A Simple Mixture Model for Sparse Overcomplete ICA

Mike Davies & Nikolaos Mitianoudis
Queen Mary, University of London,
Mile End Road, London E1 4NS

SUBMITTED TO
IEE Proceedings-Vision, Image and Signal Processing
Special Issue on
NON-LINEAR AND NON-GAUSSIAN SIGNAL PROCESSING

Abstract

We explore the use of Mixture of Gaussians (MoGs) for noisy and overcomplete ICA when the source distributions are very sparse. The sparsity model can often be justified if an appropriate transform, such as the Modified Discrete Cosine Transform, is used. Given the sparsity assumption we are able to introduce a number of simplifying approximations to the observation density that avoids the exponential growth of mixture components. We further derive an efficient clustering algorithm whose complexity grows linearly with the number of sources and show that it is capable of performing reasonable separation.

I. Introduction

Independent Component Analysis (ICA) is a framework for separating out a mixture $x \in \Re^m$ of independent sources $s \in \Re^n$ with little or no prior information other than the nonGaussianity of the source distributions. If the number of sources is equal to the number of mixtures the problem is one of identifying the mixing matrix A and the source estimates are simply $\hat{s} = A^{-1}x$. One popular approach is to build a probability density model for the observed data with the associated independence constraints. Model estimation can then be done using simple learning algorithms based upon the Maximum Likelihood (ML) principle (e.g. [4]).

When extending this principle to the case where the number of sources, n, is greater than the number of sensors, m, and the inclusion of additive noise, neither the estimation of A, nor even the estimation of s given A are straightforward. One solution, introduced in [18], is to model the source distributions as Mixtures of Gaussians (MoGs). The resulting observation density is then also a MoG, even in the presence of additive (Gaussian) noise. The beauty of this model is that the Maximum Likelihood solution can be determined using the EM algorithm [9].

Unfortunately the number of mixtures in the observation density grows exponentially with the number of sources. Thus, even for a simple distribution model and a modest number of sources the algorithm will become intractable. Approximations using either variational learning [1] or MCMC methods [19] have previously been introduced to overcome this problem. Here we introduce an additional assumption that the source distributions are sparse in a similar manner to [22]. Sparse data representations have been popular for a long time in the coding community since they provide good coding efficiency. Using this assumption we approximate the observation density by a truncation of the mixture model. The accuracy of this is a function of source sparsity and allows us to define different levels of approximation. We concentrate on the crudest level of approximation where we obtain a simple clustering algorithm whose complexity is linear in the

number of sources. Furthermore, in this case, it is possible to replace part of the EM estimation step by a conditional maximization that has effectively integrated out the source variables in a similar manner to the Mixtures of Principal Component Analyzers (MPCA) algorithm of Tipping and Bishop [21]. The resulting algorithm falls into the category of Alternating Expectation Conditional Maximization (AECM) algorithms [17] and exhibits improved convergence.

Finally, we include a discussion on potential extensions of this work, and in particular, to deal with complex data which is important in real world audio signal separation [8].

The rest of the paper is set out as follows. In section II, we introduce the problem and describe some previous approaches. We then review, in section III, the MoG approach to overcomplete ICA, highlighting the complexity issues. In section IV, we discuss the ability to transform signals into sparse data representations. Section V then introduces our sparse MoG model and describes the resulting algorithm. We conclude with a comparison between the sparse MoG approximation and the full MoG model using 2-state source mixtures. These are also compared to Lewicki's overcomplete ICA algorithm based upon Laplacian source distributions.

II. Overcomplete ICA

Our model for noisy overcomplete ICA is as follows:

$$x_t = As_t + e_t \tag{1}$$

where A is an $m \times n$ matrix, with $n > m$, x_t is the observation vector, s_t is the *independent* source vector and e_t is Gaussian noise with covariance J. Furthermore, in this paper, we will assume that the source distributions have zero mean and are sparse (we will discuss this in more detail in section IV).

To obtain the ML estimates for A and J we must marginalise over the unknown source values:

$$P(x|A, J) = \int P(x|s, A, J)P(s)ds \tag{2}$$

As noted in [2] this integral is in general analytically intractable.

