An Iterative Bayesian Algorithm for Sparse Component Analysis (SCA) in Presence of Noise

H. Zayyani[1]* *Student Member*, M. Babaie-Zadeh[1] *Member* and C. Jutten[2] *Fellow*

***Abstract*—We present a Bayesian approach for Sparse Component Analysis (SCA) in the noisy case. The algorithm is essentially a method for obtaining sufficiently sparse solutions of under-determined systems of linear equations with additive Gaussian noise. In general, an under-determined system of linear equations has infinitely many solutions. However, it has been shown that sufficiently sparse solutions can be uniquely identified. Our main objective is to find this unique solution. Our method is based on a novel estimation of source parameters and Maximum A Posteriori (MAP) estimation of sources. To tackle the great complexity of the MAP algorithm (when the number of sources and mixtures become large), we propose an Iterative Bayesian Algorithm (IBA). This IBA algorithm is based on the MAP estimation of sources, too, but optimized with a steepest-ascent method. The convergence analysis of the IBA algorithm and its convergence to true global maximum are also proved. Simulation results show that the performance achieved by the IBA algorithm is among the best, while its complexity is rather high in comparison to other algorithms. Simulation results also show the low sensitivity of the IBA algorithm to its simulation parameters.**

***Index Terms*-Sparse component analysis, sparse decomposition, atomic decomposition, sparse source separation, blind source separation.**

I. Introduction

Finding (sufficiently) sparse solutions of under-determined systems of linear equations (possibly in the noisy case) has been studied extensively in recent years [1], [2], [3], [4], [5], [6], [7]. The problem has a growing range of applications in signal processing. To introduce the problem in more details, we will use the context of Sparse Component Analysis (SCA) [8]. The discussions, however, may be easily followed in other contexts of application, for example, in finding a 'sparse decomposition' of a signal on an overcomplete dictionary, which is the goal of the so-called overcomplete 'atomic decomposition' [9]. Sparse representations are well suited for content analysis, i.e., extracting structure or meaning of a signal [10]. They may also be applied to signal compression applications to facilitate the storage, processing and communication of signals [11]. Another application of sparse decomposition is related to its denoising capability [12], [13]. Recently, interesting applications in decoding of real error correcting codes have been reported [14], [15], [16]. Also, some applications in the sampling theory have been initiated which can be regarded as unifying the sampling and the coding theories [17], [18].

SCA can be viewed as a method to achieve separation of sparse sources [3], [6], [7], [8], [19], [20]. The Blind Source Separation (BSS) problem is to recover m unknown sources from n observed mixtures of them, where little or no information is available about the sources (except their statistical independence) and about the mixing system. In this paper, we consider the noisy linear instantaneous model at each instant:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{x}$, $\mathbf{s}$ and $\mathbf{n}$ are $n \times 1$, $m \times 1$ and $n \times 1$ vectors of sources, mixtures and white Gaussian noises, respectively, and $\mathbf{A}$ is the $n \times m$ mixing matrix. In the under-determined case ($m > n$), estimating the mixing matrix is not sufficient to recover the sources, since the mixing matrix is not invertible. Then it appears that the estimation of sources requires other prior information on the sources. One prior information that can result in source recovery in under-determined case is the sparsity of sources. The sparsity of a source vector means that almost all its entries are zero (or near zero) and only a few entries are nonzero. If we restrict ourselves to sufficiently sparse solutions of the under-determined system of linear equations, it has been shown that the solution is unique [1], [2], [21].

SCA can be solved in two steps: first estimating the mixing matrix, and then estimating the sources. The first step may be accomplished by means of clustering [3], [22] or other methods [6], [7]. The second step requires finding the sparse solution of (1) assuming $\mathbf{A}$ to be known [23], [24], [25], [26], [27]. Finally, some methods estimate the mixing matrix and sources simultaneously [5], [20]. In this paper, we focus on the source estimation, assuming $\mathbf{A}$ is already known.

Atomic decomposition [9] is another viewpoint to the same mathematical problem as above. In this viewpoint, we have just 'one' signal whose samples are collected in the $m \times 1$ signal vector $\mathbf{x}$ and the objective is to express it as a linear combination of a set of predetermined signals where their samples are collected in vectors $\{\mathbf{a}_i\}_{i=1}^{m}$. After [28], the $\mathbf{a}_i$'s are called atoms and they collectively form a dictionary over which the signal is to be decomposed. In this paper, we also

[1]Electrical engineering department and Advanced Communication Research Institute (ACRI), Sharif university of technology, Tehran, Iran.

[2]GIPSA-lab, Grenoble, and Institut Universitaire de France, France.

* This work has been partially funded by Iran NSF (INSF) under contract number 86/994, by Iran Telecom Research Center (ITRC), and also by center for International Research and Collaboration (ISMO) and French embassy in Tehran in the framework of a GundiShapour collaboration program.

First author: Hadi Zayyani, email: zayyani@ee.sharif.edu, Tel: +98 21 44670481, Fax: +98 21 66023261.

Second author: Massoud Babaie-zadeh, email: mbzadeh@sharif.edu, Tel: +98 21 66165925, Fax: +98 21 66023261.

Third author: Christian Jutten, email: christian.jutten@gipsa-lab.grenoble-inp.fr, Tel: +33 (0)4 76574351, Fax: +33 (0)4 76574790.

consider an additive noise term in the decomposition. So we can write $\mathbf{x} = \mathbf{A}\mathbf{s}+\mathbf{n}$, where $\mathbf{A}$ is the $n\times m$ dictionary (matrix) with the columns being the atoms and $\mathbf{s}$ is the $m \times 1$ vector of coefficients. The vector $\mathbf{n}$ can be interpreted either as the noisy term of the original signal that we intend to decompose (the term that is not in the span of our dictionary) or as the allowed error for the decomposition process. A dictionary with $m > n$ atoms is called *overcomplete*. It is clear that the representation of a signal on an overcomplete dictionary is not unique. However, as above, if a signal has a sufficiently sparse representation over such a dictionary, it would be unique under some mild conditions on the dictionary [4].

To obtain the sparse solution of (1), an approach is to search for solutions having minimum ℓ^0-norm, i.e. minimum number of nonzero components. This method is computationally intractable when the dimension increases (due to the need for a combinatorial search), and it is too sensitive to noise (because any small amount of noise completely change the ℓ^0-norm of a vector). Then, one of the most successful approaches of finding the sparse solution is Basis Pursuit (BP) [9], which achieves a convexification of the problem by replacing the ℓ^0-norm with an ℓ^1-norm. In other words, BP proposes to find the solution of (1) for which $\sum_{i=1}^{m} |s_i|$ is minimized. The minimum ℓ^1-norm solution is also the MAP source estimation under Laplacian source prior for the noiseless case [29]. As a main benefit, it can be easily implemented by Linear Programming (LP) methods (especially fast interior-point LP solvers). Recently, a fast method called smoothed-ℓ^0 method was proposed to minimize a smoothed approximation of the ℓ^0-norm [25], [26]. The FOCal Underdetermined System Solver (FOCUSS) uses ℓ^p-norm as a replacement for the ℓ^0-norm and a Lagrange multiplier method for the optimization [21]. Other simple approaches are Matching-Pursuit (MP) [28], [30] and Orthogonal Matching Pursuit (OMP) [31] algorithms. Another refined algorithm is the stage-wise Orthogonal Matching Pursuit (stOMP) [32]. Recently, a fast iterative detection-estimation algorithm was used to solve this problem [23]. Moreover, Bayesian approaches have been proposed: Sparse Bayesian Learning (SBL) [33] and recently, Bayesian Compressive Sensing (BCS) [34]. Finally, sparse reconstruction are possible using some new gradient methods like Gradient Pursuits [35] and Gradient Projection [36].

Among available methods for sparse decomposition, the fast methods (e.g. MP) usually don't produce accurate results, while BP which is guaranteed to asymptotically obtain the exact minimum ℓ^0-norm solution will become computationally demanding for large dimensions. With regard to the new applications of this area in decoding [14] or de-noising [12] in which the performance of the algorithm in presence of noise is very important, finding a new algorithm with a better accuracy is a necessity. In this paper, our main concern is in this way.

In our algorithm, we first proposed a 3-step (sub-)optimum (in the MAP sense) method for SCA in the noisy underdetermined case for Bernoulli-Gaussian (BG) sources. It has the drawback of large complexity which is not tractable when there are many sources and mixtures. To tackle the great complexity, an iterative method for MAP estimation of sources is presented. This leads to a more efficient algorithm, which is again a 3-step iterative Bayesian algorithm, among which two steps are expectation and maximization steps. These steps resemble the classical EM algorithm [37] which is also successfully used in [38].

BG sources is a usual model in the field of deconvolution, which is reviewed by Mendel in [39], and it was used especially in seismic deconvolution [39], [40], [41], [42], [43]. Using ℓ^1-norm to find sparse solutions of linear systems of equations can also be traced back to Taylor et al. [40] in this field. The relaxation of the discrete nature of BG processes by using a Gaussian mixture model has been considered in [43], based on a cost function with a weighting parameter. The authors used an EM algorithm for estimating their parameters, with a heuristic method for choosing the weighting parameter. In contrast, our work is originated from a MAP algorithm and the treatment is totally Bayesian and the cost function is dependent on the statistical parameters. We also used iterative Bayesian techniques to estimate all the unknowns in a rather unified way. So, the global algorithm resembles an EM algorithm, while we used some other Bayesian techniques for parameter estimation.

Our approach can be categorized as a Bayesian method for SCA [5], [20], [45] which is efficient for large number of sources and mixtures.

The paper starts with the statement of the system model in Section II. Then, a MAP solution is proposed in Section III. Based on this algorithm, we develop the new Iterative Bayesian Algorithm (IBA) in Section IV. Section V provides proof of convergence of the proposed IBA algorithm. Finally, we present some simulation results in Section VI.

