(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})}, \tag{9}$$

where $p_{\mathbf{x}}(\mathbf{x})$ is the joint PDF of the random vector $\mathbf{x}$.

Note that the simple score function (Equation 6) for scalar random variable is a particular case of Equation 8 or 9. Clearly, for scalar random variables, MSF and JSF are equal.

Definition 4 (SFD)
Score Function Difference (SFD) of $\mathbf{x}$ is the difference between its JSF and MSF, i.e.:

$$\beta_{\mathbf{x}}(\mathbf{x}) = \varphi_{\mathbf{x}}(\mathbf{x}) - \psi_{\mathbf{x}}(\mathbf{x}). \tag{10}$$

The following theorem relates the SFD of a random vector to the independence of its components.

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

$$\varphi_{\mathbf{x}}(\mathbf{x}) = \psi_{\mathbf{x}}(\mathbf{x}). \tag{11}$$

Proof
Here, the theorem is only proven for the two-dimensional case. Its generalization to higher dimensions is obvious.

If the components of $\mathbf{x}$ are independent, then Equation 11 is evident. Conversely, suppose that Equation 11 holds, then it can be proven that the components of $\mathbf{x}$ are independent. From Equation 11, the following is obtained:

$$\frac{\partial}{\partial x_1} \ln p_{\mathbf{x}}(x_1, x_2) = \frac{\partial}{\partial x_1} \ln p_{x_1}(x_1). \tag{12}$$

Integrating both sides of this equation, with respect to x_1, results in:

$$\ln p_{\mathbf{x}}(x_1, x_2) = \ln p_{x_1}(x_1) + \ln q(x_2)$$

$$\Rightarrow p_{\mathbf{x}}(x_1, x_2) = p_{x_1}(x_1) q(x_2). \tag{13}$$

Integrating both sides of this equation, with respect to x_1, from $-\infty$ to $+\infty$, provides:

$$q(x_2) = p_{x_2}(x_2), \tag{14}$$

which proves the theorem.

MUTUAL INFORMATION AND ITS GRADIENT

By definition, the components of a random vector $\mathbf{x} = (x_1, \ldots, x_N)^T$ are independent if, and only if:

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{N} p_{x_i}(x_i). \tag{15}$$

A usual measure of statistical dependence between the components of $\mathbf{x}$ is the mutual information $I(\mathbf{x})$ [29], which is nothing but the Kullback-Leibler divergence between $p_{\mathbf{x}}(\mathbf{x})$ and $\prod_{i=1}^{N} p_{x_i}(x_i)$, i.e.:

$$I(\mathbf{x}) = D\left(p_{\mathbf{x}}(\mathbf{x}) \,\|\, \prod_i p_{x_i}(x_i)\right)$$

$$= \int_{\mathbf{x}} p_{\mathbf{x}}(\mathbf{x}) \ln \frac{p_{\mathbf{x}}(\mathbf{x})}{\prod_i p_{x_i}(x_i)} d\mathbf{x}. \tag{16}$$

It is well known that $I(\mathbf{x})$ is always nonnegative and vanishes if, and only if, the x_i's are independent. Thus, for separable mixtures, $I(\mathbf{y})$, which is an independence criterion for the estimated sources, can be used as a criterion for estimating the separating structure parameters. In other words, for separating the sources, the separating system parameters must be obtained, such that the mutual information of the outputs reaches its minimum (zero). This can be done with a gradient-based algorithm, which requires computation of the mutual information variation when its argument has changed by a small random vector $\boldsymbol{\Delta}$.

Now, the following theorem [30] is introduced:

Theorem 2
Let $\boldsymbol{\Delta}$ be a 'small' random vector, with the same dimension as $\mathbf{x}$. Then:

$$I(\mathbf{x} + \boldsymbol{\Delta}) - I(\mathbf{x}) = E\left\{\boldsymbol{\Delta}^T \beta_{\mathbf{x}}(\mathbf{x})\right\}$$

$$+ \text{higher order terms in } \boldsymbol{\Delta}, \tag{17}$$

where $\beta_{\mathbf{x}}$ denotes the SFD of $\mathbf{x}$.

Note that for any differentiable multivariate function f:

$$f(\mathbf{x} + \boldsymbol{\Delta}) - f(\mathbf{x}) = \boldsymbol{\Delta}^T (\nabla f(\mathbf{x}))$$

$$+ \text{higher order terms in } \boldsymbol{\Delta}, \tag{18}$$

where $\nabla f(\mathbf{x})$ is the gradient of f. A comparison between Equations 17 and 18 shows that SFD can be called the stochastic gradient of the mutual information.

ESTIMATING EQUATIONS

Conditional Mean

Let x and y be two joint random variables. Here, the problem of estimating the conditional mean (or nonlinear regression curve [31]) $g(x) = E\{y|x\}$ from their joint observed samples $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$ is considered. This is required for estimating SFD (Equation 26) and functions g_i (Equation 36). Since these functions are nonlinear, a problem of nonlinear regression is encountered.

