BAYESIAN SEPARATION OF DISCRETE SOURCES VIA GIBBS SAMPLING

Stéphane Senecal and Pierre-Olivier Amblard

LIS UPRESA CNRS 5083
BP 46, 38402 Saint Martin d'Hères Cedex, France
Ph. : 33 4 76 82 71 07, Fax : 33 4 76 82 63 84
e-mail: Stephane.Senecal@inpg.fr, Bidou.Amblard@inpg.fr

ABSTRACT

Source separation, one of the most recent domain of signal processing, consists in recovering signals mixed by an unknown transmission channel. Many approaches manage the separation using different techniques such as likelihood and information theory ([1],[6]) or high order statistics ([7]). This paper proposes a Bayesian approach to the problem of an instantaneous linear mixing, considering the sources are discrete. The powerful Gibbs sampling algorithm enables to recover binary sources, but also the mixing coefficients and noise levels with efficiency.

1. INTRODUCTION

The following source separation problem is considered. The observations are represented by m independent signals $(x_1, ..., x_m)$ where each sample $x_i(t)$ of the signal x_i is an instantaneous linear mixing of n samples of source signals $(s_1, ..., s_n)$ corrupted by an additive Gaussian noise :

$$x_i(t) = \sum_{j=1}^{n} a_{i,j} s_j(t) + b_i(t) \tag{1}$$

for $t = 1, \ldots, T$ and this can be vectorially rewritten as

$$\underline{x}(t) = \underline{\underline{A}} \times \underline{s}(t) + \underline{b}(t) \tag{2}$$

$$\underline{b}(t) \sim N(\underline{0}, \underline{\underline{\sigma_b^2}}) \tag{3}$$

where $\sim$ denotes the relation of a random variable and its probability law. $N(\underline{m}, \underline{\underline{\Sigma}})$ denotes the Gaussian law with mean $\underline{m}$ and covariance matrix $\underline{\underline{\Sigma}}$.
We particularly focus on telecommunications models which constrain source densities to be discrete. We restrict here to the case of 2-PSK signals. Hence, distributions of $s_j(t)$ are set to $\frac{1}{2}\delta_{+1} + \frac{1}{2}\delta_{-1}$. This hypothesis is very implicative as it reduces general fields of incertitudes of source separation. Actually, source signals are always recovered up to a permutation and a scaling matrix effects. Indeed,

$$\underline{\underline{A}} \times \underline{s}(t) = \underline{\underline{A}} \times \underline{\underline{P}} \times \underline{\underline{\Lambda}} \times \underline{\underline{\Lambda}}^{-1} \times \underline{\underline{P}}^t \times \underline{s}(t) \tag{4}$$

holds where $\underline{\underline{P}}$ is a permutation matrix and $\underline{\underline{\Lambda}}$ a diagonal power scaling matrix. Here, the discrete density of sources constraints the latter matrix to have only elements equal to $+1$ or -1 as the power of each source signal is set to 1.

2. BAYESIAN INFERENCE

Bayesian approach consists in recovering source signals $\underline{s}(1 \to T)$ using the *posterior* density

$$p(\underline{s}(1 \to T)|\underline{x}(1 \to T)) \tag{5}$$

This density conjointly takes in account *prior* information of source signals available in $p(\underline{s}(1 \to T))$ and likelihood thanks to the fundamental Bayes formula

$$p(\underline{s}|\underline{x}) \propto p(\underline{x}|\underline{s}) \times p(\underline{s})$$

The knowledge of (5) gives us all estimators for $\underline{s}(t)$ such as modes, mean or other characteristics of the *posterior* density. Computing density (5) often requires an integration through the parameters of the model, called $\underline{\theta}$ for instance:

$$p(\underline{s}|\underline{x}) = \int p(\underline{s}, \underline{\theta}|\underline{x}) d\underline{\theta}$$

$$\propto \int p(\underline{x}|\underline{s}, \underline{\theta}) p(\underline{\theta}) d\underline{\theta}$$

In general case, these integrals cannot be calculated analytically and require estimation through Monte Carlo methods for instance.
The general Bayesian approaches to the source separation problem presented in [4] and [5] takes in account noise in the model and perform the maximisation of *posterior* probability densities. Unfortunately, methods used to compute the maximization only guarantee the theoretical convergence to a local *maximum* of the *posterior* density. If this density is unimodal, there is no problem, but taking *priors* of the form

$$p(\underline{\underline{A}}) \propto \exp(-\frac{1}{2\sigma_A^2} \|\underline{\underline{I}} - \underline{\underline{A}}^t \times \underline{\underline{A}}\|^2) \tag{6}$$

constraining the mixing matrix $\underline{\underline{A}}$ to be orthogonal (cf [4] and [5]), may imply a *posterior* density (5) extremely complex with a lot of local *maxima*.
Thus, it appears important to use methods performing a global maximization of *posterior* densities or a global optimisation of an other cost function. This can be achieved thanks to Monte-Carlo Markov Chains (MCMC) methods such as "simulated annealling" described in [12]. Moreover, being able to get the full *posterior* density and not only to estimate its modes, allows to use all information present in observed signals and thus to make any estimation.