II. System model

The noise vector in the model (1) is assumed zero-mean Gaussian with a covariance matrix $\sigma_n^2\mathbf{I}$. For modeling the sparse sources, we assumed the sources are inactive with probability p, and are active with probability $1-p$ (sparsity of sources implies that p is near 1). In the inactive case, the value of sources is zero and in the active case the value is obtained from a Gaussian distribution. This is called a Bernoulli-Gaussian model. The BG model is a realistic model in the sparse deconvolution applications and it has been extensively used in this literature [39], [40], [41], [42], [43]. This model has also been used in [16] for simultaneously modeling the impulse and background noise in a real-field channel coding system. It is also used to model the impulse noise in a communication channel [46], or the sparse vector in an application of regression [47]. This model is also suitable for sparse decomposition applications where we want to decompose a signal as a combination of only a few atoms of the dictionary, while the coefficients of the other atoms are zero. So, the probability density of the sources in SCA (or coefficients in sparse decomposition) is:

$$p(s_i) = p\delta(s_i) + (1-p)N(0, \sigma_r^2) \tag{2}$$

In this model, any sample of the sources can be written as $s_i = q_i r_i$ where q_i is a binary variable (with binomial

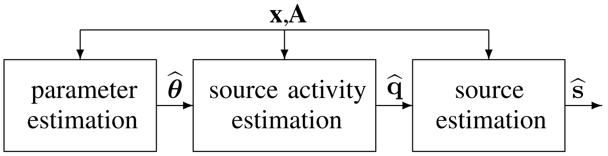


Fig. 1. The block diagram of our MAP algorithm

distribution) and r_i is the amplitude of the i'th source with Gaussian distribution. So the source vector can be written as:

$$\mathbf{s} = \mathbf{Q}\mathbf{r}, \qquad \mathbf{Q} = \mathrm{diag}(\mathbf{q}) \tag{3}$$

where $\mathbf{q} \triangleq (q_1, q_2, ..., q_m)^T$ and $\mathbf{r} \triangleq (r_1, r_2, ..., r_m)^T$ are the 'source activity vector' and 'amplitude vector', respectively. Each element of the source activity vector shows the activity of the corresponding source. That is:

$$q_i = \begin{cases} 1 & \text{if } s_i \text{ is active} \quad \text{(with probability } 1-p) \\ 0 & \text{if } s_i \text{ is inactive} \quad \text{(with probability } p) \end{cases} \tag{4}$$

Consequently, the probability $p(\mathbf{q})$ of source activity vector $\mathbf{q}$ is equal to:

$$p(\mathbf{q}) = (1-p)^{n_a}(p)^{m-n_a} \tag{5}$$

where n_a is the number of active sources i.e. the number of 1's in $\mathbf{q}$. Note that, in this paper, we use the same notation $p(.)$ for both the probability and for Probability Density Function (PDF).

III. MAP Algorithm

In [24], we proposed a three step MAP algorithm for the noisy sparse component analysis. Here, we explain it in details, because it is the basis of our IBA algorithm. Its block diagram is shown in Fig. 1, and consists of 3 steps. The first step is the estimation of the source parameters (p, σ_r) and noise parameter (σ_n). If we define the parameter vector as $\boldsymbol{\theta} = (p, \sigma_r, \sigma_n)^T$, the objective of this step is to obtain an estimation of $\boldsymbol{\theta}$. The second step is the estimation of the source activity vector $(\mathbf{q})$ which is defined in (4). The last step is the source estimation.

A. Parameter Estimation

The parameter estimation step is done by a novel method based on second and fourth order moments (for p and σ_r) and by a special application of EM algorithm [48](for σ_n). For estimating p and σ_r, let one of the mixtures, be $x = \sum_{i=1}^m a_i s_i + n = y + n$. Then, neglecting the noise contribution (i.e. assuming $\sigma_n \ll \sigma_r$), and taking into account that the sources are independent and zero-mean, then the relationship between the moments of the mixture and the moments of the sources are:

$$E(x^2) \approx E(y^2) = (\sum_i a_i^2)E(s_i^2) \tag{6}$$

$$E(x^4) \approx E(y^4) = (\sum_i a_i^4)E(s_i^4) + (6 \sum_{i,j,i\neq j} a_i^2 a_j^2)E^2(s_i^2) \tag{7}$$

In (6) and (7), we have assumed that the noise term is small and so the moments of the mixture without noise (y) are approximately the same as the moment of the true mixture (x). From the above equations and by assuming that all the sources have equal moments (due to identical parameters) and the a_i's are known, the source moments can be computed. Especially, concerning the second and fourth order moments of the sources, one can obtain the following relations:

$$m_2 = E(s_i^2) = (1-p)\sigma_r^2 \tag{8}$$

$$m_4 = E(s_i^4) = 3(1-p)\sigma_r^4 \tag{9}$$

From (8) and (9), we can compute the two source parameters:

$$\hat{\sigma}_r = \sqrt{\frac{\hat{m}_4}{3\hat{m}_2}} \tag{10}$$

$$\hat{p} = 1 - \frac{3\hat{m}_2^2}{\hat{m}_4} \tag{11}$$

where $\hat{m}_2$ and $\hat{m}_4$ are estimated by equations (6) and (7). Moreover, $E(x^2)$ and $E(x^4)$ are estimated from empirical expectation (weighted sum) of the corresponding mixture. To estimate the observation noise variance (σ_n), we use a special application of EM algorithm introduced in [48] which gives a maximum likelihood estimation of parameters of a Mixture of Gaussian (MoG) (centers, probabilities and variances) distribution from its samples. Since each s_i can be either active or inactive, a mixture $x = \sum_{i=1}^m a_i s_i + n$ can be modeled by a mixture of 2^m Gaussian components. Since p is close to one $(p \approx 1)$, one can neglect powers of $(1-p)$ greater than one, which leads to the following approximation of noisy sparse MoG:

$$p(x) \approx (p)^m N(0, \sigma_n^2) + (p)^{(m-1)}(1-p)\sum_{i=1}^m N(0, a_i^2\sigma_r^2 + \sigma_n^2) \tag{12}$$

This equation shows that each mixture has a MoG distribution and the smallest variance of these Gaussians is the variance of the noise (σ_n). Consequently, for estimating σ_n, the EM algorithm in [48] is applied to one of the mixture signals.

B. Estimation of the source activity vector

To estimate the source activity vector $(\mathbf{q})$, we use the MAP detector which maximizes the posterior probability $p(\mathbf{q}|\mathbf{x})$. Using the Bayes rule:

$$p(\mathbf{q}|\mathbf{x}) = \frac{p(\mathbf{q})p(\mathbf{x}|\mathbf{q})}{p(\mathbf{x})} \tag{13}$$

the MAP detector should maximize $p(\mathbf{q})p(\mathbf{x}|\mathbf{q})$. The prior source activity probability $(p(\mathbf{q}))$ is given by (5). The likelihood $(p(\mathbf{x}|\mathbf{q}))$ has a Gaussian distribution with the following conditional covariance:

$$\mathbf{Q}_q = E(\mathbf{x}\mathbf{x}^T|\mathbf{q}) = \mathbf{A}\mathbf{V}_q\mathbf{A}^T + \sigma_n^2\mathbf{I} \tag{14}$$

where the matrix $\mathbf{V}_q$ is the conditional covariance of the sources and can be stated as:

$$\mathbf{V}_q = \sigma_r^2\mathbf{Q} \tag{15}$$

where $\mathbf{Q}$ was defined in (3). Consequently, the prior probability $p(\mathbf{x}|\mathbf{q})$ has the following Gaussian form:

$$p(\mathbf{x}|\mathbf{q}) = \frac{1}{\sqrt{\det(2\pi\mathbf{Q}_q)}}\exp(\frac{-1}{2}\mathbf{x}^T\mathbf{Q}_q^{-1}\mathbf{x}) \tag{16}$$

and the MAP detector maximizes the $p(\mathbf{q})p(\mathbf{x}|\mathbf{q})$ over all 2^m cases for all source activity vectors:

$$\mathbf{q}_{\text{MAP}} = \underset{\mathbf{q}}{\text{argmax}}\,\frac{p(\mathbf{q})}{\sqrt{\det(2\pi\mathbf{Q}_q)}}\exp(\frac{-1}{2}\mathbf{x}^T\mathbf{Q}_q^{-1}\mathbf{x}) \tag{17}$$

C. Estimation of sources

After estimating the source activity vector, the estimation of sources is achieved with the estimation of amplitudes (r_i's), i.e. of amplitude vector ($\mathbf{r}$). The vector $\mathbf{r}$ has a Gaussian distribution and hence its MAP estimation is equal to its Linear Least Square (LLS) estimation [49]. The LLS estimation of $\mathbf{r}$ by knowing the source activity vector and the mixture vector has the following form [49]:

$$\widehat{\mathbf{r}}_{\text{LLS}} = E(\mathbf{r}\mathbf{x}^T|\mathbf{q})E(\mathbf{x}\mathbf{x}^T|\mathbf{q})^{-1}\mathbf{x} \tag{18}$$

The calculation of the two terms $E(\mathbf{r}\mathbf{x}^T|\mathbf{q})$ and $E(\mathbf{x}\mathbf{x}^T|\mathbf{q})$ results in the following equations:

$$E(\mathbf{r}\mathbf{x}^T|\mathbf{q}) = \sigma_r^2\mathbf{Q}\mathbf{A}^T \tag{19}$$

$$E(\mathbf{x}\mathbf{x}^T|\mathbf{q}) = \sigma_r^2\mathbf{A}\mathbf{Q}\mathbf{A}^T + \sigma_n^2\mathbf{I} \tag{20}$$

Then, the source amplitudes are estimated as:

$$\widehat{\mathbf{r}} = \sigma_r^2\mathbf{Q}\mathbf{A}^T(\sigma_r^2\mathbf{A}\mathbf{Q}\mathbf{A}^T + \sigma_n^2\mathbf{I})^{-1}\mathbf{x} \tag{21}$$

The maximization of $p(\mathbf{q})p(\mathbf{x}|\mathbf{q})$ may be done by exhaustive search over the discrete space of vectors $\mathbf{q}$ with 2^m discrete elements. This method can solve the SCA problem only for a small number of sources (for example $m < 12$). However, the complexity of this algorithm, based on an exhaustive search, can be alleviated with the IBA algorithm described in the next section.

IV. Iterative Bayesian Algorithm (IBA)

A. Basic Idea

In the MAP algorithm, the maximization of the posterior probability given by (17) is done by a combinatorial search over the discrete space. In this section, we propose a maximization method based on first converting the problem to a continuous maximization and then using a steepest-ascent algorithm. In this purpose, we use a mixture of two Gaussian model centered around 0 and 1 with sufficiently small variances. Thus, the discrete binomial variable q_i is converted to a variable with the following form:

$$q_i \sim pN(0,\sigma_0^2) + (1-p)N(1,\sigma_0^2) \tag{22}$$

To avoid local maxima of (17) a gradually decreasing variance is used in the different iterations (this is similar to what is done in simulated annealing algorithms, and to Graduated NonConvexity [50]). However, (17), as a cost function with respect to $\mathbf{q}$, is very complex to work with.

The main idea of our algorithm is that the source estimation is equivalent to estimation of vectors $\mathbf{q}$ and $\mathbf{r}$ (as observed from (3)) and can be done iteratively. First, an estimated vector $\widehat{\mathbf{q}}$ is assumed and then the MAP estimate of the vector $\mathbf{r}$ based on the known $\widehat{\mathbf{q}}$ and the observation vector $\mathbf{x}$ is obtained (we refer to it as $\widehat{\mathbf{r}}$). Secondly, the MAP estimate of vector $\mathbf{q}$ is obtained based on estimated vector $\widehat{\mathbf{r}}$ and observation vector $\mathbf{x}$ (we refer to it as vector $\widehat{\mathbf{q}}$). Therefore, the MAP estimation of sources is achieved through two other elementary MAP estimation steps.