The problem of smooth curve fitting consists in estimating a 'smooth enough' function $y = g(x)$, only from the data points $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$. This can be done by minimizing the expression:

$$\lambda \sum_{i=1}^{N} (y_i - g(x_i))^2 + (1-\lambda) \int (g''(x))^2 \, dx, \qquad (19)$$

where $0 \leq \lambda \leq 1$ is the smoothing parameter; the greater λ, the better fitting to data, and the smaller λ, the smoother function. It can be proved that the solution of this problem is a cubic spline, i.e., a 3rd order polynomial between two successive x_i's [32,33]. This spline is called smoothing spline (In MATLAB's spline toolbox, the function 'csaps' can be used to calculate the smoothing spline).

It can be seen, heuristically, that in the curve fitting problem, the best value which can be assigned to $g(x)$ is $E\{y|x\}$. Hence, for estimating the regression curve, one can use the smoothing spline which fits on the data (other nonlinear regressors could also be used, but are not optimal with respect to the cost function defined in Expression 19). In fact, this is a well-known and widely used method in statistics theory [34]. Figure 2 shows a sample case, in which x and y are two independent random variables with uniform distribution on $(-1, 1)$, having 1000 joint samples of them being observed.

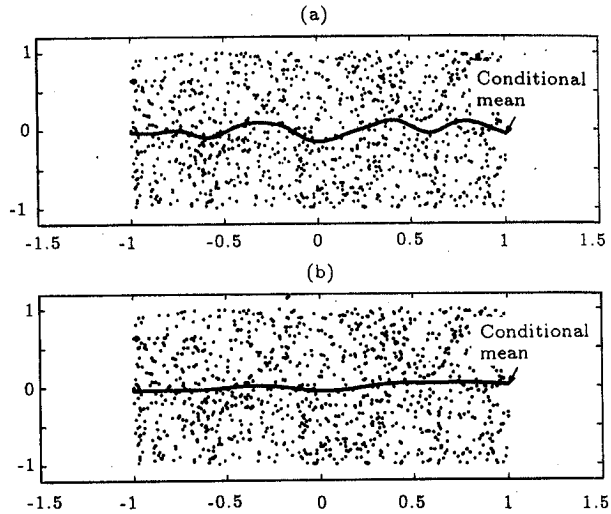

Figure 2. Calculating conditional mean by means of curve fitting: a) $\lambda = 0.99$, b) $\lambda = 0.8$.

Kernel Estimators for JSF and MSF

Kernel estimators are well-known for estimating the PDF of a random variable [35]. These estimators can also be used for estimating score functions.

Let $x_1, \ldots, x_T$ be T observed samples of a scalar random variable x. A kernel $K_\sigma(x)$ is a function which is symmetric around zero and integrates to 1. Thus, any symmetric probability density function (for example, a Gaussian distribution) can be a kernel function. One also defines a smoothing parameter σ, called the bandwidth, which is derived from the variance σ^2 of the kernel. Then, the estimated PDF of x is:

$$\hat{p}_x(x) = \frac{1}{T} \sum_{t=1}^{T} K_\sigma(x - x_t). \qquad (20)$$

Increasing σ implies increasing the degree of smoothness of the estimated PDF: If it is too small, the PDF estimate is very noisy; if it is too large, one obtains a rough estimate of the shape of the kernel. Moreover, if $K_\sigma(x)$ is differentiable, a kernel estimator for the score function of x will be [36]:

$$\hat{\psi}_x(x) = \frac{\hat{p}'_x(x)}{\hat{p}_x(x)} = \frac{\sum_{t=1}^{T} K'_\sigma(x - x_t)}{\sum_{t=1}^{T} K_\sigma(x - x_t)}. \qquad (21)$$

This method is easily applicable to the N-dimensional case, provided that one has a sufficient number of samples. In fact, due to the curse of dimensionality, this is generally impossible over dimensions 4 or 5 [35]. Let $K(\mathbf{x})$ be an N-variate PDF, for example, the multivariate Gaussian:

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} (\det \Sigma)^{1/2}} \exp\left\{\frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x}\right\}. \qquad (22)$$

Then, the estimated PDF of $\mathbf{x}$ from the observed data set $\mathbf{x}_1, \ldots, \mathbf{x}_T$ will be:

$$\hat{p}_{\mathbf{x}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} K(\mathbf{x} - \mathbf{x}_t), \qquad (23)$$

and the ith component of JSF is:

$$\hat{\varphi}_i(\mathbf{x}) = \frac{\frac{\partial}{\partial x_i} \hat{p}_{\mathbf{x}}(\mathbf{x})}{\hat{p}_{\mathbf{x}}(\mathbf{x})} = \frac{\sum_{t=1}^{T} \frac{\partial K}{\partial x_i}(\mathbf{x} - \mathbf{x}_t)}{\sum_{t=1}^{T} K(\mathbf{x} - \mathbf{x}_t)}. \qquad (24)$$

Estimating SFD

In [37], an efficient SFD estimator has been proposed for separating convolutive mixtures. However, this estimator does not lead to satisfactory results in PNL mixtures.