In fact, general MCMC methods (Metropolis Hastings algorithms and Gibbs sampling for instance (see [8], [9], [10], [11]) enable to sample from any general distribution and then to draw sources and the model (1) parameters from the *posterior* density

$$p(\underline{s}(1 \to T), \underline{\underline{\sigma_b^2}}, \underline{\underline{A}}|\underline{x}(1 \to T)) \quad (7)$$

Thus, Gibbs sampling will help to estimate source signals $\underline{s}(t)$ but also mixing coefficients and even noise levels.
Such estimation can also be computed using the method described in [2] performing the maximization of the likelihood function through an Expectation-Maximization (EM) and Monte-Carlo methods. The model (2),(3) leads to a global optimisation of a cost function and to the estimation of $(\underline{\underline{A}}, \underline{\underline{\sigma_b^2}})$. The discrete characteristic of sources is not used explicitely in this algorithm but just to estimate the mixing matrix thanks to Monte-Carlo method. Another approach presented in [3], based on cardinals of source signal values' sets and Hankel matrix determinants, enables to estimate the mixing matrix up to the relation (4). This method is particularly adapted if the discrete values of sources are unknown.
These methods also take in account effects of additive Gaussian noise in the model (1) with unknown variance. The great advantage of Gibbs sampling is that it computes directly discrete estimated sources instead of applying a separating matrix on observed signals which is not convenient in case of important noise level.

3. THE GIBBS SAMPLER

The aim of Gibbs sampling is to draw a multidimensional random variable $(y_1, ...y_p)$ through a given probability density $p(y_1, ..., y_p)$. The main hypothesis its application requires is the possibility to easily simulate through marginal densities $p(y_k|y_1, ..., y_{k-1}, y_{k+1}, ..., y_p)$ or the knowledge of their analytic formulae.
The k-th step of the algorithm is the following :

- $y_1^k \sim p(y_1|y_2^{k-1}, ..., y_p^{k-1})$
- $y_2^k \sim p(y_2|y_1^k, y_3^{k-1} ..., y_p^{k-1})$
  ...
- $y_{p-1}^k \sim p(y_{p-1}|y_1^k, ..., y_{p-2}^k, y_p^{k-1})$
- $y_p^k \sim p(y_p|y_1^k, ..., y_{p-2}^k, y_{p-1}^k)$

The reader will find further information and full theoretical details on the Gibbs sampling algorithm in [10] and [11].

4. SOURCE SEPARATION ALGORITHM

The model (1) is considered for $1 \le i \le m, t = 1, \dots, T$. The sequences $\{s_j(t)\}_t$ and $\{b_i(t)\}_t$ are independent and identically distributed with the following densities :

$$s_j(t) \sim \frac{1}{2}\delta_{+1} + \frac{1}{2}\delta_{-1}$$
$$b_i(t) \sim N(0, \sigma_i^2) \quad (8)$$

We suppose n, the number of sources to be known.
The mixing coefficients $a_{i,j}$ are supposed to be unknown. The choice of a *prior* on these parameters is a difficult problem. For simplicity, we choose

$$a_{i,j} \sim N(a_{i,j}|0, \sigma_a^2) \quad (9)$$

where the hyperparameter σ_a measures the uncertainty on the amplitudes of mixing coefficients. The values of σ_i (cf (8)) and σ_a are unknown but they can be modelised thanks to conjugate *prior* distributions $IG(\alpha, \beta)$ where (α, β) are known constants. Such choice will be explained in §4.3.
The algorithm is simply the Gibbs sampler adapted to simulate the *posterior* density (7) and gives the following implementation for the $(k+1)$-th step for instance :
1: $\underline{s}^{k+1}(1 \to T) \sim p(\underline{s}(1 \to T)|\underline{x}(1 \to T), \underline{\underline{A}}^k, \underline{\sigma^2}(k), \sigma_a^2(k))$
2: $\underline{\underline{A}}^{k+1} \sim p(\underline{\underline{A}}^k|\underline{x}(1 \to T), \underline{s}^{k+1}(1 \to T), \underline{\sigma^2}(k), \sigma_a^2(k))$
3: $\sigma_a^2(k+1) \sim p(\sigma_a^2|\underline{x}(1 \to T), \underline{s}^{k+1}(1 \to T), \underline{\underline{A}}^{k+1}, \underline{\sigma^2}(k))$
4: $\underline{\sigma^2}(k+1) \sim p(\underline{\sigma^2}|\underline{x}(1 \to T), \underline{s}^{k+1}(1 \to T), \underline{\underline{A}}^{k+1}, \sigma_a^2(k+1))$