A number of ways forward have been proposed. Lewicki and Sejnowski [14] used the Laplace approximation for the integral which resulted in a gradient based update for learning A similar to the case of equal numbers of sources and sensors. The source distributions were fixed *a priori* and assumed to be Laplacian distributed. This assumption means that estimating s_t can be achieved

using linear programming [14] which was required at each iterate. The Laplace approximation was further examined by Bermond and Cardoso in [2].

Recently, Girolami [11] introduced an EM algorithm with auxiliary variational parameters to approximate the case of Laplacian source distributions. This provides a lower bound on $P(x|A,J)$ that is solvable using EM. In this framework an update for the noise covariance can also be estimated.

Hyvarinen [12] questioned whether we need to use the marginal distribution at all and proposed maximizing the joint $P(x,s|A,J)$ with respect to A and s. This will introduce some bias into the estimates, however if the sources are highly separable the bias should be small.

Finally, a very flexible structure was introduced by Moulines *et al.* [18] of using a MoG model for ICA. This was also extensively studied by Attias [1] under the name of Independent Factor Analysis (IFA). The MoG approach offers an exact solution to the overcomplete problem as well as providing a mechanism for learning the source distributions and noise covariance. However, as we shall see, this approach is not without its problems. We will discuss the MoG approach to ICA in detail in the next section.

III. The Mixture of Gaussians Model

One way to approximate a nonGaussian distribution is to use a mixture of Gaussians. We can then write the distribution for the ith source as a pth order MoG (we only consider zero mean MoGs, however this is not a general restriction of the MoG formulation):

$$P(s_t^{(i)}) = \sum_{k=1}^{p} \omega_{i,k} \mathcal{N}_{s^{(i)}}(0, \sigma_{i,k}^2), \text{ where } \sum_{k=1}^{p} \omega_{i,k} = 1 \tag{3}$$

For convenience we introduce an indexing variable $q(t) = [q_1(t), \ldots, q_n(t)]$ where each $q_k(t)$ can take a discrete value from 1 to p and represents the state of the mixture for the kth source at time t. The beauty of the MoG model is that, given $q(t)$, the likelihood for x_t is Gaussian:

$$\begin{aligned} P(x|A,J,q) &= \int \mathcal{N}_e(x - As, J) \left[\prod_{i=1}^{n} \mathcal{N}_{s^{(i)}}(0, \sigma_{i,q_i}^2) \right] ds \\ &= \mathcal{N}_x \left(0, J + \sum_{i=1}^{n} a_i a_i^T \sigma_{i,q_i}^2 \right) \end{aligned} \tag{4}$$

where a_i is the ith column of A.

Marginalising out the index variable $q(t)$ we get the following expression for the observation

E-step

$$\text{Evaluate } \tilde{P}_q = P(q|x, A, J, \sigma, \omega)$$
$$\text{Evaluate } \tilde{P}_s = P(s|x, q, A, J, \sigma, \omega)$$

M-step

$$(A, J) = \arg\max_{(A,J)} E\{\log P(x|s, q, A, J)|\tilde{P}_s, \tilde{P}_q\}$$
$$\sigma = \arg\max_{\sigma} E\{\log P(s|q, \omega, \sigma)|\tilde{P}_s, \tilde{P}_q\}$$
$$\omega = \arg\max_{\omega} E\{\log P(q|\omega)|\tilde{P}_q\}$$

TABLE I

AN EM ALGORITHM FOR MOG-BASED ICA

density function:

$$p(x|A, J, \sigma, \omega) = \begin{array}{l} \sum_{q_1=1}^{p} \dots \sum_{q_n=1}^{p} \omega_{1,q_1} \dots \omega_{n,q_n} \\ \times \mathcal{N}_x \left(0, J + \sum_{k=1}^{n} a_k a_k^T \sigma_{k,q_k}^2\right) \end{array} \quad (5)$$

It is clear that the observation density is also a MoG, albeit with a constrained set of parameters.

Using this structure it is possible to derive an EM algorithm to iteratively maximize $P(x|A, J, \sigma, \omega)$ [18], [1]. The key steps are indicated in table I.