SFD can also be estimated by:

$$\hat{\beta}(\mathbf{x}) = \hat{\varphi}(\mathbf{x}) - \hat{\psi}(\mathbf{x}), \tag{25}$$

where $\hat{\varphi}(\mathbf{x})$ and $\hat{\psi}(\mathbf{x})$ are kernel estimates of JSF and MSF, respectively. Unfortunately, like the previous method, this method does not lead to good results in PNL mixtures.

In this section, another SFD estimate is proposed which leads to a good performance in separating PNL mixtures. The estimate is based on the following theorem.

Theorem 3

Let $\varphi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \ldots, \varphi_N(\mathbf{x}))^T$ and $\psi(\mathbf{x}) = (\psi_1(x_1), \ldots, \psi_N(x_N))^T$ be the JSF and MSF of the random vector $\mathbf{x}$, respectively. Then:

$$\psi_i(x) = E\{\varphi_i(\mathbf{x}) \mid x_i = x\}. \tag{26}$$

Proof

Without loss of generality, let $i = 1$, then:

$$\begin{aligned} & E\{\varphi_1(\mathbf{x}) \mid x_1 = x\} \\ &= \int_{x_2,\ldots,x_N} \varphi_1(\mathbf{x}) p(x_2, \ldots, x_N \mid x_1) dx_2 \cdots dx_N \\ &= \int_{x_2,\ldots,x_N} \frac{\frac{\partial}{\partial x_1} p(\mathbf{x})}{p(\mathbf{x})} \cdot \frac{p(\mathbf{x})}{p_1(x_1)} dx_2 \cdots dx_N \\ &= \frac{1}{p_1(x_1)} \cdot \frac{\partial}{\partial x_1} \int_{x_2,\ldots,x_N} p(\mathbf{x}) dx_2 \cdots dx_N \\ &= \frac{1}{p_1(x_1)} \cdot \frac{\partial}{\partial x_1} p_1(x_1) \\ &= \psi_1(x_1), \end{aligned} \tag{27}$$

which proves the theorem.

Remember that x_i's are independent if, and only if, $\varphi_i = \psi_i$. Then, the above theorem claims that if the components of $\mathbf{x}$ are statistically dependent, then φ_i is not equal to ψ_i, while it is equal to its mean. In other words, statistical dependence introduces fluctuations in JSF around its constant mean.

As an example, let $x_1 = s_1$ and $x_2 = s_2 + ks_1$ be two mixtures of two independent random variables s_1 and s_2 with uniform PDFs on $[-0.5, 0.5]$ and consider their JSF and MSF. When $k = 0$, x_1 and x_2 are independent and $\varphi_1 = \psi_1$. When k varies, ψ_1 remains unchanged (because it does not depend on k), but φ_1 changes. Figure 3 shows the plot of the kernel estimates of φ_1 and ψ_1 versus x_1, for $k = 0.5$. From this figure, it can easily be seen that introducing statistical dependencies results in variations of φ_1 around its mean, as expected following Theorem 3.

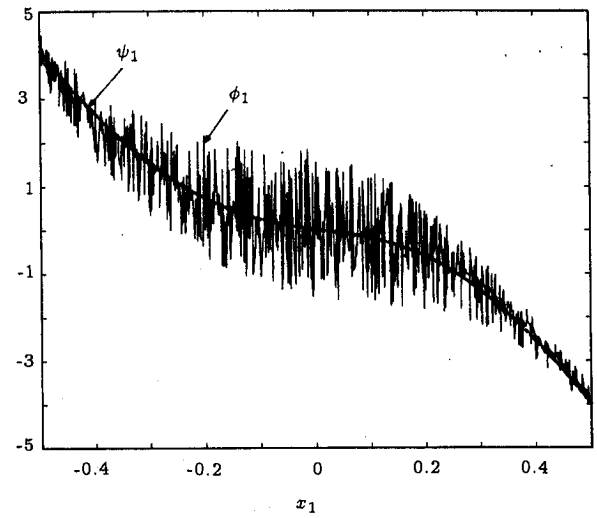

Figure 3. Kernel estimates of φ_1 and ψ_1 for two dependent variables.

The above paragraph shows that SFD (β_i) is, in fact, a measure of the variations of φ_i around its mean. These variations are due to the statistical dependence of the other components of $\mathbf{x}$.

Hence, to estimate β_i, a kernel estimate of φ_i is first computed, then a smooth enough curve (a smoothing spline) is fitted on the data set (x_i, φ_i) for estimating ψ_i. Finally, by subtracting φ_i from its smoothed version, one obtains an estimate of β_i.

Estimating Equations for the Separating System

To achieve source separation, the mutual information $I(\mathbf{y})$ must be minimized. Hence, the gradients of I, with respect to the system parameters, $\mathbf{B}$ and g_i's, must be determined. Consider now some small perturbations in these parameters of the form:

$$\mathbf{B} \to \tilde{\mathbf{B}} = \mathbf{B} + \mathbf{\Delta} \cdot \mathbf{B} = (\mathbf{I} + \mathbf{\Delta})\mathbf{B}, \tag{28}$$

$$g_i \to \tilde{g}_i = g_i + \epsilon_i \circ g_i, \tag{29}$$

where $\mathbf{\Delta}$ is a *small* matrix, ϵ_i's are *small* functions, and $\mathbf{I}$ denotes the identity matrix. Since $g_i(e_i) = x_i$, Equation 29 becomes:

$$\tilde{x_i} = x_i + \epsilon_i(x_i) = x_i + \delta_i, \tag{30}$$

where $\delta_i \triangleq \epsilon_i(x_i)$.