4.1. Simulation of source signals

Through the $(k+1)$-th step of Gibbs sampling, $s_j^{k+1}(t)$ is drawned from

$$p(s_j|\underline{x}(t), s_{l<j}^{k+1}(t), s_{l>j}^k(t), \underline{\underline{A}}^k, \underline{\sigma^2}(k), \sigma_a^2(k))$$
$$\propto p(\underline{x}(t)|s_j, s_{l<j}^{k+1}(t), s_{l>j}^k(t), \underline{\underline{A}}^k, \underline{\sigma^2}(k)) \times p(s_j)$$

Since the $x_i(t)$-s are independent conditionaly to $\underline{s}(t), \underline{\underline{A}}, \underline{\sigma^2}$, we get

$$p(\underline{x}(t)|s_j, s_{l<j}^{k+1}(t), s_{l>j}^k(t), \underline{\underline{A}}^k, \underline{\sigma^2}(k))$$
$$= \prod_{i=1}^n p(x_i(t)|s_j, s_{l<j}^{k+1}(t), s_{l>j}^k(t), \underline{a}_{i,.}^k, \sigma_i^2(k))$$
$$\propto \exp(-\sum_{i=1}^n \frac{1}{2\sigma_i^2(k)} \times$$
$$(x_i(t) - \sum_{l<j} a_{i,l}^k s_l^{k+1}(t) - a_{i,j}^k s_j(t) - \sum_{l>j} a_{i,l}^k s_l^k(t))^2)$$

Discrete densities of sources gives a distribution for $s_j^{k+1}(t)$ easily drawnable with an acceptance-rejection method :

- Computation of $p(s_j^{k+1}(t) = +1) \propto$

$$\exp(-\sum_{i=1}^n \frac{1}{2\sigma_i^2(k)}(x_i(t) - \sum_{l<j} a_{i,l}^k s_l^{k+1}(t) - a_{i,j} - \sum_{l>j} a_{i,l}^k s_l^k(t))^2)$$

- Computation of $p(s_j^{k+1}(t) = -1) \propto$

$$\exp(-\sum_{i=1}^n \frac{1}{2\sigma_i^2(k)}(x_i(t) - \sum_{l<j} a_{i,l}^k s_l^{k+1}(t) + a_{i,j} - \sum_{l>j} a_{i,l}^k s_l^k(t))^2)$$

- Generate $u \sim U_{[0,1]}$
- If
$$u \le \frac{p(s_j^{k+1}(t) = +1)}{p(s_j^{k+1}(t) = +1) + p(s_j^{k+1}(t) = -1)}$$
then $s_j^{k+1}(t) = +1$, otherwise $s_j^{k+1}(t) = -1$.

The instantaneous characteristic of the linear mixing model (2) enables to simulate all samples $s_j^{k+1}(t)$ for $1 \le t \le T$ in parallel.

4.2. Simulation of the mixing coefficients

Gibbs sampling enables also to estimate the coefficients of the instantaneous mixing $a_{i,j}$. Thus in the $(k+1)$-th step of the algorithm, $a_{i,j}^{k+1}$ is drawned from

$$p(a_{i,j}|\underline{x}(1\to T),\underline{s}^{k+1}(1\to T),a^k_{l\neq i,p\neq j},\underline{\sigma^2}(k),\sigma_a^2(k))$$

$$\propto p(x_i(1\to T)|\underline{s}^{k+1}(1\to T),a^{k+1}_{i,p<j},a_{i,j},a^k_{i,p>j},\sigma_i^2(k))\times$$

$$p(a_{i,j}|\sigma_a^2(k))$$

Since the $x_i(t)$-s are independent conditionaly to $\underline{s},\underline{\underline{A}},\underline{\sigma^2}$, we get

$$p(x_i(1\to T)|\underline{s}^{k+1}(1\to T),a^{k+1}_{i,p<j},a_{i,j},a^k_{i,p>j},\sigma_i^2(k))$$