In the first step, a source activity vector $\widehat{\mathbf{q}}$ is assumed and the estimation of $\mathbf{r}$ will be computed. Because the vector $\mathbf{r}$ is Gaussian, its MAP estimate is equal to the Linear Least Square (LLS) estimation [49] and can be computed as follows:

$$\widehat{\mathbf{r}}_{\text{MAP}} = \widehat{\mathbf{r}}_{\text{LLS}} = E(\mathbf{r}|\mathbf{x},\widehat{\mathbf{q}}) = E(\mathbf{r}\mathbf{x}^T|\widehat{\mathbf{q}})E(\mathbf{x}\mathbf{x}^T|\widehat{\mathbf{q}})^{-1}\mathbf{x} \tag{23}$$

This step can be called Expectation step or Estimation step (E-step). Computation and simplification of (23) (similar to what done in [24]) leads to the following equation which is similar to (21):

$$\widehat{\mathbf{r}} = \sigma_r^2\widehat{\mathbf{Q}}\mathbf{A}^T(\sigma_r^2\mathbf{A}\widehat{\mathbf{Q}}\mathbf{A}^T + \sigma_n^2\mathbf{I})^{-1}\mathbf{x} \tag{24}$$

In the second step which can be called Maximization step (M-step), we estimate $\mathbf{q}$ based on known $\widehat{\mathbf{r}}$ and observed $\mathbf{x}$. The MAP estimation is:

$$\widehat{\mathbf{q}}_{\text{MAP}} = \underset{\mathbf{q}}{\text{argmax}}\,p(\mathbf{q}|\mathbf{x},\widehat{\mathbf{r}}) \equiv \underset{\mathbf{q}}{\text{argmax}}\,p(\mathbf{q}|\widehat{\mathbf{r}})p(\mathbf{x}|\mathbf{q},\widehat{\mathbf{r}}) \equiv$$
$$\underset{\mathbf{q}}{\text{argmax}}\,p(\mathbf{q})p(\mathbf{x}|\mathbf{q},\widehat{\mathbf{r}}) \equiv \underset{\mathbf{q}}{\text{argmax}}(\log p(\mathbf{q}) + \log p(\mathbf{x}|\mathbf{q},\widehat{\mathbf{r}})) \tag{25}$$

Equation (22) implies that, in (25) $p(\mathbf{q})$ can be computed as a continuous variable:

$$p(\mathbf{q}) = \prod_{i=1}^{m}p(q_i) = \prod_{i=1}^{m}\left[p\exp(\frac{-q_i^2}{2\sigma_0^2})+(1-p)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})\right] \tag{26}$$

In addition, the term $p(\mathbf{x}|\mathbf{q},\widehat{\mathbf{r}})$ in (25) can be computed as:

$$p(\mathbf{x}|\mathbf{q},\widehat{\mathbf{r}}) = p_n(\mathbf{x}-\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}) =$$
$$(2\pi\sigma_n^2)^{\frac{-m}{2}}\exp\left(\frac{-1}{2\sigma_n^2}(\mathbf{x}-\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})^T(\mathbf{x}-\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})\right) \tag{27}$$

Consequently, (25) writes as:

$$\text{M-step}: \quad \widehat{\mathbf{q}} = \underset{q}{\text{argmax}}\,L(\mathbf{q}) \tag{28}$$

where

$$L(\mathbf{q}) = \sum_{i=1}^{m}\log(p(q_i)) - \frac{1}{2\sigma_n^2}(\mathbf{x}-\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})^T(\mathbf{x}-\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}) \tag{29}$$

Maximization of $L(\mathbf{q})$ in the M-step can be done by the steepest-ascent method. The main steepest-ascent iteration is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mu\frac{\partial L(\mathbf{q})}{\partial\mathbf{q}} \tag{30}$$

where μ is the step-size in the steepest-ascent method. After calculating the gradient (see Appendix A), the steepest-ascent algorithm for the M-step is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \frac{\mu}{\sigma_0^2}\mathbf{g}(\mathbf{q}) + \frac{\mu}{\sigma_n^2}\text{diag}(\mathbf{A}^T(\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}-\mathbf{x})).\widehat{\mathbf{r}} \tag{31}$$

where $\mathbf{g}(\mathbf{q})$, given in Appendix A by (58), depends on σ_0. In successive iterations of our simulations, we gradually decrease the variance σ_0 according to the series $\sigma_0^{(i)} = \alpha\sigma_0^{(i-1)}$ where α is selected between 0.6 and 1. In Section V, a suitable range on values of μ for the convergence of the proposed IBA algorithm is calculated. This suitable interval means that if the step size μ is selected in this interval, with a probability close to one, the convergence is insured. Assuming the columns of $\mathbf{A}$ are normalized to have unit norms, the suitable interval is:

$$0 < \mu < \frac{2}{\frac{1}{\sigma_0^2} + \frac{mM^{*2}}{\sigma_n^2}} \tag{32}$$

where $M^* = \sigma_r Q^{-1}(\frac{1-\sqrt[m]{0.99}}{2})$. As we see from (31) the second summand is responsible for increasing the prior probability $p(\mathbf{q})$ while the third summand is responsible for decreasing the noise power $||\mathbf{x} - \mathbf{As}||^2$. When σ_0 is much larger than σ_n, the second term is weaker than the third term and as a result the exactness of $\mathbf{x} = \mathbf{As}$ is more relevant than the sparsity of $\mathbf{s}$. When σ_0 is comparable to σ_n, both terms are effective to yield the equilibrium point between sparsity and noise.

In summary, the overall algorithm is a 3-step iterative algorithm the first two steps of which are E-step and M-step in (24) and (31), while the last step, explained in the next section, is the parameter estimation step.

B. Parameter estimation

So far the parameters (p, σ_r and σ_n) are assumed to be known. The parameter estimation is necessary and can also be done iteratively. We also assume that the columns of the matrix have unit norms. In other words, the basis atoms of our dictionary are normalized to have unit norm. With these assumptions and by assuming the ergodicity of sources (*i.e.* the mixtures can be considered as samples of a random variable $x_j = \sum_{i=1}^m a_{ji}s_i + e_j$ where $a_{ji} = b_{ji} \big/ \sqrt{b_{1i}^2 + b_{2i}^2 + ... + b_{ni}^2}$ and b_{ji} is a random variable with uniform distribution on [-1,1] and s_i and e_j are random variables), and by neglecting the noise power, we have $E(x_j^2) = mE(a_{ji}^2)E(s_i^2)$. In addition, we know that $\sum_{j=1}^n a_{ji}^2 = 1$ and hence we deduce $E(a_{ji}^2) = \frac{1}{n}$. Finally, from (8), we have $E(s_i^2) = (1-p)\sigma_r^2$, and therefore we can write:

$$\widehat{\sigma_r} = \sqrt{\frac{nE(x_j^2)}{m(1-\widehat{p})}} \tag{33}$$

With the initialization of $\widehat{p}$ with $\widehat{p}^{(0)}$, we will have $\widehat{\sigma_r} = \sqrt{\frac{nE(x_j^2)}{m(1-\widehat{p}^{(0)})}}$. For the starting noise variance, we choose $\widehat{\sigma_n}^{(0)} = \widehat{\sigma_r}/10$. In [27], we used the algorithm with a simple non iterative parameter estimation step that was stated above. But, here these simple non iterative estimates are just used as the initial estimation of the parameters. Along the iterations, we update the estimation of these parameters using the following simple equations:

$$\widehat{p} = \frac{||\mathbf{q}||_0}{m} \tag{34}$$

```
• Initialization:
    1) Let $\boldsymbol{\theta}_0$ equal to the initial parameter estimation:
       $\hat{p}^{(0)}$:  arbitrary value in [0.5 1],
       $\hat{\sigma}_r^{(0)} = \sqrt{\frac{nE(x_j^2)}{m(1-\widehat{p}^{(0)})}}$, $\hat{\sigma}_n^{(0)} = \frac{\hat{\sigma}_r^{(0)}}{10}$.
    2) Let $\mathbf{s}_0$, $\mathbf{q}_0$ and $\mathbf{r}_0$ equal to the initial solution from
       minimum $\ell^2$-norm:
       $\mathbf{s}_0 = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{x}$,
       $\mathbf{q}_0 = (\mathbf{s}_0 > Th)$, $\mathbf{r}_0 = \mathbf{s}_0.(\mathbf{s}_0 > Th)$.
• Until Convergence do:
    1) E-step: solution obtained in (24).
    2) M-step:
       – for $j = 1, \ldots, n_{\text{iteration}}$:
           * Update $\mathbf{q}$ with (31)
           * Update $\sigma_0^{(j)} = \alpha\sigma_0^{(j-1)}$
    3) Parameter Estimation Step: using (33), (34), (35) and
       (36).
• Final answer is $\hat{\mathbf{s}} = \mathbf{q}_{\text{final}}\mathbf{r}_{\text{final}}$.
```

Fig. 2. The final overall IBA algorithm. estimation.

$$\widehat{\sigma_n} = \frac{||\mathbf{x} - \mathbf{A}\widehat{\mathbf{s}}||_2}{\sqrt{n}} \tag{35}$$

$$\widehat{\sigma_r} = \frac{||\mathbf{r}||_2}{\sqrt{m}} \tag{36}$$

where it is assumed that the number of sources m is known in advance. In addition to (36), we can use (33) to update the estimation of σ_r in the iterations.

It is shown in Appendix B that the formulas (34), (35) and (36) are the MAP estimates of the corresponding parameters based on knowing all the other unknowns.

Finally, the complete IBA algorithm, including the parameter estimation step, is summarized in Fig. 2. Since the active sources are Gaussian with a variance σ_r^2, we choose the threshold parameter Th as a fraction of σ_r. The exact value of Th determines the number of active sources in the first step. We will discuss about this parameter in the simulation results section.

V. Convergence Analysis

In this section, we first prove that the log posterior probability $L(\mathbf{s}) \triangleq \log p(\mathbf{s}|\mathbf{x})$ is concave. Therefore, the unique maximum of this function is the global maximum. Then, we prove the convergence of the steepest-ascent algorithm in the M-step and find the suitable interval for the step-size (μ) for the convergence of the steepest-ascent algorithm. Finally, the convergence of the overall IBA algorithm will be proved.