Now, from Theorem 2, the following theorem is derived.

Theorem 4

The variation of $I(\mathbf{y})$, with respect to the variations $\mathbf{\Delta}$ and ϵ_i, is:

$$\begin{aligned} I(\tilde{\mathbf{y}}) - I(\mathbf{y}) &= E\{\beta_{\mathbf{y}}(\mathbf{y})^T \cdot \mathbf{\Delta} \cdot \mathbf{y}\} \\ &\quad + E\{\beta_{\mathbf{y}}(\mathbf{y})^T \cdot \mathbf{B} \cdot \delta\} + \text{Higher order terms}, \end{aligned} \tag{31}$$

where β denotes the SFD of $\mathbf{y}$.

The proof results from direct calculation and is discarded here.

After simple calculations, the first right-side term of Equation 31 becomes:

$$\sum_i \sum_j \Delta_{ij} E\{\beta_i(\mathbf{y}) y_j\}. \tag{32}$$

Hence, the relative (or natural) gradient [7,13] of I with respect to $\mathbf{B}$ is:

$$\nabla_{\mathbf{B}} I = E\{\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^T\}. \tag{33}$$

It is easy to prove that this leads to the same equations as shown for linear instantaneous mixtures [21], i.e.:

$$(\nabla_{\mathbf{B}} I)_{ij} = \begin{cases} -E\{\psi_i(y_i) y_j\} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases} \tag{34}$$

Defining $\boldsymbol{\alpha}(\mathbf{y}) \triangleq \mathbf{B}^T \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})$, the second right-side term of Equation 31 becomes:

$$\begin{aligned} & E\left\{\boldsymbol{\delta}^T \mathbf{B}^T \boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\right\} \\ &= E\left\{\boldsymbol{\delta}^T \boldsymbol{\alpha}(\mathbf{y})\right\} \\ &= \sum_i E\{\epsilon_i(x_i)\alpha_i(\mathbf{y})\} \\ &= \sum_i E\{\epsilon_i(x_i) E\{\alpha_i(\mathbf{y}) \mid x_i\}\} \\ &= \sum_i \int_{-\infty}^{\infty} \epsilon_i(x) E\{\alpha_i(\mathbf{y}) \mid x_i = x\} p_{x_i}(x) dx. \end{aligned} \tag{35}$$

Hence, the gradient of I, with respect to the function $g_i(x)$, via the weighting function $p_{x_i}(x)$ can be defined as:

$$h_i(x) \triangleq (\nabla_{g_i} I)(x) = E\{\alpha_i(\mathbf{y}) \mid x_i = x\}. \tag{36}$$

THE ALGORITHM

For separating post nonlinear mixtures, the steepest descent algorithm has been used. In each iteration, the SFD of the outputs is computed using the method presented previously. Then, the following iterations are used:

$$\mathbf{B} = (\mathbf{I} - \mu_1 \nabla_{\mathbf{B}} I)\mathbf{B}, \tag{37}$$

$$x_i = x_i - \mu_2 h_i(x_i), \tag{38}$$

where $\nabla_{\mathbf{B}} I$ and $h_i(x)$ are computed using Equation 33 (or Equation 34) and Equation 36, respectively. μ_1 and μ_2 are two small positive constants and, in Equation 36, the conditional mean is computed using smoothing splines.

However, there are a few indeterminacies that need more attention. The first indeterminacies are the mean and variance of x_i's. To cancel these indeterminacies, x_i's are normalized at each iteration so that their means and variances are equal to zeros and ones, respectively. This prevents the algorithm from diverging.

The second indeterminacy is the output variance. If this indeterminacy is not considered, then matrix $\mathbf{B}$ can converge to zero matrix, generating the trivial solution $\mathbf{y} = \mathbf{0}$. To prevent this situation, one can normalize the output variance at each iteration. It is also possible to normalize the output by replacing the main diagonal of $\nabla_{\mathbf{B}} I$ by $\text{diag}(\sigma^2_{y_1} - 1, \ldots, \sigma^2_{y_N} - 1)$, as proposed by. Taleb and Jutten [10]. (Note that, as can be seen from Equation 34, the main diagonal of $\nabla_{\mathbf{B}} I$ is zero.) This will cause the algorithm to generate unit variance outputs.

The convergence is achieved after about 100 iterations. However, the performance (output SNR or crosstalk) depends on the sample size. Moreover, the values of $g_i(x_i(n))$ are computed independently for different values of n, without any attention to the smoothness of the function g_i. This will result in fluctuating (and, thus, neither continuous nor invertible) functions g_i's. This is in contradiction to the main assumptions on the functions f_i's (and, hence, g_i's), i.e., continuity and invertibility [14]. For solving this problem, smoothing splines are used in each iteration for estimating the functions g_i's (although, it results in slower convergence). The smoothing parameter (λ) (see (19)) must be chosen near 1 (for example $\lambda = 0.9999$), for achieving a good estimation of the inverses of f_i's.