$$=\prod_{t=1}^{T}p(x_i(t)|\underline{s}^{k+1}(t),a^{k+1}_{i,p<j},a_{i,j},a^k_{i,p>j},\sigma_i^2(k))$$

$$\propto\prod_{t=1}^{T}\exp(-\frac{1}{2\sigma_i^2(k)}\times$$

$$(x_i(t)-\sum_{p<j}a^{k+1}_{i,p}s_p(t)^{k+1}-a_{i,j}s_j(t)^{k+1}-\sum_{p>j}a^k_{i,p}s_p(t)^{k+1})^2)$$

$$\propto\exp(-\frac{1}{2\sigma_1^2(k)}\times$$

$$\sum_{t=1}^{T}(x_i(t)-\sum_{p<j}a^{k+1}_{i,p}s_p(t)^{k+1}-a_{i,j}s_j(t)^{k+1}-\sum_{p>j}a^k_{i,p}s_p(t)^{k+1})^2)$$

Thus we obtain a Gaussian likelihood density distribution for $a_{i,j}$ $N(\mu_{a_{i,j}},\sigma^2_{a_{i,j}})$ where

$$\sigma_{a_{i,j}}=\frac{\sigma_i(k)}{\sqrt{\sum_{t=1}^{T}s_j^{k+1}(t)^2}}$$

$$\mu_{a_{i,j}}=\frac{1}{\sum_{t=1}^{T}s_j^{k+1}(t)^2}\times(\sum_{t=1}^{T}s_j^{k+1}(t)\times$$

$$(x_i(t)-\sum_{p<j}a^{k+1}_{i,p}s_p^{k+1}(t)-\sum_{p>j}a^k_{i,p}s_p^{k+1}(t)))$$

Thus

$$a_{i,j}^{k+1}\sim N(\mu_{a_{i,j}},\sigma^2_{a_{i,j}})\times N(0,\sigma_a^2(k))$$

$$a_{i,j}^{k+1}\sim N(\frac{\mu_{a_{i,j}}}{1+\frac{\sigma^2_{a_{i,j}}}{\sigma_a^2(k)}},\frac{1}{\frac{1}{\sigma^2_{a_{i,j}}}+\frac{1}{\sigma_a^2(k)}})\qquad(10)$$

If we focus on the identification of the mixing channel (i.e. the coefficients $a_{i,j}$ of the mixing matrix), the Gaussian characteristic of *posterior* $a_{i,j}$ distributions (10) implies that their means are also their unique -and so theoretically global-modes.

In practice the "burn-in" phenomenon of the simulation algorithm makes appearing small local *maxima* of the *posterior* density which are more and more neglectible as the number of iterations of the Gibbs sampling is increasing.

4.3. The choose of conjugate *priors*

In a case of a lack of information on noise levels σ_i^2 and the hyperparameter σ_a^2, *prior* information can be given. It can be settled for instance that $\sigma_i^2\sim IG(\alpha_i,\beta_i)$ and $\sigma_a^2\sim IG(\alpha_a,\beta_a)$ where IG represents the Inverse Gamma law whose density is

$$f(x|\alpha,\beta)=\frac{\beta^\alpha}{\Gamma(\alpha)}x^{-(\alpha+1)}\exp(-\frac{\beta}{x})$$

for $x>0$ $(\alpha>0,\beta>0)$.

The law $IG(\alpha,\beta)$ admits a mean $E[x]=\frac{\beta}{\alpha-1}$ for $\alpha>1$ and a variance $V[x]=\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$ for $\alpha>2$. Its density has a unique mode at $\frac{\beta}{\alpha+1}$.

The Inverse Gamma law is used in order to take a conjugate *prior*. This choice implies that the *posterior* densities has also the same Inverse Gamma form and thus is easily simulable. Moreover, this law tends to be non-informative (i.e. having a flat or uniform probability density) if $(\alpha,\beta)\to(0,0)$. Full explanations about conjugate *priors* and algorithms of IG sampling can be found in [8].