A. Convexity of the posterior probability

The log posterior probability function $L(\mathbf{s})$ can be written as $L(\mathbf{s}) \propto \log p(\mathbf{s}) + \log p(\mathbf{x}|\mathbf{s})$, that we can decompose as:

$$L(\mathbf{s}) = \sum_{i=1}^m \log(p(s_i)) - \frac{1}{2\sigma_n^2}(\mathbf{x} - \mathbf{As})^T(\mathbf{x} - \mathbf{As}) \tag{37}$$

To prove the concavity of $L(\mathbf{s})$, we prove the concavity of each $\log(p(s_i))$ and the convexity of $c(\mathbf{s}) \triangleq (\mathbf{x} - \mathbf{As})^T(\mathbf{x} - \mathbf{As})$. Then, since the sum of concave functions is also concave, the proof is completed. As before, the probability distribution of sources in (2) is assumed to be a BG distribution $p(s_i) = \frac{p}{\sigma_1\sqrt{2\pi}}\exp(\frac{-s_i^2}{2\sigma_1^2}) + \frac{1-p}{\sigma_2\sqrt{2\pi}}\exp(\frac{-s_i^2}{2\sigma_2^2})$ with sufficiently small σ_1. To prove the concavity of $\log(p(s_i))$, we need to prove that its second derivative is negative. It is equivalent to prove that $h(s_i) \triangleq p(s_i)p''(s_i) - p'(s_i)^2 < 0$. Calculation of this function leads to the following formula:

$$\frac{1}{2\pi}h(s_i) = -\frac{p^2}{\sigma_1^4}\exp(\frac{-s_i^2}{\sigma_1^2}) - \frac{(1-p)^2}{\sigma_2^4}\exp(\frac{-s_i^2}{\sigma_2^2})$$
$$+\frac{p(1-p)}{\sigma_1\sigma_2}\left[(\frac{s_i}{\sigma_1^2} - \frac{s_i}{\sigma_2^2})^2 - \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right]\exp(\frac{-s_i^2}{2\sigma_1^2} - \frac{-s_i^2}{2\sigma_2^2}) \quad (38)$$

Let $s_i = 0$. Then $\frac{1}{2\pi}h(0) = -\frac{p^2}{\sigma_1^4} - \frac{p(p-1)}{\sigma_1\sigma_2}(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}) < 0$. Now assume that $s_i \neq 0$. If we denote $t \triangleq \frac{s_i^2}{\sigma_1^2}$, then some terms in (38) are in the form of $t^n\exp(-t)$. If $\sigma_1 \to 0$ or equivalently $t \to \infty$ then $t^n\exp(-t) \to 0$. Using these results, the limit of the function $h(s_i)$ is equal to:

$$\lim_{\sigma_1 \to 0} h(s_i) = -\frac{(1-p)^2}{\sigma_2^4}\exp(\frac{-s_i^2}{\sigma_2^2}) < 0 \quad (39)$$

which proves the concavity of $\log(p(s_i))$. The convexity of $c(\mathbf{s}) = (\mathbf{x} - \mathbf{A}\mathbf{s})^T(\mathbf{x} - \mathbf{A}\mathbf{s}) = ||\mathbf{x} - \mathbf{A}\mathbf{s}||^2$ is obvious due to its parabolic shape.

B. Convergence of the steepest-ascent

From (29) and by defining $\mathbf{s}_{k+1} \triangleq \mathbf{Q}_{k+1}\mathbf{r}_k$, $\mathbf{s}_k \triangleq \mathbf{Q}_k\mathbf{r}_k$, $\mathbf{H} \triangleq \mathbf{A}^T\mathbf{A}$ and $\mathbf{b} = 2\mathbf{x}\mathbf{A}^T$, we have:

$$L(\mathbf{q}_{k+1}) - L(\mathbf{q}_k) =$$
$$\sum_{i=1}^{m}\log\frac{p(q_{k+1,i})}{p(q_{k,i})} - \frac{1}{2\sigma_n^2}[\mathbf{s}_{k+1}^T\mathbf{H}\mathbf{s}_{k+1}^T - \mathbf{s}_k^T\mathbf{H}\mathbf{s}_k^T - \mathbf{b}(\mathbf{s}_{k+1} - \mathbf{s}_k)] \quad (40)$$

If the above expression is positive, then the sequence $L(\mathbf{q}_k)$ is an increasing sequence and since $L(\mathbf{q})$ is upper bounded by the value $m\log(p)$, then the sequence has a limit and the convergence of the M-step (steepest-ascent) will be proved. To derive the positivity condition of (40), we write the M-step iteration as $\mathbf{q}_{k+1} = \mathbf{q}_k + \mu\mathbf{c}_k$ where $\mathbf{c}_k = \left.\frac{\partial L(\mathbf{q})}{\partial\mathbf{q}}\right|_{\mathbf{q}=\mathbf{q}_k}$. Therefore, we have $\mathbf{s}_{k+1} - \mathbf{s}_k = \mu\mathbf{C}_k\mathbf{r}_k$ where $\mathbf{C}_k = \text{diag}(\mathbf{c}_k)$. Substituting in (40) results in:

$$L(\mathbf{q}_{k+1}) - L(\mathbf{q}_k) =$$
$$\sum_{i=1}^{m}\log\frac{p(q_{k+1,i})}{p(q_{k,i})} - \frac{\mu}{2\sigma_n^2}[\mu\mathbf{r}_k^T\mathbf{C}_k\mathbf{H}\mathbf{C}_k\mathbf{r}_k + (2\mathbf{s}_k^T\mathbf{H} - \mathbf{b})\mathbf{C}_k\mathbf{r}_k] \quad (41)$$

After some algebra detailed in Appendix C, one obtains the following inequality:

$$\frac{1}{\mu}[L(\mathbf{q}_{k+1}) - L(\mathbf{q}_k)] \geqslant \mathbf{c}_k^T[\mathbf{I} - \mu\mathbf{R} - \frac{\mu}{2\sigma_n^2}\mathbf{R}_1\mathbf{H}\mathbf{R}_1]\mathbf{c}_k \quad (42)$$

where $\mathbf{R}_1 = \text{diag}(r_i)$ and $\mathbf{R}$ is defined in Appendix C. If the symmetric matrix $\mathbf{D} \triangleq \mathbf{I} - \mu\mathbf{R} - \frac{\mu}{2\sigma_n^2}\mathbf{R}_1\mathbf{H}\mathbf{R}_1$ is Positive Definite (PD) then the sequence $L(\mathbf{q}_k)$ will be increasing, and, as mentioned above, the convergence of the M-step is guaranteed. For $\mathbf{D}$ being PD, all its eigenvalues should be positive. If we define $\mathbf{E} \triangleq \mathbf{R} + \frac{1}{2\sigma_n^2}\mathbf{R}_1\mathbf{H}\mathbf{R}_1$ then $\mathbf{D} = \mathbf{I} - \mu\mathbf{E}$. The PD property of $\mathbf{D}$ results in the following equation:

$$\mu < \frac{1}{\lambda_{\max}(\mathbf{E})} \quad (43)$$

where $\lambda_{\max}(\mathbf{E})$ stands for the maximum eigenvalue of $\mathbf{E}$. Since $\mathbf{R}$ is diagonal, $\lambda_i(\mathbf{E}) = \lambda_i(\mathbf{R}) + \frac{1}{2\sigma_n^2}\lambda_i(\mathbf{R}_1\mathbf{H}\mathbf{R}_1)$. Again since $\mathbf{R}$ is diagonal, its eigenvalues are $\lambda_i(\mathbf{R}) = \sum_{j=1}^{2}\frac{r_j(q_{k,i})}{2\sigma_0^2} = \frac{1}{2\sigma_0^2}$, then we have:

$$\lambda_{\max}(\mathbf{E}) < \frac{1}{2\sigma_0^2} + \frac{1}{2\sigma_n^2}\lambda_{\max}(\mathbf{R}_1\mathbf{H}\mathbf{R}_1) \quad (44)$$

To obtain an upper bound for the maximum eigenvalue of $\mathbf{E}$, we should find the upper bound for the maximum eigenvalue of $\mathbf{R}_1\mathbf{H}\mathbf{R}_1$. Since $\mathbf{R}_1\mathbf{H}\mathbf{R}_1 = (\mathbf{A}\mathbf{R}_1)^T(\mathbf{A}\mathbf{R}_1)$, it is PD and hence all of its eigenvalues are positive. Therefore we can write:

$$\lambda_{\max}(\mathbf{R}_1\mathbf{H}\mathbf{R}_1) \leqslant \sum_{i=1}^{m}\lambda_i = \text{trace}(\mathbf{R}_1\mathbf{H}\mathbf{R}_1) = \text{trace}(\mathbf{R}_1^2\mathbf{H}) \quad (45)$$

Since $\mathbf{R}_1 \triangleq \text{diag}(r_i)$, the elements of $\mathbf{R}_1$ and hence $\lambda_{\max}(\mathbf{R}_1\mathbf{H}\mathbf{R}_1)$ can theoretically be infinitely large. However, noting that $p(|r_i| < M) = 1 - 2Q(\frac{M}{\sigma_r})$, we can determine a suitable interval for the maximum value of $\lambda_{\max}(\mathbf{R}_1\mathbf{H}\mathbf{R}_1)$. For the cases which $|r_i| < M$, $\text{trace}(\mathbf{R}_1^2\mathbf{H}) = \sum_{i=1}^{m} r_i^2 h_{ii} < M^2\sum_{i=1}^{m} h_{ii}$ with probability $\gamma = (1 - 2Q(\frac{M}{\sigma_r}))^m$ which we want to be very close to 1 (for example $\gamma = 0.99$). Hence, we must choose the value as $M^* = \sigma_r Q^{-1}(\frac{1-\sqrt[m]{\gamma}}{2})$. Consequently, to be sure that with a probability γ close to 1 the convergence is guaranteed, μ should be selected in the interval:

$$0 < \mu < \frac{2}{\frac{1}{\sigma_0^2} + \frac{M^{*2}}{\sigma_n^2}\text{trace}(\mathbf{A}^T\mathbf{A})} \quad (46)$$

By imposing the condition of normalized column to $\mathbf{A}$, the diagonal elements of $\mathbf{A}^T\mathbf{A}$ is equal to one and hence we have $\text{trace}(\mathbf{A}^T\mathbf{A}) = m$. Therefore, the suitable interval can be simplified as:

$$0 < \mu < \frac{2}{\frac{1}{\sigma_0^2} + \frac{mM^{*2}}{\sigma_n^2}} \quad (47)$$

C. Convergence of the IBA algorithm

So far, we show the convergence of the M-step which is implemented with steepest-ascent method. To prove the convergence of the proposed IBA algorithm which is like an EM algorithm, we should prove that the sequence $L(\mathbf{s}_k)$ is increasing, where $L(\mathbf{s}_k)$ is the log posterior probability defined in (37). Increasing of this sequence is evident throughout the M-step because the $L(\mathbf{s})$ is equivalent to $L(\mathbf{q})$ throughout the M-step. In the E-step the LLS estimation of $\mathbf{r}$ is computed by (24). This estimation is also a MAP estimation. So, it maximizes our measure which is the posterior probability in the MAP sense. Since the logarithm function is monotonically increasing, MAP estimation is equivalent to maximizing the log posterior $L(\mathbf{s})$. So, the log posterior probability $L(\mathbf{s})$

increases in each E-step and M-step, inside any k-th iteration. So, in each iteration we can write:

$$L(\mathbf{s}_{\text{M}-\text{step}}^{(k+1)}) > L(\mathbf{s}_{\text{E}-\text{step}}^{(k+1)}) > L(\mathbf{s}_{\text{M}-\text{step}}^{(k)}) > L(\mathbf{s}_{\text{E}-\text{step}}^{(k)}) \quad (48)$$

where the sequences of sources are:

$$\mathbf{s}_{\text{E}-\text{step}}^{(k)} = \mathbf{Q}_{k-1}\mathbf{r}_k \quad (49)$$

$$\mathbf{s}_{\text{M}-\text{step}}^{(k)} = \mathbf{Q}_{k}\mathbf{r}_k \quad (50)$$

$$\mathbf{s}_{\text{E}-\text{step}}^{(k+1)} = \mathbf{Q}_{k}\mathbf{r}_{k+1} \quad (51)$$

$$\mathbf{s}_{\text{M}-\text{step}}^{(k+1)} = \mathbf{Q}_{k+1}\mathbf{r}_{k+1} \quad (52)$$

Therefore, the sequence of $L(\mathbf{s}_k)$ is converging to a local maximum $\mathbf{s}^*$. The concavity of the $L(\mathbf{s})$ guarantee that this local maximum is equal to the global maximum which is the MAP solution of sparse sources.