Figure 4 gives the resulting separation algorithm.

EXPERIMENTS

In this section, some computer simulations are presented for a PNL mixture of two sources. The independent sources are sine and triangle waveforms, with an irrational frequency ratio. Figure 5a shows the joint distribution of the sources and points out their independence. The sources are mixed with the mixing matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}. \tag{39}$$

The sensor nonlinearities are:

$$\begin{aligned} f_1(x) &= \frac{1}{10}(x + x^3), \\ f_2(x) &= \frac{3}{10}x + \tanh 3x, \end{aligned} \tag{40}$$

- Initialization:
  1. $\mathbf{B} = \underline{\mathrm{I}}$
  2. $\mathbf{x} = \underline{\mathrm{e}}$
- Loop:
  1. Compute outputs by $\mathbf{y} = \mathbf{B}\mathbf{x}$.
  2. Estimate $\beta_{\mathbf{y}}(\mathbf{y})$ (SFD of $\mathbf{y}$).
  3. For $i = 1, \ldots, N$, estimate $h_i(x_i)$, using Equation 36 and smoothing splines.
  4. For $i = 1, \ldots, N$, modify x_i by $x_i = x_i - \mu_2 h_i(x_i)$.
  5. For $i = 1, \ldots, N$, do a smoothing process on x_i, using a smoothing spline.
  6. For $i = 1, \ldots, N$, normalize x_i by $x_i = \frac{x_i - \hat{\mu}_{x_i}}{\hat{\sigma}_{x_i}}$.
  7. Normalize $\mathbf{B}$ by:
$$\mathbf{B} = \mathbf{B} \begin{pmatrix} \hat{\sigma}_{x_1} & & \\ & \ddots & \\ & & \hat{\sigma}_{x_N} \end{pmatrix}$$
  8. Estimate $\mathbf{D} = \nabla_{\mathbf{B}} I$, using Equation 33.
  9. Replace the main diagonal of $\mathbf{D}$ by $\mathrm{diag}(\sigma^2_{y_1} - 1, \ldots, \sigma^2_{y_N} - 1)$.
  10. Modify $\mathbf{B}$ by $\mathbf{B} = (\mathbf{I} - \mu_1 \mathbf{D})\mathbf{B}$.
- Repeat until convergence

Figure 4. The separation algorithm for PNL mixtures.

and, as can be seen in Figure 5b, they generate highly nonlinear mixtures. The observation signals are shown in Figure 6.

A 1000-point size data sample is used. The algorithm step sizes, μ_1 and μ_2, are equal to 0.3. Gaussian kernels, with zero mean and $\sigma = 0.4$, are used for estimation of the JSF. Smoothing splines are used for estimating SFD and $h_i(x_i)$, with smoothing parameters (λ) equal to 0.9 and 0.1, respectively. Finally the smoothing spline parameter for g_i's is $\lambda = 0.9999$.

The separation results can be viewed in Figures 7 to 10. Figure 7a represents the joint distribution of (x_1, x_2) and shows that the sensor nonlinearities have been compensated. Figure 7b shows the joint distribution of outputs and points out their independence. The compensated functions $g_1 \circ f_1$ and $g_2 \circ f_2$ are sketched in Figure 8. The output signals can be viewed in Figure 9, which are good estimates of the sources. Finally, Figure 10 shows the output Signal to Noise Ratios (SNRs), defined by (assuming no permutation):

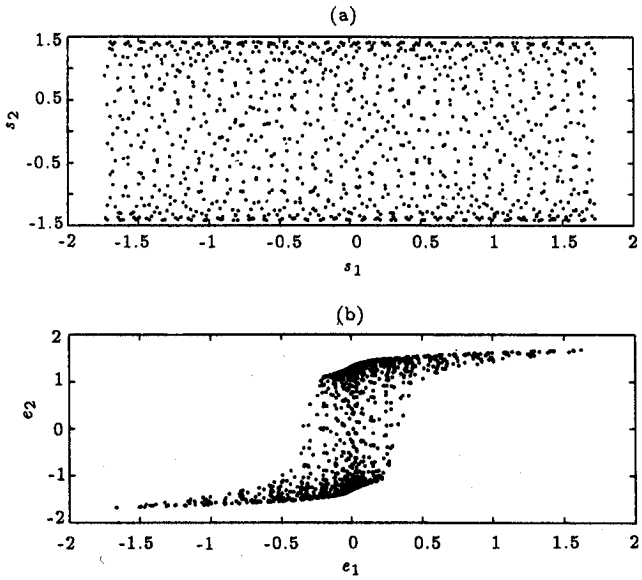

Figure 5. a) Source and b) Observed joint distributions.