4.3.1. Simulation of σ_a^2

Thus, during the $(k+1)$-th step of the algorithm, $\sigma_a^2(k+1)$ is drawned from

$$p(\sigma_a^2|\underline{x}(1\to T),\underline{s}^{k+1}(1\to T),\underline{\underline{A}}^{k+1},\underline{\sigma^2}(k))$$

$$\propto p(\underline{\underline{A}}^{k+1}|\sigma_a^2)\times p(\sigma_a^2)$$

$$\propto N(0,\sigma_a^2)^{m\times n}\times IG(\sigma_a^2|\alpha_a,\beta_a)$$

$$\propto(\frac{1}{2\pi\sigma_a})^{\frac{m\times n}{2}}\exp(-\frac{1}{2\sigma_a^2}\sum_{i,j}a^2_{i,j}(k+1))\times$$

$$(\sigma_a^2)^{-(\alpha_a+1)}\exp(-\frac{\beta_a}{\sigma_a^2})$$

$$\propto(\sigma_a^2)^{-(\alpha_a+\frac{m\times n}{2}+1)}\times$$

$$\exp(-\frac{1}{\sigma_a^2}(\frac{1}{2}\sum_{i,j}a^2_{i,j}(k+1)+\beta_a))$$

Finally,

$$\sigma_a^2(k+1)\sim IG(\alpha_a+\frac{n\times m}{2},\frac{1}{2}\sum_{i,j}a^2_{i,j}(k+1)+\beta_a)$$

4.3.2. Simulation of σ_i^2

During the $(k+1)$-th step of the algorithm, $\sigma_i^2(k+1)$ is drawned from

$$p(\sigma_i^2|\underline{x}(1\to T),\underline{s}^{k+1}(1\to T),\underline{\underline{A}}^{k+1},\sigma^2_{l\neq i}(k))$$

$$=p(\sigma_i^2|x_i(1\to T),\underline{s}^{k+1}(1\to T),\underline{a}_{i,.}^{k+1})$$

$$\propto p(x_i(1\to T)|\underline{s}^{k+1}(1\to T),\underline{a}_{i,.}^{k+1},\sigma_i^2)\times p(\sigma_i^2)$$

And, since $\{x_i(t)\}$ are independent conditionnaly to $\underline{s},\underline{\underline{A}},\underline{\sigma^2}$

$$p(x_i(1\to T)|\underline{s}^{k+1}(1\to T),\underline{a}_{i,.}^{k+1},\sigma_i^2)$$

$$= \prod_{t=1}^{T} p(x_i(t)|\underline{s}^{k+1}(t), \underline{a}_{i,.}^{k+1}, \sigma_i^2)$$

$$= \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{1}{2\sigma_i^2}(x_i(t) - \sum_{j=1}^{n} a_{i,j}^{k+1} s_j(t)^{k+1})^2)$$

$$\propto (\sigma_i^2)^{\frac{T}{2}} \exp(-\frac{1}{2\sigma_i^2} \sum_{t=1}^{T}(x_i(t) - \sum_{j=1}^{n} a_{i,j}^{k+1} s_j(t)^{k+1})^2)$$

Thus,

$$p(\sigma_i^2|x_i(1 \to T), \underline{s}^{k+1}(1 \to T), \underline{a}_{i,.}^{k+1}) \propto$$

$$IG(\frac{T}{2}-1, \frac{1}{2}\sum_{t=1}^{T}(x_i(t) - \sum_{j=1}^{n} a_{i,j}^{k+1} s_j(t)^{k+1})^2) \times IG(\sigma_i^2|\alpha_i, \beta_i)$$

Finally,

$$\sigma_i^2(k+1) \sim IG(\alpha_i + \frac{T}{2}, \frac{1}{2}\sum_{t=1}^{T}(x_i(t) - \sum_{j=1}^{n} a_{i,j}^{k+1} s_j(t)^{k+1})^2 + \beta_i)$$

4.4. Implementation

To sum up the algorithm, the $(k+1)$-th step of the Gibbs sampler goes like this:

- Draw $\underline{s_j}^{k+1}$ for $j = 1, \ldots, n$ from the acceptation-reject method described in §4.1 and the knowledge of
$$(\underline{x}, \underline{s}_{1,\ldots,j-1}^{k+1}, \underline{s}_{j+1,\ldots,n}^{k}, \underline{\underline{A}}^{k}, \underline{\sigma}^2(k))$$
- Draw $\underline{\underline{A}}^{k+1}$. According to (10), the simulation can be computed column by column and all the mixing coefficients are drawned from a Gaussian law.
-
$$\sigma_a^2(k+1) \sim IG(\alpha_a + \frac{n \times m}{2}, \frac{1}{2}\sum_{i,j} a_{i,j}^2(k+1) + \beta_a)$$
- Draw $\underline{\sigma}^2(k+1)$ for $i = 1, \ldots, m$ from
$$IG(\alpha_i + \frac{T}{2}, \frac{1}{2}\sum_{t=1}^{T}(x_i(t) - \sum_{j=1}^{n} a_{i,j}^{k+1} s_j(t)^{k+1})^2 + \beta_i)$$