The expectations are all taken with respect to the densities evaluated in the E-step, as indicated. Furthermore each of the optimizations in the M-step is analytically tractable.

Once the system parameters have been estimated the sources can be recovered using either MAP estimates or MMSE estimates conditional on A etc. In the simulations below we will always use MMSE estimates. For full details of the EM procedure and the source recovery the reader is referred to [1].

A. Algorithmic Complexity

Unfortunately from (5) we see that the number of Gaussians is p^n, so even for a small number of sources and a minimal mixture order this approach becomes impractical. (e.g. $n = 10, p = 2 \rightarrow p^n = 1024$).

One way to reduce complexity is to approximate the Likelihood by another function that is less complex to solve. This is the idea behind the variational learning proposed by Attias [1], where the conditional density for the hidden variables is replaced by a simpler function with a factorial form. Another approach due to Olshausen and Millman [19] is to use a Monte Carlo

learning mechanism. Below we will adopt yet another approach: one of writing the observed density model (5) as a summation of Gaussians with decaying weights. Truncating this sum at any point gives us a pseudo-likelihood that approximates the true one. We will see that this is particularly useful if we can adopt a sparse source representation.

IV. Sparse Signal Representations

Most previous work on overcomplete ICA assumes the source distributions are sparse in some sense (with the exception of discrete sources, e.g. [20]). By sparsity we mean that most of the source data coefficients do not differ significantly from zero. Common statistical models for sparsity are minimum L_0 or L_1 norms [6], [15], Laplacian distributions [13], [14] (equivalent to minimum L_1) or 2-state Gaussian mixture models [19], [7]. For example Lee et al. [13] used the fact that the time sample distribution of speech is well approximated by a Laplacian distribution in their ICA algorithm.

Sparsity is good in ICA for two reasons. First, as noted by Cardoso in [5], the statistical accuracy with which the mixing matrix A can be estimated is a function of how non-Gaussian the source distributions are. Thus, roughly speaking, the sparser the sources are the less data is needed to estimate A. Secondly the quality of the source estimates, given A, is also better for sparser sources [22].

Although ICA performance is better for sparse data, this in itself does not help us if the source data is not sparse. Fortunately many natural signals are sparse when represented in the appropriate manner. Indeed the coding community has researched many linear transforms that make audio, image and video data sparse, such as the DCT, the Fourier Transform, the wavelet transform and their derivatives [15]. From a coding perspective sparse representations provide better compression using scalar quantization [10].

From a source separation perspective it is straightforward to incorporate such transforms into the ICA model. Following [22], if we transform the signals to a sparse representation using the linear transform $c^{(i)} = Ms^{(i)}$ then the ICA model in equation (1) becomes the following equivalent ICA problem in the transform domain:

$$\tilde{x}_k = Ac_k + \epsilon_k \tag{6}$$

where $\tilde{x}^{(i)} = Mx^{(i)}$ are the observation signals in the transform domain and $\epsilon^{(i)} = Me^{(i)}$ are the transformed noise signals which are also still Gaussian due to the linearity of M.

A popular coding transform for audio that we will use in the simulations below is the Modified Discrete Cosine Transform (MDCT) [15]. This is an orthogonal, real-valued lapped transform that is well suited for coding tonal-like signals such as music and is used in most current audio coding protocols [3].

To see the effect of the MDCT on the audio signals we can look at the distributions of an audio sample in both the time domain and the MDCT domain. While the time domain samples are modestly sparse (i.e. well approximated as Laplacian) the signal is extremely sparse in the MDCT domain. Figure 1 shows the histograms for both the time domain and the MDCT domain. Below this are plots of the Laplacian distribution (fitted to the time domain) and the 2-state MoG model (fitted to the MDCT data) which we will use in the next section. Note, since the MDCT is an orthogonal transform, both these distributions have the same variance. The difference in sparsity is quite dramatic.