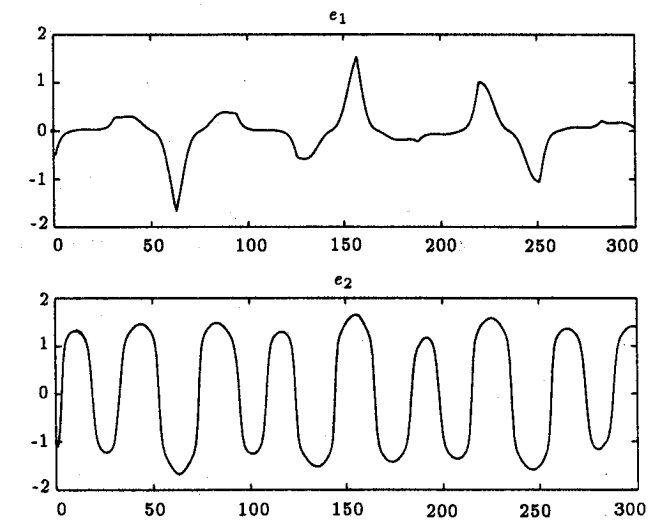

Figure 6. Observed signals.

$$\mathrm{SNR}(y_i, s_i) = 10 \log_{10} \frac{E\{s_i^2\}}{\{(s_i - y_i)^2\}}, \quad i = \{1, 2\}, \tag{41}$$

In the second experiment, the mixture of three independent identically distributed (iid) signals are considered with uniform distribution. The mixing matrix is:

$$\mathbf{A} = \begin{pmatrix} 1 & 0.3 & 0.3 \\ 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{pmatrix}. \tag{42}$$

The first two sensor nonlinearities are as Equation 40 and for the third nonlinearity $f_3(x) = f_2(x)$. For separating this mixture, a data block of 1000 samples

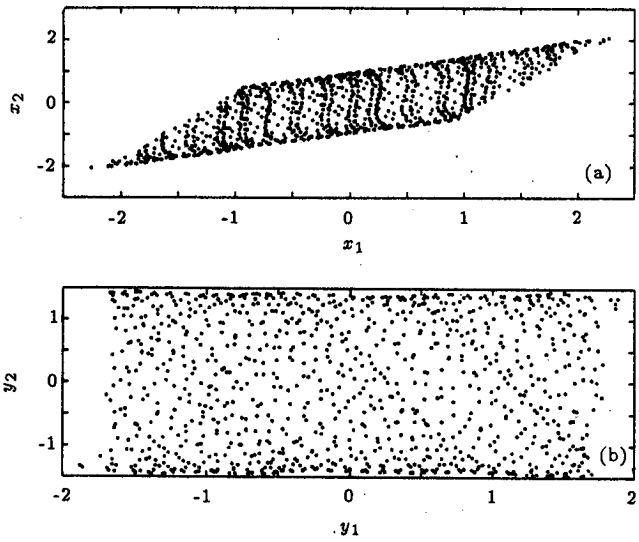

Figure 7. a) Joint distribution of (x_1, x_2) shows that the nonlinearities have been compensated; b) Joint distribution of outputs points out their independence.

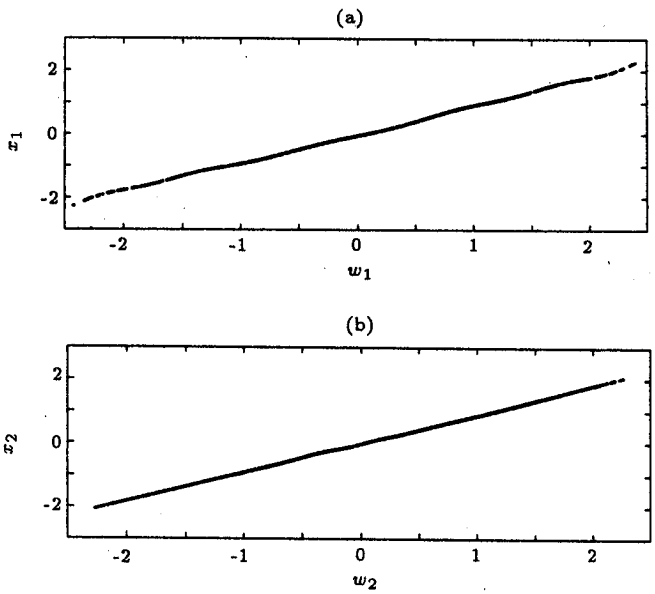

Figure 8. Compensated functions: a) $g_1 \circ f_1$ and b) $g_2 \circ f_2$.

is used. The parameters are $\mu_1 = \mu_2 = 0.2$ and Gaussian kernels were used with $\sigma = 0.2$ for estimating JSF. The smoothing spline parameters were $\lambda = 0.4$ for estimating MSF, $\lambda = 0.1$ for estimating h_i's and $\lambda = 0.9999$ for smoothing g_i's. Figure 11 shows the output SNR versus the iterations.

A similar experiment has been performed with a mixture of four sources. The algorithm still works, but is very time consuming. In fact, the estimation of the multivariate score functions (4-D) becomes tricky and requires a large sample size. However, one achieves between 18 dB to 22 dB after 150 iterations.