5. NUMERICAL EXAMPLE

5.1. Model studied

We have generated synthetic data according to the model :

$$x_1(t) = 4s_1(t) - 3s_2(t) + \sigma_1 b_1(t)$$

$$x_2(t) = s_1(t) + 5s_2(t) + \sigma_2 b_2(t)$$

for $t = 1, .., 1024$ with

$$b_1(t), b_2(t) \sim N(0, 1)$$

and $[\sigma_1; \sigma_2] = [1.58; 3.60]$ providing Signal to Noise Ratios $SNR_1 = \frac{1}{10}$ and $SNR_2 = \frac{1}{2}$ (cf Figure (2)).
Plotting histograms from the distribution of x_1 and x_2 would help us to get a first estimation of respectively σ_1 and σ_2,

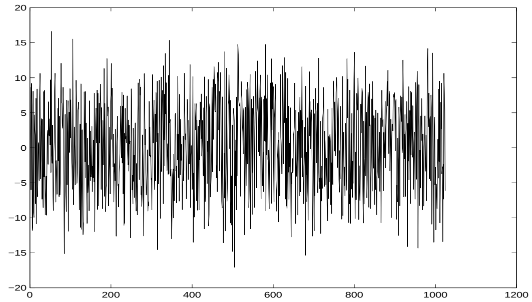

Figure 1: Second observation signal (1024 samples,σ_2 = 3.60)

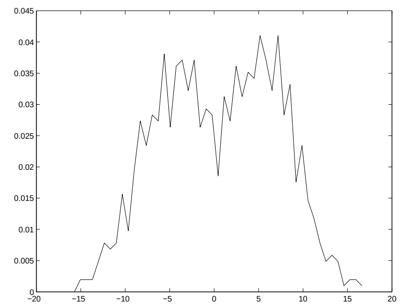

Figure 2: Histogram of the second observation signal, the high noise-level here will make source separation difficult

but we assume here that the noise levels are totally unknown, as the hyperparemeter σ_a.
Thus, we chose for them non-informative *priors* of the form $IG(\alpha, \beta)$ with $(\alpha, \beta) = (0.1, 0.1)$.
Then, 500 iterations of the Gibbs sampling are runned to compute the distribution

$$p(s_1(1 \to 1024), s_2(1 \to 1024), a_{1,1}, a_{1,2},$$

$$a_{2,1}, a_{2,2}, \sigma_a^2, \sigma_1^2, \sigma_2^2|x_1(1 \to 1024), x_2(1 \to 1024))$$

5.2. Numerical results

Sources $(\hat{s}_1(t), \hat{s}_2(t))$ estimated are equivalent to real signals $(-s_2(t), -s_1(t))$ with precision 3% and 3.5% respectively estimated by Bit Error Rates (BER). These ratios represents the number of wrong estimated binary samples $\{\hat{s_j}(t)\}_{1 \leq t \leq T}$ on the length of the signal $\hat{s_j}$.
For comparison, if the exact mixing matrix $\underline{\underline{A}}$ is known, we can compute (11) as an estimator of $\underline{s}(t)$. Thus, sources are recovered with a BER precision around 5%,6%, showing the necessity to include the observation noise in separation methods for this model and to compute directly estimated sources as the estimation of a separating matrix is not sufficient in this case.

$$\underline{z}(t) = (\underline{\underline{A}^{-1}} \times \underline{x}(t)) \quad (11)$$

For instance, if the EM-likelihood method (initialized with the JADE algorithm, see [2]) is applied to the data, sources are recovered with precision rates around 5% thanks to the relation (11) where the mixing matrix has been estimated by (12) after a dozen iterations.

$$\begin{pmatrix} 2.9360 & 3.7915 \\ -4.9432 & 0.9228 \end{pmatrix} \qquad (12)$$

The EM-likelihood method enables to estimate noise levels $[\hat{\sigma_1}; \hat{\sigma_2}] = [1.7921; 3.7435]$.
The Gibbs sampling algorithm gives also good estimations of noise levels σ_1 and σ_2 (see Figures (3) and (6) for instance). The mean of simulated σ_1 computed from the 100th iteration to the 500th iteration is equal to 1.51 (real value=1.58). Its distribution shows a sharp mode at 1.53 and other local modes due to the burn-in of the sampler are discarded as the number of iterations increases. The sim-