V. A sparse MoG approximation

Assuming that we can find some sparse representation for our source signals, we can make some strong simplifying approximations to the MoG model. We defined a distribution as sparse if most of the probability mass lies close to zero. In terms of a MoG model we can approximate a sparse distribution with a dominant 'off' state associated with a Gaussian with a very small variance. Furthermore since this Gaussian will account for most of the probability mass, in terms of equation (3) we will have: $\omega_{i,1} \gg \omega_{i,k}$ when $k \neq 1$ and $\sigma_{i,1}^2 \approx 0$

Looking at equation (5), we see that in the observation mixture each Gaussian has weights that are the product of n individual source mixture weights. Indeed, if we assume $P(\text{'on'}) \sim O(\epsilon)$ the observation likelihood has mixture weights of order $O(1)$, $O(\epsilon)$, $O(\epsilon^2), \ldots$. These are in turn associated with none, one, two, etc. sources being in the 'on' state at any particular time. Thus when $P(s)$ is sparse (and ϵ is small) the contribution to the observation likelihood of most of the Gaussian elements will be very small. We therefore propose truncating number of Gaussians and only retaining those with a reasonable size weights. Learning can be achieved using the EM algorithm given above but with a pruned set of admissible index variables. Indeed, it should be noted that the resulting model is still a completely legitimate component model. However, we have in part replaced the *independence* assumption by one of *mutual exclusivity*. In other words only a fraction of the sources are allowed to be on simultaneously.

Below we will only consider the simplest form of this sparse MoG approximation. One where