VI. Experimental results

In this section, the performance of the MAP algorithm and IBA algorithm is examined with concentrating on the IBA algorithm. The results of our algorithms are compared to the BP which is practically implemented by ℓ^1-magic [44] and some other algorithms which are in the literature. This will be done by discussing the results of five experiments detailing different aspects of our algorithms, especially the effects of the parameters, of noise and of sparsity.

The sparse sources are artificially generated using the BG model in (2). In all simulations, we used $p = 0.9$, $\sigma_n = 0.01$ and $\sigma_r = 1$ for the parameters of the BG model. A different source model will be used in the fourth experiment for which the number of active sources is fixed and the locations of the active sources are chosen randomly. The noise vector is an additive white Gaussian noise with covariance matrix $\sigma_n^2\mathbf{I}$. For investigating the noise effect, we define an input Signal to Noise Ratio as:

$$\text{Input-SNR} \triangleq 20\log\left(\frac{\sigma_r}{\sigma_n}\right) \quad (53)$$

The mixing matrix entries are chosen randomly from a uniform distribution in [-1,1] and then the columns of the mixing matrix are normalized to unity. To evaluate the performance of our algorithms, we use two definitions. In sparse decomposition experiments (single realization of $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}$), the Output Signal-to-Noise Ratio defined as:

$$\text{Output-SNR} = 20\log(||\mathbf{s} - \widehat{\mathbf{s}}||_2/||\mathbf{s}||_2) \quad (54)$$

is used as the performance measure. But in sparse component analysis experiments (many realizations of $\{\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)\}_{t=1}^N$), we usually average over time to obtain the *Individual SNR* for each source, defined by:

$$\text{SNR}_i = 10\log\left(\frac{\Sigma_{t=1}^N s_i^2(t)}{\Sigma_{t=1}^N (s_i(t) - \widehat{s}_i(t))^2}\right) \quad (55)$$

and the final performance index (SNR) is the average of these SNR's.

For each experiment, the performance indices are computed by averaging over 400 different realizations with new randomly chosen mixing matrix and sparse sources.

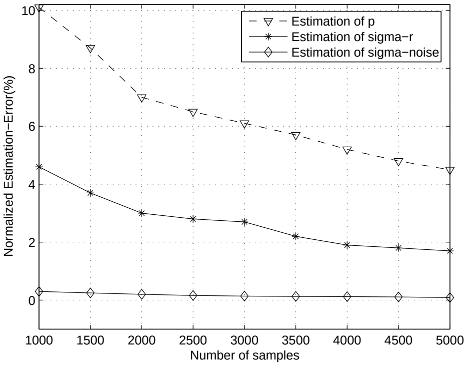


Fig. 3. The result of parameter estimation of the MAP algorithm in the case where $m = 8$, $n = 4$, $p = .9$, $\sigma_r = 1$ and $\sigma_n = .01$. The results are averaged on 400 simulations.

We use the CPU time as a measure of complexity. Although, the CPU time is not an exact measure, it can give us a rough estimation of the complexity for comparing our algorithm with other algorithms. Our simulations were performed in MATLAB7.0 environment using an Intel 2.80 GHz processor with 1024 MB of RAM and under Linux operating system.

A. Experiment 1 - Performance evaluation of the MAP algorithm

1) Parameter estimation: In this experiment, the main goal is to investigate the parameter estimation techniques described in subsection III-A. Eight sources ($m = 8$) are selected with source parameters $p = .9$, $\sigma_r = 1$ and the noise parameter $\sigma_n = .01$. Then, the mixture signals are obtained by the model (1). This experiment estimates the three unknown parameters from only N observations (or samples) of mixtures. Furthermore, 100 iterations are used for the EM algorithm for estimating σ_n. To measure the performance of estimating these parameters, we computed the normalized mean square error, defined as follows. Let (param) be the true parameter and $(\widehat{\text{param}}_i)$ be the estimated parameter in the i'th experiment, then the normalized mean square error (over the 400 realizations) is defined as:

$$e_n = \frac{\frac{1}{400}\sum_{i=1}^{400}(\text{param} - \widehat{\text{param}}_i)^2}{\text{param}^2} \quad (56)$$

The normalized mean square errors (in percent) versus the observation number are depicted in Fig. 3. As it can be seen, even for small number of samples (due to sparsity, the significant samples are then very few) the estimates for p, σ_r are acceptable. By increasing the number of observations, better estimates of parameters are obtained and the errors decrease.

2) Overall source estimation: In this experiment, the MAP algorithm is compared to BP. Moreover, to investigate the effect of parameter estimation error, the MAP algorithm was simulated in two cases: one with the actual parameters and the other with estimated parameters.

As before, the simulation parameters are $m = 8$, $p = .9$, $\sigma_r = 1$ and $\sigma_n = .01$. The number of samples is $N = 1000$

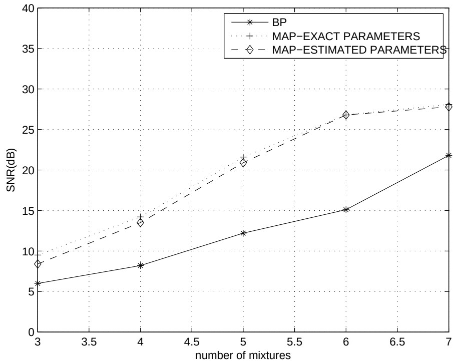


Fig. 4. The performance of MAP algorithm (with actual parameters and with estimated parameters) in comparison to LP method in the presence of noise, where $m = 8$, $N = 1000$, $p = .9$, $\sigma_r = 1$ and $\sigma_n = .01$. The results are averaged on 400 simulations.

and the number of mixtures (n) varies between 3 to 7. The results of three algorithms, namely BP, MAP with actual parameters and MAP with estimated parameters (all of them averaged on 400 simulations) are shown in Fig. 4. The results show that the average SNR (Temporal SNR) of the MAP algorithm is about 10dB better than BP. Moreover, the averaged SNR of the MAP algorithm with estimated parameters is close to the averaged SNR with actual parameters. Consequently, the MAP algorithm is robust to the parameter errors: in fact, for $N = 1000$, although the relative estimation errors of p and σ_r achieve about 10% and 5% (Fig. 3), the SNR loss with estimated parameters is less than 1dB (Fig. 4).

B. Experiment 2- Performance evaluation of the IBA algorithm

In this section, the performance of the IBA algorithm is investigated, and compared with different algorithms for various noise levels i.e. input SNR's. In this experiment, the simulation parameters are $m = 1024$, $n = 512$, $p = .9$, $\sigma_n = 0.01$ and $\sigma_r = 1$. In the M-step the value of α can be chosen between 0.6 and 0.9 (see the third experiment which investigates the parameter selection). In this experiment, we select $\alpha = 0.8$. The initial value of σ_0 is equal to 1. In this special case the suitable interval of μ in (32) is $0 < \mu < 2.1466 \times 10^{-7}$. So, we chose $\mu = 10^{-7}$. For the initial value of the parameter p, we selected the value $\hat{p}^{(0)} = 0.8$ because it is a good trade-off, and the initial value must imply a sufficient number of active sources. The practical value of Th in Fig. 2, is selected as $\frac{\hat{\sigma}_r^{(0)}}{4}$ because it provides slightly better results (refer to experiment 3). The simulations show that, for these parameters, only 4 or 5 iterations are sufficient to maximize the expression $L(\mathbf{q})$ in the M-step. IBA algorithm usually converges at the third and fourth iterations in our simulations. Therefore, 5 iterations are used for the M-step and for the overall IBA algorithm, we stop the algorithm when the condition $\frac{||\mathbf{s}^{(k)} - \mathbf{s}^{(k-1)}||_2}{||\mathbf{s}^{(k)}||_2} < 0.001$ is satisfied.

Table I shows the progress of iterative parameter estimation to actual parameters. The results of output SNR versus input SNR are shown in Fig. 5 for our IBA algorithms and some other algorithms.

TABLE I
PROGRESS OF THE ITERATIVE PARAMETER ESTIMATION TO ACTUAL PARAMETERS IN THE CASE OF $m = 1024$, $n = 512$ AND $p = 0.9$ $\sigma_r = 1$ $\sigma_n = .01$.

itr. #	p	σ_r	σ_n
1	0.8000	0.7412	0.0782
2	0.8400	0.8310	0.0524
3	0.9070	1.0412	0.0082
4	0.9119	1.1206	0.0048
5	0.9119	1.1206	0.0055

For comparison, we selected 4 representative algorithms. The first algorithm is OMP which is a pursuit algorithm which uses correlations between the signal and the atoms for selecting the current active atom. It also projects the residual signal to the surface of active atoms to update the coefficients at the end of each iteration. The second algorithm is BP which replaces ℓ^0-norm by ℓ^1-norm for the sparsity measure and uses LP methods. The third algorithm is smoothed-ℓ^0 (SL0) which replaces the ℓ^0-norm by a smoothed approximation of ℓ^0-norm and then uses a steepest-descent method to find the sparse coefficient. Finally, the fourth algorithm is BCS algorithm which is a Bayesian algorithm which uses a hierarchical sparsness prior and a type-II Maximum Likelihood (ML) procedure like the RVM method to estimate the unknown variables.

The number of iterations are 100 for OMP. For smoothed-ℓ^0, we used a sequence of σ as $\sigma^{(k)} = 0.9\sigma^{(k-1)}$ from the initial estimated value to the final value 0.04 and $L = 3$ and $\mu = 2$ [25]. For BCS and BP, we used the author recommendations for choosing the parameters. As we can see from the figure, our performance is somewhat better than other methods like BP, OMP and SL0. Only, BCS provides better performance specially for low SNR's.

We ran another experiment for sources, different of the BG model. In this experiment, the sources are fixed and equal to 1 from index 1 to 20 and also are fixed and equal to 0 for other indexes (from 21 to 1024). The results for this special case are shown in Fig. 6. In this special case, the IBA algorithm achieves the second best performance, just after BCS algorithm.

C. Experiment 3- Dependence on the parameters

In this experiment, the effect of the simulation parameters (α and Th in Fig. 2) is investigated.

1) Effect of α: In this experiment, we consider the effect of the parameter α, which is the scale factor controlling the decreasing rate of σ_0 ($\sigma_0^{(i)} = \alpha\sigma_0^{(i-1)}$). The experiment results are shown in Fig. 7, which represents the averaged output-SNR versus α, for different values of σ_n (Fig. 7(a)) and $k = m(1-p)$ (Fig. 7(b)). The value σ_n determines the noise standard deviation. The value $k = m(1-p)$, which represents the average number of active sources, determines the degree of sparsity of the sources.