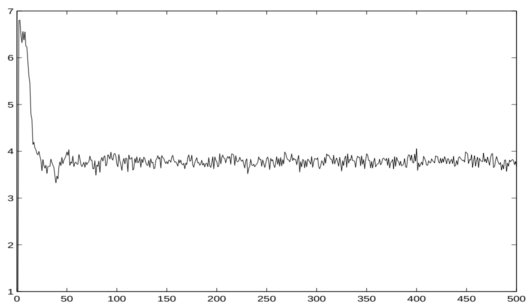

Figure 3: Estimation of $\sigma_2 = 3.60$, the mean computed from the 100th iteration to the 500th iteration is equal to 3.78

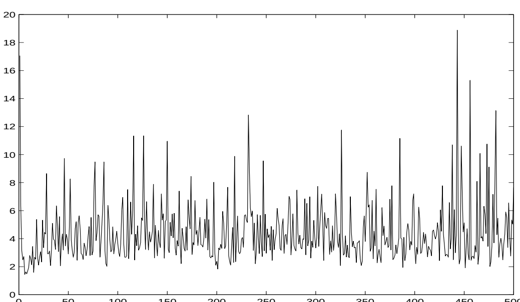

Figure 4: Estimation of σ_a, the mean computed from the 100th iteration to the 500th iteration is equal to 4.49

ulation of the hyperparameter σ_a reveals a good behaviour of the algorithm for fitting the data. Even if the estimated σ_a can variate far from its mean as showed in Figure (4), its distribution (cf Figure (7)) gives a right estimate of the amplitude variations of mixing coefficients. This value can be set around 4 or 5 for the considered model. Mixing coefficients are also well estimated (cf Figures (5) and (8)). The algorithm converges fastly to the mixing matrix up to a permutation and a diagonal power-scaling matrix. The mean of the estimated mixing matrix computed from the 100th iteration to the 500th iteration is equal to

$$\begin{pmatrix} 3.0098 & -3.9844 \\ -4.9897 & -1.0429 \end{pmatrix}$$

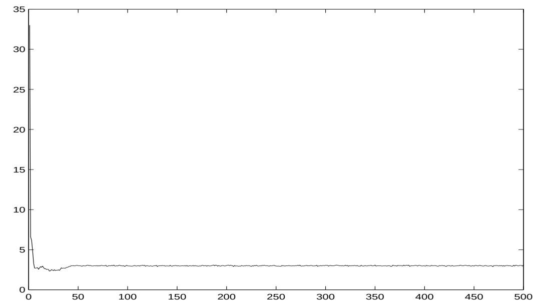

Figure 5: Estimation of $a_{1,1} = 3$, the mean computed from the 100th iteration to the 500th iteration is equal to 3.0098

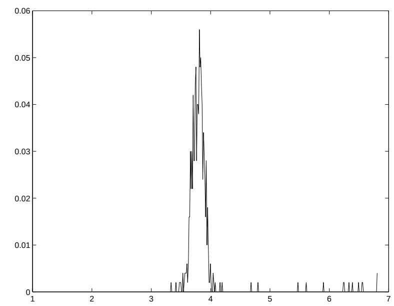

Figure 6: Histogram of the estimated σ_2. There is a mode at 3.81 (real value=3.60)

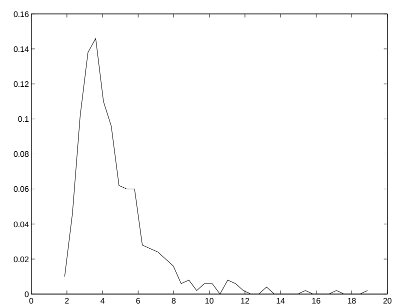

Figure 7: Histogram of the estimated σ_a. There is a mode at 3.61

The matrix which maximizes the *posterior* density

$$p(\underline{A}|\underline{x_1}, \underline{x_2}, \underline{s_1}, \underline{s_2}, \sigma_a^2, \sigma_1^2, \sigma_2^2)$$

is estimated by

$$\begin{pmatrix} 3.0177 & -3.9479 \\ -4.9929 & -1.0982 \end{pmatrix}$$

These matrix are very close to the exact mixing matrix

$$\begin{pmatrix} 4 & -3 \\ 1 & 5 \end{pmatrix}$$

For comparison, if a general source separation algorithm such as FastICA (see [6]) is applied to the data with a deflation approach and a tanh-type of non-linearity, sources are