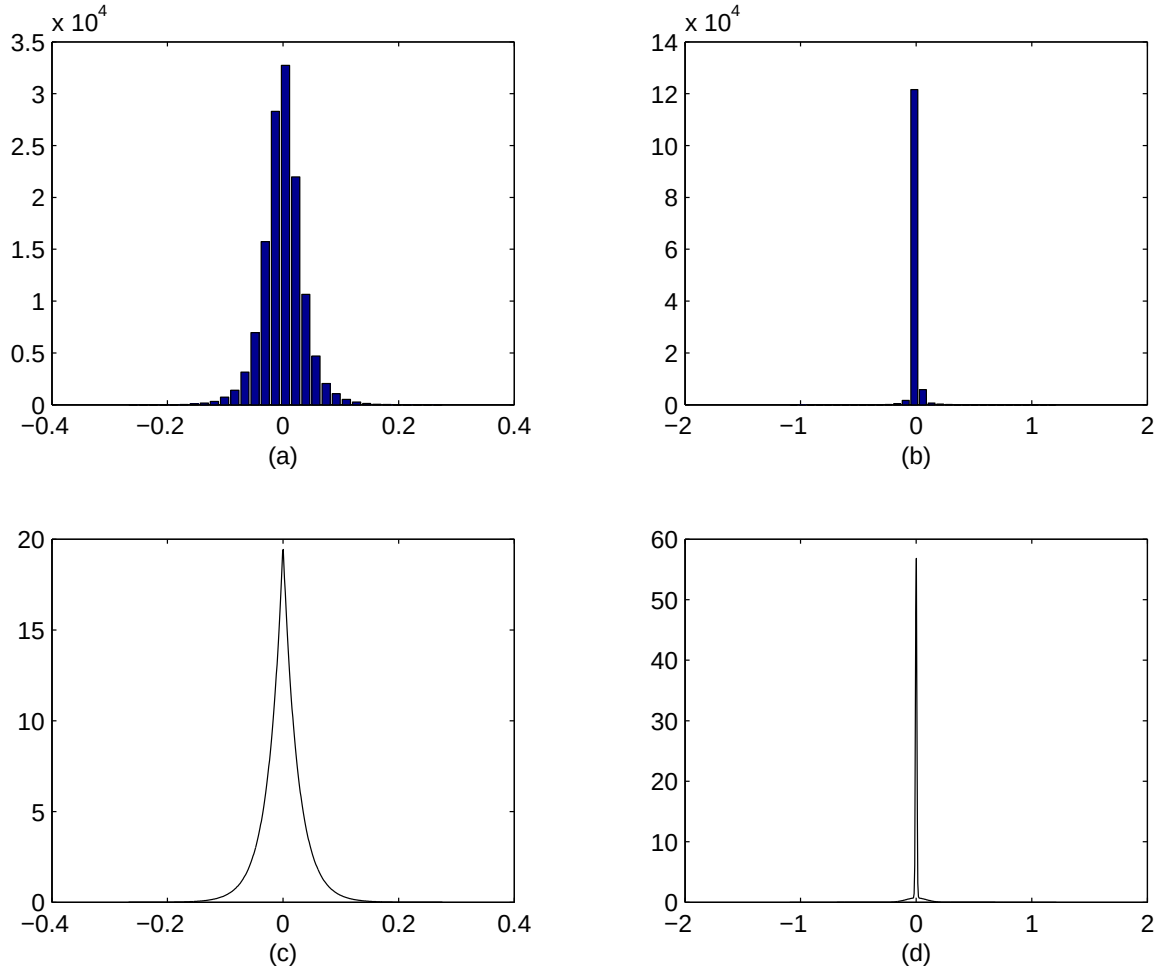

Fig. 1. Distributions for an audio sample: (a) histogram of time domain samples; (b) histogram of the MDCT domain samples; (c) a Laplacian distribution fitted to the time domain data; (d) a 2-state Gaussian mixture fitted to the MDCT data.

the sources are modelled by 2-state MoGs and only one source is allowed to be active at any given time.

A. A 2-state sparse source distribution

We now concentrate on the simplest possible sparse MoG model. That is a 2-state mixture. The first state is effectively the 'off' state and is assumed to have a variance well below the observation noise and we therefore set $\sigma_{i,1} = 0$. The second state is the 'on' state and represents large coefficients. As most of the probability mass is in the 'off' state we have $P_i(\text{'on'}) = \omega_{i,2} \sim O(\epsilon)$

While such a source model might be considered over simplistic it has sufficient complexity to

capture the salient features for very sparse data. Indeed 2-state mixture models have been used very successfully by Olshausen and Millman [19] to estimate overcomplete bases for images and for modelling sparsity in wavelet decompositions [7].

B. 2-state 'winner-takes-all' approximation

Applying the 2-state source model to the ICA problem means that the full MoG observation model will have 2^n Gaussian elements. The crudest possible truncation of this mixture is to only retain Gaussian elements with weights of $O(\epsilon)$. That is to only allow one source to be 'on' at any given time. This 'winner-takes-all' (WTA) approximation gives us the following pseudo-likelihood which is a MoG with only $n+1$ Gaussians:

$$p(x_t|A,J,\eta) \approx \eta_0 \mathcal{N}_x(0,J) + \sum_{k=1}^{n} \eta_k \mathcal{N}_x\left(0, J + a_k a_k^T\right) \tag{7}$$

where η_k are the weights for the observation mixtures.

Recall that we have set the 'off' variances to zero (since they were assumed much smaller than $|J|$). Furthermore each 'on' state for a source is only associated with one Gaussian. As such we no longer need to calculate the 'on' variance parameters as these become absorbed into A by the standard ICA scale ambiguity. Note this is not quite the same as using a fixed prior distribution (as in the Lewicki and Sejnowski algorithm [14] since each source still has a single nuisance parameter governing the probability of the 'on' state, which in turn gives us some measure of sparsity.

The EM algorithm for the mixture model in equation (7) is essentially a simple clustering algorithm with a complexity that grows linearly with respect to the number of sources. It can be implemented in exactly the same manner as the full MoG model (table I) with a restricted set of allowable indices. However, given the simplicity of equation (7), we are also able to make a further algorithmic improvement that speeds up convergence using an extension of EM called Alternating Expectation Conditional Maximization (AECM) [16].

It is well known that the convergence rate for EM depends on the proportion of missing information in the prescribed EM framework [16]. In the case in point we have a high proportion of missing data: both the index variables $q(t)$ and the source signals $s_t^{(i)}$. This can lead to painfully slow convergence in a similar manner to ordinary factor analysis [16].