It is clear from Fig. 7(a) that the value of α can be selected properly between a lower bound (0.6) and an upper bound (0.9). In this experiment the values of other parameters are

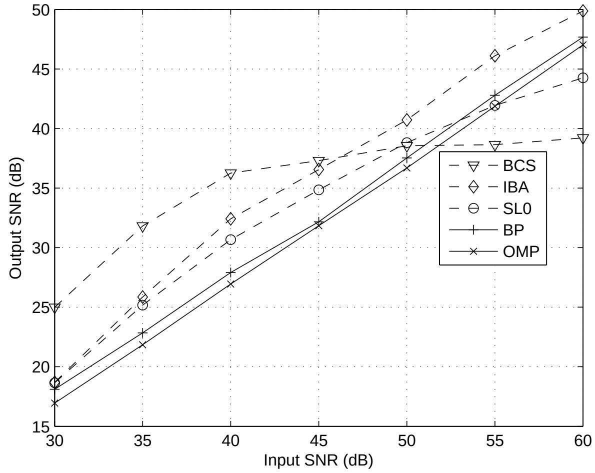


Fig. 5. Output-SNR vs Input-SNR. Results of the IBA algorithm compared with other algorithms. The simulation parameters are $m = 1024$, $n = 512$, $p = .9$, $\sigma_r = 1$, $\alpha = .8$, and $\mu = 10^{-7}$. Five iterations are used for the M-step (steepest-ascent). Results are averaged on 400 simulations.

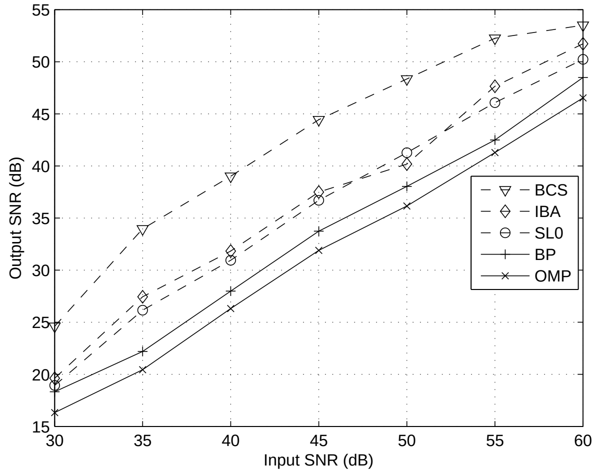


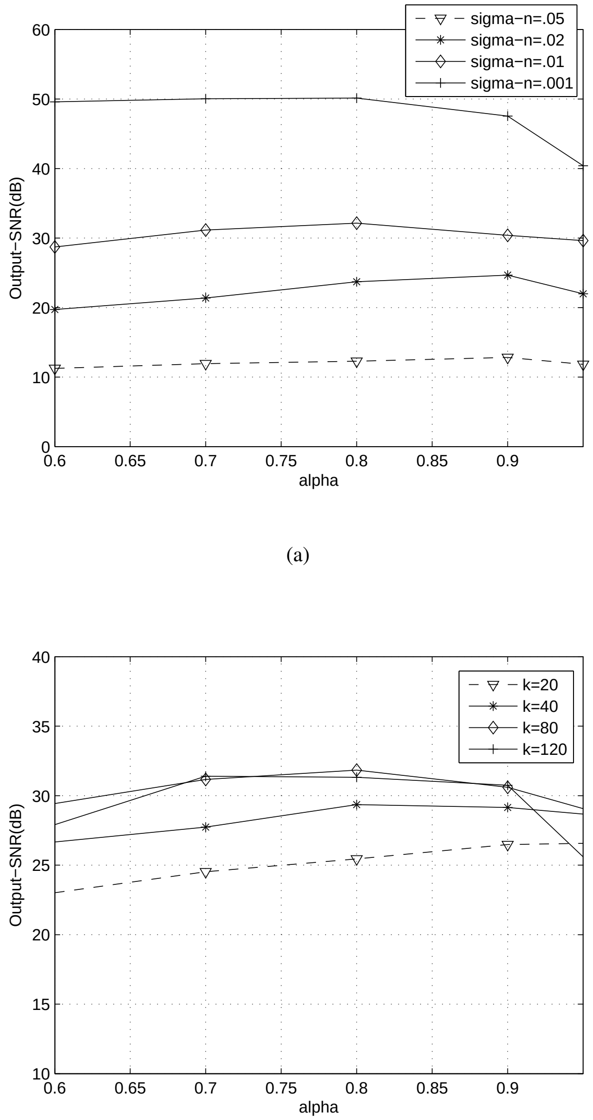
Fig. 6. Results of the IBA algorithm compared with other algorithms, for non BG sources. The simulation parameters are $m = 1024$, $n = 512$, $p = .9$, $\sigma_r = 1$, $\alpha = .8$, and $\mu = 10^{-7}$. Five iterations are used for the M-step (steepest-ascent). Results are averaged on 400 simulations.

$\mu = 10^{-7}$, $k = m(1-p) = 100$. The performance decreases when α is too close to one (in this case .95). However, our experiments show the low sensitivity of our algorithm to this parameter. A good choice of this parameter is around 0.8.

From [1] and [2], we know that $k \leq n/2$ is a theoretical limit for uniqueness of the sparsest solution. In Fig. 7(b), the results of the SNR versus α in various sparsity conditions ($20 \leq k \leq 120$) are shown. Again, our experiments show the IBA algorithm is not too sensitive to the parameter α: $\alpha = 0.8$ is still a good choice.

2) Effect of Th: In this experiment, the effect of the threshold Th in Fig 2 is investigated. We define a threshold factor as:

$$\text{Threshold-factor} \triangleq \frac{\hat{\sigma_r}}{Th}$$



(a)

(b)

Fig. 7. Performance of the IBA algorithm vs parameter α for $m = 1024$, $n = 512$ and $\sigma_r = 1$. In (a) k (number of active sources) is fixed to 100 and the effect of noise is investigated. In (b) σ_n is fixed to 0.01 and the effect of sparsity factor is analyzed. Values of k are 20, 40, 80, 120. Results are averaged on 400 simulations.

We ran the IBA algorithm with various values of this threshold factor. The results are shown in Fig 8 which represents the output-SNR versus the threshold factor. Although the algorithm has a low sensitivity to this threshold, but the value for the threshold factor should be less than 5. Arbitrarily, we chose a threshold factor equal to 4, and $Th = \frac{\hat{\sigma_r}}{4}$ in the IBA algorithm.

D. Experiment 4 - Effect of sparsity on the performance

Here, we experimentally consider the question: How much sparse a source should be to make the decomposition possible? As mentioned before, we have the theoretical limit of $n/2$

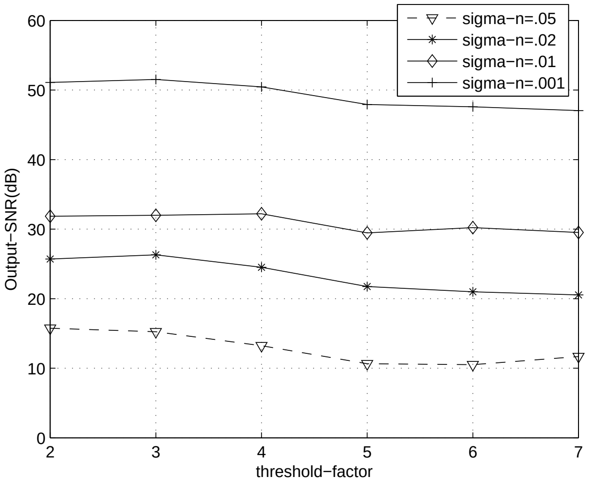


Fig. 8. The effect of the threshold in the IBA algorithm. The simulation parameters are $m = 1024$,$n = 512$, $p = .9$, $\sigma_r = 1$ and $\mu = 10^{-7}$. Results are averaged on 400 simulations.

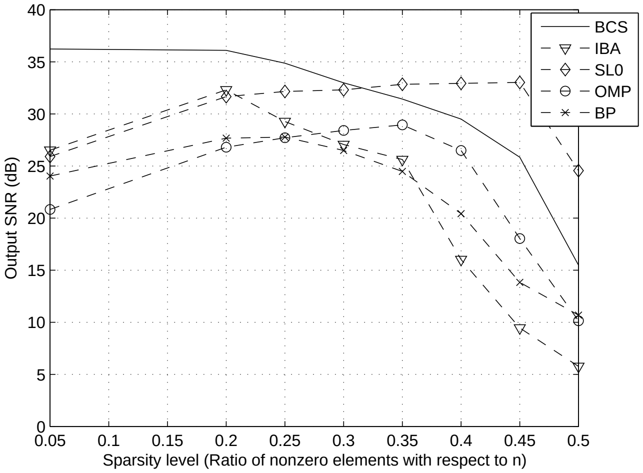


Fig. 9. Output-SNR as a function of sparsity level (active sources/n) for various algorithms. Simulation parameters are $m = 1024$, $n = 512$, $\sigma_r = 1$ and $\sigma_n = 0.01$. Results are averaged on 400 simulations.

on the maximum number of active sources at each sample (column) of source matrix to insure the uniqueness of the sparsest solution. But most of SCA methods are unable to practically achieve this limit [1].

To be able to measure the effect of sparsity, instead of generating the sources according to BG model (2), at each time t, we activate exactly k entries out of m components of $\mathbf{s}(t)$, (the column t of the source matrix). The locations and the values of these k elements are chosen randomly. k is related to the sparsity level of the sources, defined as:

$$\text{Sparsity-level} \triangleq \frac{k}{n}$$

where k is the number of active sources.

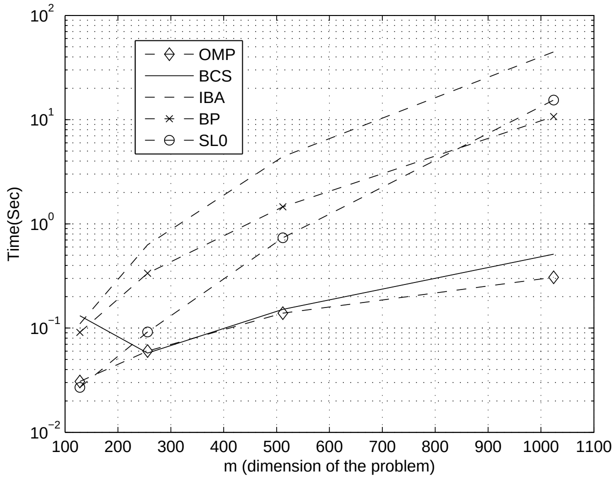


Fig. 10. Average CPU time in seconds versus problem dimension (m). For all dimensions $n = 0.5m$. The parameters are $p = .9$, $\sigma_r = 1$ and $\sigma_n = .01$. The results are averaged on 400 simulations.

Figure 9 shows the output SNR (averaged on 400 simulations), as a function of sparsity level, for several values of α and compares the results with other algorithms. It can be seen that the IBA algorithm achieves a good performance especially when the sparsity level is low.

E. Experiment 5 - Complexity

In this last experiment, the relative complexity (or speed) of the IBA algorithm is compared to other algorithms. The measure to be used is the 'average CPU time' required by each algorithm. More specifically, we plot 'average time' versus m, where m is the number of sources. The number of mixtures is $n = 0.5m$. The various values of m are 128, 256, 512 and 1024. Figure. 10 summarizes the results of average simulation times (on 400 simulation) for recovering the sources at one time step. As we can see, the IBA algorithm has relatively a high complexity, a price paid for its good performance.