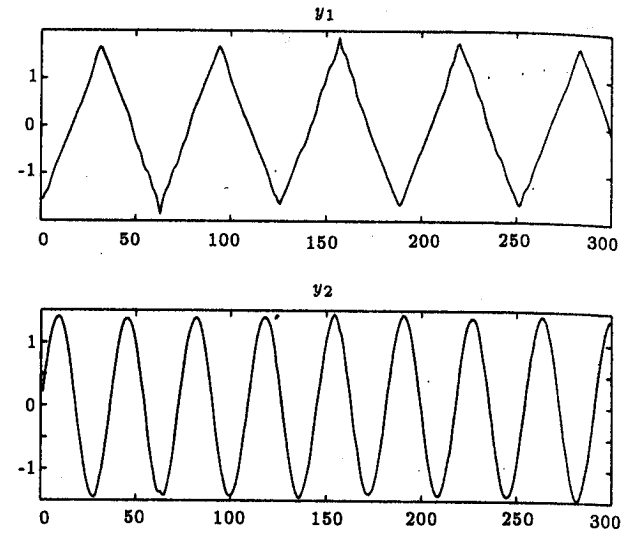

Figure 9. Estimated sources.

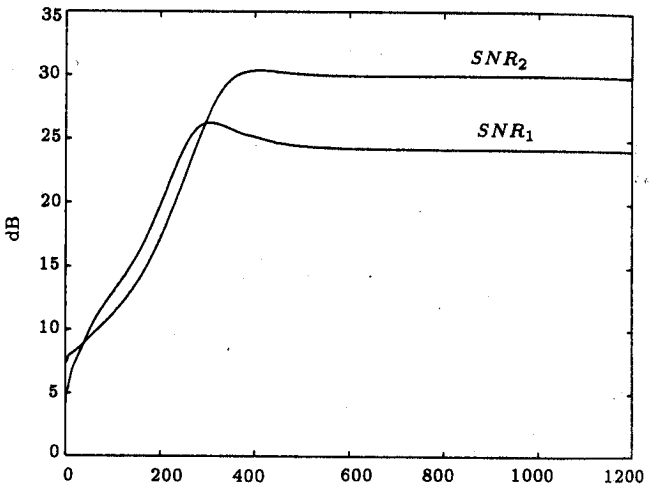

Figure 10. Output SNRs versus iteration for the first experiment.

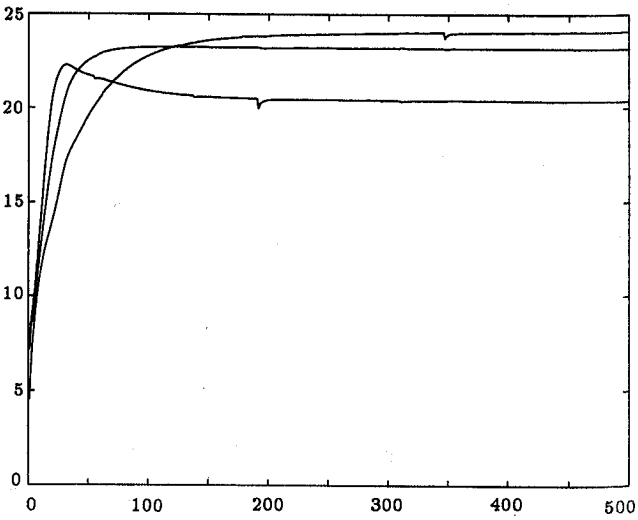

Figure 11. Output SNRs versus iteration for separating three random sources.

These experiments indicate the separation ability of the algorithm, even for hard nonlinear mixtures.

CONCLUSION

In this paper, multivariate score functions have been introduced, i.e., JSF and MSF. Then, it could be seen that their difference (SFD) is the stochastic gradient of the mutual information, which can be used to design gradient based methods for minimizing the mutual information. Also, an estimator is proposed for SFD by using smoothing splines. After deriving the estimating equations and putting all of it together, a new separation algorithm is developed for PNL mixtures. These experiments show the quality of the proposed method.

The main advantage of this new method is its generality. In fact, the multiplicative Relations 1 and 4 are not relied upon. Such relations do not exist in convolutive cases, where the components of the mixing matrix are linear filters instead of some scalars. Therefore, this new method can be generalized to convolutive mixtures, as well as other types of mixtures. Currently, these generalizations are being studied by the authors.

In fact, the main drawback of the method is that it requires estimation of multivariate PDF: it demands a large enough sample size, growing very rapidly with the dimension, i.e., with the source number. Thus practically, the method is tractable for a small number of sources, up to three or four.