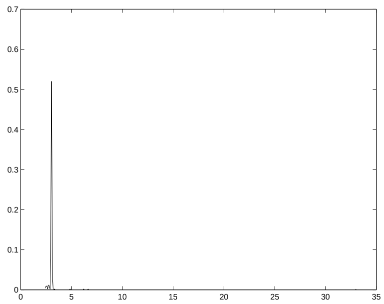

Figure 8: Histogram of the estimated $a_{1,1}$. There is a sharp mode at 3.0177 (real value=3)

recovered with a BER precision of 2% for the first one and 5% for the second one. This gap between precision rates comes from the model considered in §5.1 where the second mixing is highly corrupted by noise.
Thus, the deflation approach uses all the information to recover a first source and had difficulties to recover the second one as it appears with the estimated mixing matrix

$$\begin{pmatrix} 3.2859 & 3.9567 \\ -6.0527 & 2.4062 \end{pmatrix}$$

This method identify the first mixing and then a first source with accuracy but not the second one as the noise level is too high ($=\frac{1}{2}$).
The symmetric approach and other type of non-linearity give equivalent results which are quite good as the FastICA algorithm does not take in account observation noise in the model (1), but these methods cannot estimate the noise levels.
In the algorithm described in §4.4, the Gibbs sampler does not require more than one hundred iterations to reach its stationary density (7), thus having a computing cost time close to EM-likelihood maximization or FastICA.

6. CONCLUSION

The Gibbs sampling algorithm separates discrete sources and estimates the mixing coefficients with accuracy. Moreover, this method takes in account the discrete characteristics of sources and estimates noise levels of the model. Performance and convergence of the algorithm are under study. These characteristics depend on the choice of the *prior* on $\underline{\underline{A}}$. We chose a Gaussian law for the simplicity of its simulaation but *priors* constraining the matrix $\underline{\underline{A}}$ to be orthogonal or non-singular would involve more complicated calculations such as runs of Metropolis-Hastings algorithm in a step of the Gibbs sampler.
It would be interesting now to complete the method with reversible-jumps MCMC methods in order to treat the problem with an unknown number of sources by drawing the number of source signals with the *posterior* density. In case of general discrete source separation,probabilities of symbols could also be taken account in the *posterior* density and then be estimated through the Gibbs sampling.
This method could also be generalised to different kind of mixing model and even non-linear ones.

7. ACKNOWLEDGEMENT

The authors thank Pr. Christian Jutten for his helpful comments.

8. REFERENCES

[1] J.-F. Cardoso. Infomax and maximum likelihood for source separation". In *IEEE Letters on Signal Processing*, vol. 4, April 1997, pp. 112-114.

[2] A. Belouchrani and J.-F. Cardoso. Maximum likelihood source separation for discrete sources. In *Signal Processing VII*, P. Grant, W. Sandham, M. Holt and C. Cowan, Eds., 1994. JADE program available on *http://sig.enst.fr/ cardoso/stuff.html*

[3] Fabrice Gamboa and Elisabeth Gassiat. Source separation when the Input sources are discrete or have constant modulus. In *IEEE Transactions on Signal Processing*, vol.45, n12, December 1997, pp. 3062-3072.

[4] K. Knuth. A Bayesian approach to source separation. In *proceedings of the First International workshop on Independdent Component Analysis and Signal Separation : ICA'99*, Aussois, France (C. Jutten, J.-F. Cardoso and P. Loubaton, eds.), pp. 283-288, 1999.

[5] Ali Mohammad-Djafari. A Bayesian approach to source separation. *MaxEnt 1999*, Boise, Idaho, USA.

[6] Aapo Hyvärinen and Erkki Oja. *"Independent Component Analysis : a Tutorial"*. Laboratory of Computer and Information Science, Helsinki University of Technology. Avalaible on *http://www.cis.hut.fi/projects/ica/*

[7] J.L. Lacoume, P.O. Amblard and P. Comon. *"Statistiques d'ordre supérieur pour le traitement du signal"*. Masson, 1997 (in French).

[8] Christian Robert. *"L'analyse statistique Bayésienne"*. Statistique mathématique et probabilité, Economica, 1992.
English version : *"The Bayesian Choice- A Decision Theoretic Motivation"*. Springer Verlag, 1996.

[9] José M. Bernardo and Adrian F.M. Smith. *"Bayesian Theory"*. John Wiley and Sons, 1994.

[10] Christian Robert. *"Méthodes Monte-Carlo par Chaînes de Markov"*. Statistique mathématique et probabilité, Economica, 1996 (in French).

[11] Radford M. Neal. *"Probabilistic Inference Using Markov Chain Monte Carlo Methods"*. Technical Report CRG-TR-93-1, Dpt. of Computer Science, University of Toronto.

[12] P. J. M. van Laarhoven and E. H. L. Aarts. *"Simulated Annealling : Theory and Applications"*. Mathematics and Its Applications, Kluwer Academic Publishers, 1988.