VII. Conclusions

In this article, we have presented a new IBA algorithm for finding sparse solutions of underdetermined system of linear equations in the context of SCA. This IBA algorithm is based on iterative MAP estimation of the sources. In this paper, the high complexity of Bayesian methods is reduced by an iterative algorithm which resembles the EM algorithm. The M-step is done by a steepest-ascent method. Its step size should be chosen properly for insuring convergence of the steepest-ascent and of the global IBA algorithm. Moreover, the paper includes the convergence analysis of the algorithm and the convergence proof to the true optimum solution. Experimental results show the IBA algorithm has a low sensitivity to the simulation parameters, and achieves good performance, although not the best, compared with other algorithms especially BCS, which is currently the best algorithm, both for accuracy and speed. Future works includes improvement of Bayesian algorithms, by explicitely taking into account the noise for achieving performance as optimum as possible.

APPENDIX A
DERIVATION OF THE STEEPEST-ASCENT

From (28), we have:

$$\frac{\partial L(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \sum_{i=1}^{m} \log(p(q_i)) - \frac{1}{2\sigma_n^2} \frac{\partial}{\partial \mathbf{q}} (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})^T (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}) \tag{57}$$

We define $\mathbf{g}(\mathbf{q}) \triangleq -\sigma_0^2 \frac{\partial}{\partial \mathbf{q}} \sum_{i=1}^{m} \log(p(q_i))$ and $\mathbf{n}(\mathbf{q}) \triangleq (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})^T (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})$. With these definitions the scalar function $g(q_i)$ and the $\mathbf{n}(\mathbf{q})$ (with omitting the constant terms) can be computed as:

$$g(q_i) = \frac{p q_i \exp(\frac{-q_i^2}{2\sigma_0^2}) + (1-p)(q_i - 1)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})}{p \exp(\frac{-q_i^2}{2\sigma_0^2}) + (1-p)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})} \tag{58}$$

$$\mathbf{n}(\mathbf{q}) = -2\mathbf{x}^T \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} + \widehat{\mathbf{r}}^T \mathbf{Q}\mathbf{A}^T \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} \tag{59}$$

With the definitions $\mathbf{C} \triangleq \mathbf{A}^T\mathbf{A}$, $\mathbf{n_1}(\mathbf{q}) \triangleq -2\mathbf{x}^T \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}$ and $\mathbf{n_2}(\mathbf{q}) \triangleq \widehat{\mathbf{r}}^T \mathbf{Q}\mathbf{C}\mathbf{Q}\widehat{\mathbf{r}}$, we can write:

$$\frac{\partial n_1(\mathbf{q})}{\partial \mathbf{q}} = \mathrm{diag}(-2\mathbf{x}^T\mathbf{A}).\widehat{\mathbf{r}} \tag{60}$$

If we define $\mathbf{W} \triangleq \mathbf{Q}\widehat{\mathbf{r}}$ ($m \times 1$ vector) then $\mathbf{n}_2(\mathbf{q}) = \mathbf{W}^T\mathbf{C}\mathbf{W}$ and so we have:

$$\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial q_i} = \sum_{j=1}^{m} \frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial W_j} \frac{\partial W_j}{\partial q_i} \tag{61}$$

From the vector derivatives, we have $\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial \mathbf{W}} = 2\mathbf{C}\mathbf{W} \triangleq \mathbf{d}$. And from the definition of $\mathbf{W}$ we get $\frac{\partial W_j}{\partial q_i} = \widehat{r}_i \delta_{ij}$. So (61) is converted to $\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial q_i} = \sum_{j=1}^{m} d_j \widehat{r}_i \delta_{ij} = \widehat{r}_i d_i$. So the vector form of (61) is equal to:

$$\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial \mathbf{q}} = \mathrm{diag}(\mathbf{d}).\widehat{\mathbf{r}} \tag{62}$$

From (60), (62), $\mathbf{n}(\mathbf{q}) = \mathbf{n}_1(\mathbf{q}) + \mathbf{n}_2(\mathbf{q})$ and definitions of vectors $\mathbf{d}$ and $\mathbf{C}$, we can write:

$$\frac{\partial \mathbf{n}(\mathbf{q})}{\partial \mathbf{q}} = 2\mathrm{diag}(\mathbf{A}^T\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} - \mathbf{A}^T\mathbf{x}).\widehat{\mathbf{r}} \tag{63}$$

Finally, (63), (57), (30) with the definitions of $\mathbf{n}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ yields the steepest-ascent iteration (31).

APPENDIX B
MAP ESTIMATION OF PARAMETERS

To compute the MAP estimate of σ_r, assuming the other parameters are known, we should maximize the posterior probability $p(\sigma_r|\hat{\mathbf{r}}, \hat{\mathbf{q}}, \hat{\sigma_n}, \hat{p}, \mathbf{x}) \equiv p(\sigma_r)p(\hat{\mathbf{r}}|\sigma_r)$. We do not impose any prior information about σ_r. So, the MAP estimation of σ_r should maximize $p(\hat{\mathbf{r}}|\sigma_r)$. This distribution is equal to:

$$p(\hat{\mathbf{r}}|\sigma_r) = (2\pi\sigma_r^2)^{-\frac{m}{2}} \exp(\frac{-1}{2\sigma_r^2}\hat{\mathbf{r}}^T\hat{\mathbf{r}}) \tag{64}$$

Differentiating the above equation with respect to σ_r, the MAP estimate of σ_r is :

$$\hat{\sigma}_{r,\mathrm{MAP}}^2 = \frac{1}{m}||\mathbf{r}||_2^2 \tag{65}$$

To compute the MAP estimate of σ_n, assuming the other parameters are known, we must maximize the posterior probability $p(\sigma_n|\hat{\mathbf{q}}, \hat{\mathbf{r}}, \hat{\sigma_r}, \hat{p}, \mathbf{x}) \equiv p(\mathbf{x}|\sigma_n, \hat{\mathbf{r}}, \hat{\mathbf{q}}) = p_n(\mathbf{x} - \mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{r}})$. This probability distribution is equal to:

$$p_n(\mathbf{x} - \mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{r}}) = (2\pi\sigma_n^2)^{-\frac{n}{2}} \exp(\frac{-1}{2\sigma_n^2}(\mathbf{x} - \mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{r}})^T(\mathbf{x} - \mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{r}})) \tag{66}$$

Differentiating the above equation with respect to σ_n yields to the following equation:

$$\hat{\sigma}_{n,\mathrm{MAP}}^2 = \frac{||\mathbf{x} - \mathbf{A}\hat{\mathbf{s}}||_2^2}{n} \tag{67}$$

Finally, to calculate the MAP estimate of p, assuming the other parameters are known, the posterior probability $p(p|\hat{\mathbf{q}}, \hat{\mathbf{r}}, \hat{\sigma_r}, \hat{\sigma_n}, \mathbf{x})$ should be maximized. This probability is equivalent to $p(\hat{\mathbf{q}}|p)p(\mathbf{x}|\hat{\mathbf{q}}, \hat{\mathbf{r}}, \hat{\sigma_n})$ in which only the term $p(\hat{\mathbf{q}}|p)$ depends on p. Differentiating $p(\hat{\mathbf{q}}|p) = p^{(m-n_a)}(1-p)^{n_a}$ with respect to p and setting it to zero, results in the following estimate of p:

$$\hat{p}_{\mathrm{MAP}} = \frac{m - n_a}{m} = \frac{||\mathbf{q}||_0}{m} \tag{68}$$

APPENDIX C
DERIVING INEQUALITY

For calculating the first term of (41) $B \triangleq \sum_{i=1}^{m} \log \frac{p(q_{k+1,i})}{p(q_{k,i})}$, we define $p(q_i) \triangleq \sum_{j=1}^{2} \pi_j g_j(q_i)$ where $\pi_1 \triangleq p$ and $\pi_2 \triangleq 1 - p$ and $g_j(q_i) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp(\frac{-(q_i - m_j)^2}{2\sigma_0^2})$. Based on (22), we selected the two variances equal to σ_0 and $m_1 = 0$ and $m_2 = 1$. Then we will have:

$$B = \sum_{i=1}^{m} \log\left\{\frac{\sum_j \pi_j g_j(q_{k,i} + \mu c_{k,i})}{\sum_j \pi_j g_j(q_{k,i})}\right\}$$

$$B = \sum_{i=1}^{m} \log\left\{\frac{\sum_j \pi_j g_j(q_{k,i}) \exp(\frac{-\mu^2 c_{k,i}^2 - 2\mu c_{k,i}(q_{k,i} - m_j)}{2\sigma_0^2})}{\sum_j \pi_j g_j(q_{k,i})}\right\} \tag{69}$$

where $c_{k,i}$ is i'th element of $\mathbf{c}_k$. By defining $r_j(q_{k,i}) \triangleq \frac{\pi_j g_j(q_{k,i})}{\sum_j \pi_j g_j(q_{k,i})}$ (note that $0 < r_j(q_{k,i}) < 1$) and by considering the convexity of the logarithm function, we can write:

$$B = \sum_{i=1}^{m} \log\{\sum_{j=1}^{2} r_j(q_{k,i}) \exp(\frac{-\mu^2 c_{k,i}^2 - 2\mu c_{k,i}(q_{k,i} - m_j)}{2\sigma_0^2})\}$$

$$\geqslant \sum_{i=1}^{m}\sum_{j=1}^{2} \frac{r_j(q_{k,i})}{2\sigma_0^2}[-\mu^2 c_{k,i}^2 - 2\mu c_{k,i}(q_{k,i} - m_j)] \tag{70}$$

After some simplifications, the lower bound of B in (70) becomes:

$$C \triangleq \sum_{i=1}^{m} -\mu^2 c_{k,i}^2 \frac{\gamma_{k,i}}{2q_{k,i}} + \sum_{i=1}^{m} -\mu c_{k,i}\gamma_{k,i} + \sum_{i=1}^{m} \mu c_{k,i} \frac{r_2(q_{k,i})}{\sigma_0^2} \tag{71}$$

where $\gamma_{k,i} \triangleq q_{k,i} \sum_{j=1}^{2} \frac{r_j(q_{k,i})}{\sigma_0^2}$. We know that $\mathbf{c} = \frac{\partial L(\mathbf{q})}{\partial \mathbf{q}}$. So we have:

$$c_i = \frac{-1}{2\sigma_n^2}\frac{\partial}{\partial q_i}(\mathbf{x} - \mathbf{A}\mathbf{s})^T(\mathbf{x} - \mathbf{A}\mathbf{s}) + \frac{\partial}{\partial q_i}(\sum_{i=1}^{m} \log \sum_{j=1}^{2} \Pi_j p_j(q_i)) \tag{72}$$

where c_i is the i'th element of $\mathbf{c}$. If we define $N \triangleq (\mathbf{x}-\mathbf{As})^T(\mathbf{x}-\mathbf{As})$ and $M = \sum_{i=1}^m \log \sum_{j=1}^2 \pi_j g_j(q_i)$, then we can write $\frac{\partial N}{\partial q_i} = \frac{\partial N}{\partial s_i}\frac{\partial s_i}{\partial q_i} = [2\mathbf{A}^T(\mathbf{As}-\mathbf{x})]_i r_i$. Moreover, we have:

$$\frac{\partial M}{\partial q_i} = \frac{\pi_1 \frac{d}{dq_i} g_1(q_i) + \pi_2 \frac{d}{dq_i} g_2(q_i)}{\sum_{j=1}^2 \pi_j g_j(q_i)} \tag{73}$$