ACKNOWLEDGMENT

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

REFERENCES

1. Jung, T.P. et al. "Removing electroencephalographic artifcats by blind source separation", *Psychophysiology*, **37**, pp 167–178 (Mar. 2000).
2. Barros, A.K., Mansour, A. and Ohnishi, N. "Removing artifacts from ECG signals using independent components analysis", *Neurocomputing*, **221**, pp 173–186 (1998).
3. Nguyen Thi, H.L., Jutten, C., Kabre, H. and Caelen, J. "Separation of sources: A method for speech enhancement", *Applied Signal Processing*, **3**, pp 177–190 (1996).
4. Hérault, J. and Jutten, C. "Space or time adaptive signal processing by neural networks models", *Intern. Conf. on Neural Networks for Computing*, Snowbird, Utah, USA, pp 206–211 (1986).
5. Comon, P. "Independent component analysis, a new concept?", *Signal Processing*, **36**(3), pp 287–314, (1994).
6. Bell, T. and Sejnowski, T. "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation*, **7**(6), pp 1004–1034 (1995).
7. Cardoso, J.F. and Laheld, B. "Equivariant adaptive source separation", *IEEE Trans. on SP*, **44**(12), pp 3017–3030 (Dec. 1996).
8. Mansour, A. and Jutten, C. "A direct solution for blind separation of sources", *IEEE Trans. on Signal Processing*, **44**, pp 746–748 (1996).
9. Karhunen, J. "Neural approaches to independent component analysis and source separation", *ESANN'96, European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp 249–266 (Apr. 1996).
10. Taleb, A. and Jutten, C. "Entropy optimization, application to blind source separation", *ICANN*, Lausanne, Switzerland, pp 529–534 (Oct. 1997).
11. Hyvärinen, A. and Oja, E. "A fast fixed point algorithm for independent component analysis", *Neural Computation*, **9**, pp 1483–1492 (1997).
12. Oja, E. "The nonlinear PCA learning rule in independent component analysis", *Neurocomputing*, **17**, pp 25–45 (1997).
13. Amari, S.I. "Natural gradient works efficiently in learning", *Neural Computation*, **10**, pp 251–276 (1998).
14. Taleb, A. and Jutten, C. "Source separation in post nonlinear mixtures", *IEEE Transactions on Signal Processing*, **47**(10), pp 2807–2820 (1999).
15. Lee, T.W., Lewicki, M.S., Girolami, M. and Sejnowski, T.J. "Blind source separation of more sources than mixtures using overcomplete representations", *IEEE Signal Processing Letters*, **4**(4), pp 87–90 (Apr. 1999).
16. Pham, D.T. "Blind separation of instantaneous mixture of sources based on order statistics", *IEEE Trans. on SP*, **48**, pp 363–375 (2000).
17. Darmois, G. "Analyse générale des liaisons stochastiques", *Rev. Inst. Intern. Stat.*, in French, **21**, pp 2–8 (1953).
18. Skitovic, V.P. "Linear forms of independent random variables and the normal distribution law", *Izvestiya Akademii Nauk SSSR. Seriya Matematiceskaya*, **18**, in Russian, pp 185–200 (1954).
19. Kagan, A.M., Linnik, Y.N. and Rao, C.R., *Characterization Problems in Mathematics Statistics*, John Wiley & Sons (1973).
20. Hyvärinen, A. and Pajunen, P. "Nonlinear independent component analysis: Existence and uniqueness results", *Neural Networks*, **12**, pp 429–439 (1999).
21. Taleb, A. and Jutten, C. "Non-linear source separation: The post non-linear mixtures", *ESANN'97*, Bruges, Belgium, pp 279–284 (Apr. 1997).

22. Taleb, A., Jutten, C. and Olympieff, S. "Source separation in post nonlinear mixtures: An entropy-based algorithm", *ICASSP'98*, Seattle, WA, USA, pp 2089–92 (1998).
23. Taleb, A. and Jutten, C. "Batch algorithm for source separation in postnonlinear mixtures", *ICA'99*, Aussois, France, pp 155–160 (Jan. 1999).
24. Achard, S., *Initiation a la Séparation Aveugle de Sources dans des Mélanges Post non Linéaires*, DEA de l'INP de Grenoble, in French (June 2000).
25. Tan, Y., Wang, J. and Zurada, J. "Nonlinear blind source separation using a radial basis function network", *IEEE Transactions on Neural Network*, **12**, pp 124–134 (Jan. 2001).
26. Lee, T.-W. *Independent Component Analysis-Theory and Applications*, Kluwer (1998).
27. Roberts, S. and Everson, R., Eds., *Independent Component Analysis: Principles and Practice*, Cambrige University Press (2001).
28. Hyvärinen, A., Karhunen, J. and Oja, E., *Independent Component Analysis*, John Wiley & Sons (2001).
29. Cover, T.M. and Thomas, J.A., *Elements of Information Theory*, Wiley Series in Telecommunications, (1991).
30. Babaie-Zadeh, M., Jutten, C. and Nayebi, K. "Differential of mutual information", *IEEE Signal Processing Letters*, in press (2001).
31. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill (1991).
32. Deboor, C., *A Practical Guide to Splines*, Springer-Verlag (1978).
33. Reinsch, C.H. "Smoothing by spline functions", *Numer. Math.*, **10**, pp 177–183 (1967).
34. Eubank, R.L., *Spline Smoothing and Nonparanetric Regression*, Dekker (1988).
35. Hardle, W., *Smoothing Techniques with Implementation in S*, Springer-Verlag (1991).
36. Taleb, A., *Séparation de Sources dans des Mélanges Post Non-Linéaires*, Thèse de l'INP de Grenoble, in French (1999).
37. Babaie-Zadeh, M., Jutten, M.C. and Nayebi, K. "Separating convolutive mixtures by mutual information minimization", *Proceedings of IWANN'2001*, Granada, Spain, pp 834–842 (June 2001).