After some simplifications, we have:

$$\frac{\partial M}{\partial q_i} = -q_i \sum_{j=1}^{2} \frac{r_j(q_i)}{\sigma_0^2} + \frac{r_2(q_i)}{\sigma_0^2} \tag{74}$$

Therefore, we have:

$$\mathbf{c}_k = -\mathbf{z}_k - \gamma_k + \frac{\mathbf{r}_2(\mathbf{q}_k)}{\sigma_0^2} \tag{75}$$

where $\mathbf{r}_2(\mathbf{q}) = [r_2(q_1), r_2(q_2), ..., r_2(q_m)]^T$ and $r_2(q_i) \triangleq \frac{\pi_2 g_2(q_i)}{\sum_j \pi_2 g_2(q_i)}$ and $z_i \triangleq \frac{1}{2\sigma_n^2}[2\mathbf{A}^T(\mathbf{As}-\mathbf{x})]_i r_i$ and $\gamma_i \triangleq q_i \sum_{j=1}^2 \frac{r_j(q_i)}{\sigma_0^2}$. Finally, the lower bound C in (71) can be simplified as:

$$C = -\mu^2 \mathbf{c}_k^T \mathbf{R} \mathbf{c}_k - \mu \mathbf{c}_k^T \gamma_k + \frac{\mu}{\sigma_0^2} \mathbf{c}_k^T \mathbf{r}_2(\mathbf{q}_k) \tag{76}$$

where $\mathbf{R} \triangleq \mathrm{diag}(\sum_{j=1}^2 \frac{r_j(q_{k,i})}{2\sigma_0^2}) = \mathrm{diag}(\frac{\gamma_{k,i}}{2q_{k,i}})$. From (41), (69), (70), (71) and (76), we have:

$$\boldsymbol{L}(\mathbf{q}_{k+1}) - L(\mathbf{q}_k) \geqslant -\mu^2 \mathbf{c}_k^T \mathbf{R} \mathbf{c}_k - \mu \mathbf{c}_k^T \gamma_k$$

$$+\frac{\mu}{\sigma_0^2} \mathbf{c}_k^T \mathbf{r}_2(\mathbf{q}_k) - \frac{\mu^2}{2\sigma_n^2} \mathbf{r}^T \mathbf{C}_k \mathbf{H} \mathbf{C}_k \mathbf{r} - \frac{\mu}{2\sigma_n^2}(2\mathbf{s}_k^T \mathbf{H} - \mathbf{b}) \mathbf{C}_k \mathbf{r} \tag{77}$$

Because of the definition of z_i and $\mathbf{H}$ and with some manipulation the last term in the above equation $\frac{\mu}{2\sigma_n^2}(2\mathbf{s}_k^T \mathbf{H} - \mathbf{b})\mathbf{C}_k \mathbf{r}$ is simplified as $\mu \mathbf{z}_k^T \mathbf{c}_k$. Replacement $\gamma_k = -\mathbf{c}_k - \mathbf{z}_k + \frac{\mathbf{r_2}(\mathbf{q}_k)}{\sigma_0^2}$ from (75) into (77), results in (42).

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their fruitful comments and suggestions.

REFERENCES

[1] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," *Comm. Pure and Applied Math*, vol. 59, no. 6, pp. 797–829, 2006.

[2] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Info. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec 2003.

[3] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[4] D. L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Info. Theory*, vol. 52, no. 1, pp. 6–18, Jan 2006.

[5] M. Davies and N.Mitianoudis, "Simple mixture model for sparse overcomplete ICA," *IEE Proceeding. on Visual image and Signal Proc.*, pp. 35–43, Feb 2004.

[6] Y.Q. Li, S. Amari, A. Cichocki, D.W.C. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Trans. on Signal Proc.*, vol. 54, no. 2, pp. 423–437, 2006.

[7] P. G. Georgiev, F. J. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," in *IEEE Trans. on Neural Networks*, vol. 16, pp. 992–996 July 2005.

[8] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges," in *Proceedings of ESANN'06*, pp. 323–330, April 2006.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[10] J. L. Stark, M. Elad, and D. Donoho, "Image decomposition via the combination of sparse representataions and a variational approach," *IEEE Trans. on Image Proc.*, vol. 14, no. 10, pp. 1570–1582, October 2005.

[11] M. Elad, R. Goldenberg, and R. Kimmel, "Low bit-rate compression of facial images," *IEEE Trans. on Image Proc.*, vol. 16, no. 9, pp. 2379–2383, Sep 2007.

[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Proc.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[13] O. G. Guleryuz, "Nonlinear Approximation Based Image Recovery Using Adaptive Sparse Reconstructions and Iterated Denoising-part I: Theory," *IEEE Trans. on Image Proc.*, vol. 15, no. 3, pp. 539–554, March 2006.

[14] E. J. Candès and T. Tao, "Decoding by Linear Programming," *IEEE Trans. Info. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[15] M. Akcakaya, V. Tarokh, "On Sparsity, Redundancy and quality of frame representations," *Proceedings of ISIT2007*, pp. 951–955, June 2007.

[16] H. Zayyani, M. Babaie-zadeh, and C. Jutten, "Decoding real-field codes by an iterative Expectation-Maximization (EM) algorithm," *Proceedings of ICASSP'08*, pp. 3169-3172, April 2008.

[17] D. L. Donoho, "Compressed Sensing," *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[18] T. Blumensath and M. Davis, "Compressed sensing and source separation," *Proceedings of ICA'07*, pp. 341–348, Sep 2007.

[19] L. Vielva, D. Erdogmus, and C. Principe, "Underdetermined blind source separation using a probabilistic source sparsity model," in *Proceedings of ICA'01*, pp. 675–679, 2001.

[20] C. Fevotte and S.J. Godsill, "A Bayesian approach for blind separation of sparse sources," *IEEE Trans. on Speech and Audio Proc.* , vol. 14, no. 6, pp. 2174–2188, Nov 2006.

[21] I.F. Gorodnitski and B.D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted norm minimization algorithm," *IEEE Trans. on Signal Proc.*, vol. 45, pp. 600–616, 1997.

[22] M. Babaie-Zadeh, C. Jutten, and A. Mansour, "Sparse ICA via clusterwise PCA," *Neurocomputing*, vol. 69, pp. 1458–1466, Aug 2006.

[23] A. A. Amini, M. Babaie-Zadeh, and C. Jutten, "A fast method for sparse component analysis based on iterative detection-projection," *MaxEnt'06*, in AIP Conference Proc. 872, pp. 123–130, 2006.

[24] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Source estimation in noisy sparse component analysis," *15'th Intl. Conf. on Digital Signal Processing (DSP2007)* , pp. 219-222, July 2007.

[25] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0norm," *IEEE Trans. on Signal Proc.*, vol. 57, no. 1, pp. 289–301, January 2009.

[26] G.H. Mohimani, M. Babaie-zadeh, and C. Jutten, " Fast sparse representation based on smoothed ℓ^0-norm," *Proceedings of ICA'07*, pp. 389-396, Sep 2007.

[27] H. Zayyani, M. Babaie-Zadeh, G.H. Mohimani, and C. Jutten, "Sparse Component Analysis in Prsence of Noise Using an Iterative EM-MAP algorithm," *Proceedings of ICA'07*, pp. 438-445, Sep 2007.

[28] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[29] M. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, pp. 337–365, 2000.

[30] S. Krstulovic and R. Gribonval, "MPTK: Matching pursuit made tractable," *Proceedings of ICASSP'06*, Vol. 3, 14-19 May 2006.

[31] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with application to wavelet decomposition," in Conf. Rec. 27th Asilomar Conf.Signals, syst. Comput., vol. 1, 1993.

[32] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," Tech. Rep., 2006. Available at URL:`http://www-stat.stanford.edu/˜idrori/STOMP.pdf`.

[33] D. Wipf and B. D. Rao, "Sparse Bayesian Learning for basis selection," *IEEE Trans. on Signal Proc.*, vol. 52, no. 8, pp. 2153–2164, 2004.

[34] S. Ji, Y. Xue, and L. Carin, "Bayesian Compressive Sensing," *IEEE Trans. on Signal Proc.*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[35] T. Blumensath and M. Davies, “Gradient Pursuits,” *IEEE Trans. on Signal Proc.*, vol. 56, pp. 2370–2382, June 2008.

[36] M. A. T. Figueirado, R. D. Nowak, and S. J. Wright, “Gradient Projection for Sparse Reconstruction: Application to compressed sensing and other Inverse Problems,” *To appear in the IEEE Journal of selected topics in signal processing.*, 2007. Available at URL:`http://www.ece.wisc.edu/~nowak/GPSR.pdf.`

[37] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood estimation from incomplete data via the EM algorithm,” *J. R. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[38] M. A. T. Figueirado and R. D. Nowak, “An EM algorithm for wavelet-based image restoration,” *IEEE Trans. on Image Proc.*, vol. 12, no. 8, pp. 906–916, August 2003.

[39] J. M. Mendel, Maximum Likelihood Deconvolution, Springer, New York, 1990.

[40] H. L. Taylor, S. C. Banks, and J. F. McCoy, “Deconvolution with the ℓ^1-norm,” *Geophysics.*, vol. 44, no. 1, pp. 39–52, 1979.

[41] J. J. Kormylo and J. M. Mendel, “Maximum likelihood detection and estimation method for sparse deconvolution,” *IEEE Trans. Info. Theory*, vol. 28, no. 3, pp. 482–488, May 1982.

[42] M. Lavielle, “Bayesian deconvolution of Bernoulli-Gaussian processes,” *Signal Processing.*, vol. 33, No. 1, pp. 67–79, July 1993.

[43] I. Santamaria-Caballero, C. J. Pantaleon-Prieto, and A. Artes-Rodriguez, “Sparse deconvolution using adaptive mixed-Gaussian models,” *Signal Processing*, vol. 54, pp. 161–172, 1996.

[44] E. J. Candès and J. Romberg, “ℓ^1-magic: Recovery of sparse signals via convex programming”, 2005, Available at URL:`http://www.acm.caltech.edu/l1magic/downloads/l1-magic.pdf.`

[45] A. M. Djafari, “Bayesian source separation: beyond PCA and ICA,” in *Proceedings of ESANN’06*, pp. 313–322, April 2006.

[46] F. Labeau, J. C. Chiang, M. Kieffer, P. Duhamel, L. Vandendorpe and B. Macq, “Oversampled filter banks as error correcting codes: Theory and impulse noise cancellaion,” *IEEE Trans. on Signal Process.*, vol. 53, pp. 4619–4630, December 2005.

[47] E. Larsson and Y. Selen, “Linear Regression with a sparse parameter vector,” *IEEE Trans. on Signal Proc.*, vol. 55, pp. 451–460, 2007.

[48] A. M. Tomasi, “Estimating Gaussian mixture density with EM- a tutorial,” Available at URL:`http://www.cs.duke.edu/courses/spring04/cps196.1/handouts/EM/tomasiEM.pdf.`

[49] B. D. Anderson and J. B. Moore, Optimal filtering, Prentice Hall, 1979.

[50] A. Blake and A. Zisserman, Visual Reconstruction, MIT Press, 1987.