One solution is to explicitly integrate out missing data wherever possible. Here, because

1st E-step

Evaluate $\tilde{P}_q = P(q|x, A, J, \eta)$

1st M-step

$$A = \arg\max_A E\{\log P(x|q, A, J)|\tilde{P}_q\}$$
$$\eta = \arg\max_\eta E\{\log P(q|\eta)|\tilde{P}_q\}$$

2nd E-step

Evaluate $\tilde{P}_s = P(s|x, q, A, J, \eta)$

2nd M-step

$$J = \arg\max_J E\{\log P(x|s, q, A, J)|\tilde{P}_s, \tilde{P}_q\}$$

TABLE II

AECM ALGORITHM FOR MAXIMIZING EQUATION (7)

the columns of A only occur within a single mixture, we are able to maximize the conditional complete data likelihood $P(x|q, A, J)$ directly, without augmenting the problem with the source variables s_t. However to estimate J the source missing data is still required. This is an ACEM algorithm [17] which shares the monotonic convergence properties of EM but tends to have faster convergence (see [17], [16] for details). Table II describes our proposed AECM algorithm.

The key difference between this and the previous algorithm is in optimizing A. This can be performed directly by eigenvalue decomposition as follows. The Expectation can be written as:

$$\begin{aligned} E\{\log P(x|q, A, J)|\tilde{P}_q = k\} = \quad & -\tfrac{1}{2}\textstyle\sum_t \tilde{P}_q(t,k) x_t^T (J + a_k a_k^T)^{-1} x_t \\ & -\tfrac{1}{2}\textstyle\sum_t \tilde{P}_q(t,k) \log|J + a_k a_k^T| \\ & + \text{const.} \end{aligned} \tag{8}$$

where by $\tilde{P}_q(t, k)$ we mean the probability that $q_t = k$.

If J is $\propto I$ then the a_k that maximizes (8) is the eigenvector associated with the largest eigenvalue of the covariance matrix:

$$E\{x_t x_t^T|\tilde{P}_q = k\} = \frac{\sum_t \tilde{P}_q(t,k) x_t x_t^T}{\sum_t \tilde{P}_q(t,k)} \tag{9}$$

This property is shown and used by Tipping and Bishop in their EM algorithm for a mixture of Principal Component Analyzers (MPCA) [21].

Given that at any stage there is always an estimate for J we can transform x to a new

observation space $\tilde{x} = L^{-1}x$ where L is the Cholesky decomposition of $J = LL^T$, thus making the additive noise covariance for $\tilde{x}$ isotropic.

Although this AECM algorithm is slightly more complex per iterate than the direct EM algorithm we have found it to be considerably faster and more robust in practice.

VI. Simulations

Speech is a good example of signals that can be sparsely represented in the MDCT domain. To demonstrate our proposed algorithm we took 3 speech signals of 2 second duration, sampled at 16kHz and mixed them into two mixtures:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.1 & 0.3 & 0.6 \\ 0.3 & 0.5 & 0.2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \tag{10}$$

The original sources along with the observation mixtures are plotted in figure 2. To transform the signals into a sparse representation the MDCT transform was used with a frame size of 1024 samples (64 milliseconds). For comparison figure 3 shows the MDCT coefficients for the same signals.

The increased sparsity can also be seen in the scatter plots of the data for both the time domain samples and the MDCT domain coefficients. These are shown in figure 4. The sparsity in the scatter plot for the MDCT domain makes the mixing directions clearly identifiable (A can be easily estimated by eye).

Given these sparse mixtures we then ran the algorithms for the full MoG model (equation 5) and the sparse WTA approximation (equation 7) on the data. In both cases the sources were modelled by a 2-state MoG with the variance of the 'off' state set to zero and the variance of the 'on' state set to one. Both algorithms were run until they had converged. The resulting mixture models are displayed graphically in figure 5 and figure 6 respectively. Below each Gaussian is the value of its associated weighting within the MoG.

Although the source model has minimal complexity it still requires twice as many Gaussians to model the data. Yet it is clear from the individual weights for the full model that the last four Gaussians contribute very little to the MoG density (only accounting for 1% of the total probability mass). The dominant four Gaussians are associated with either all sources 'off' (first Gaussian) or only one source in the 'on' state. Thus the dominant Gaussians are fully captured

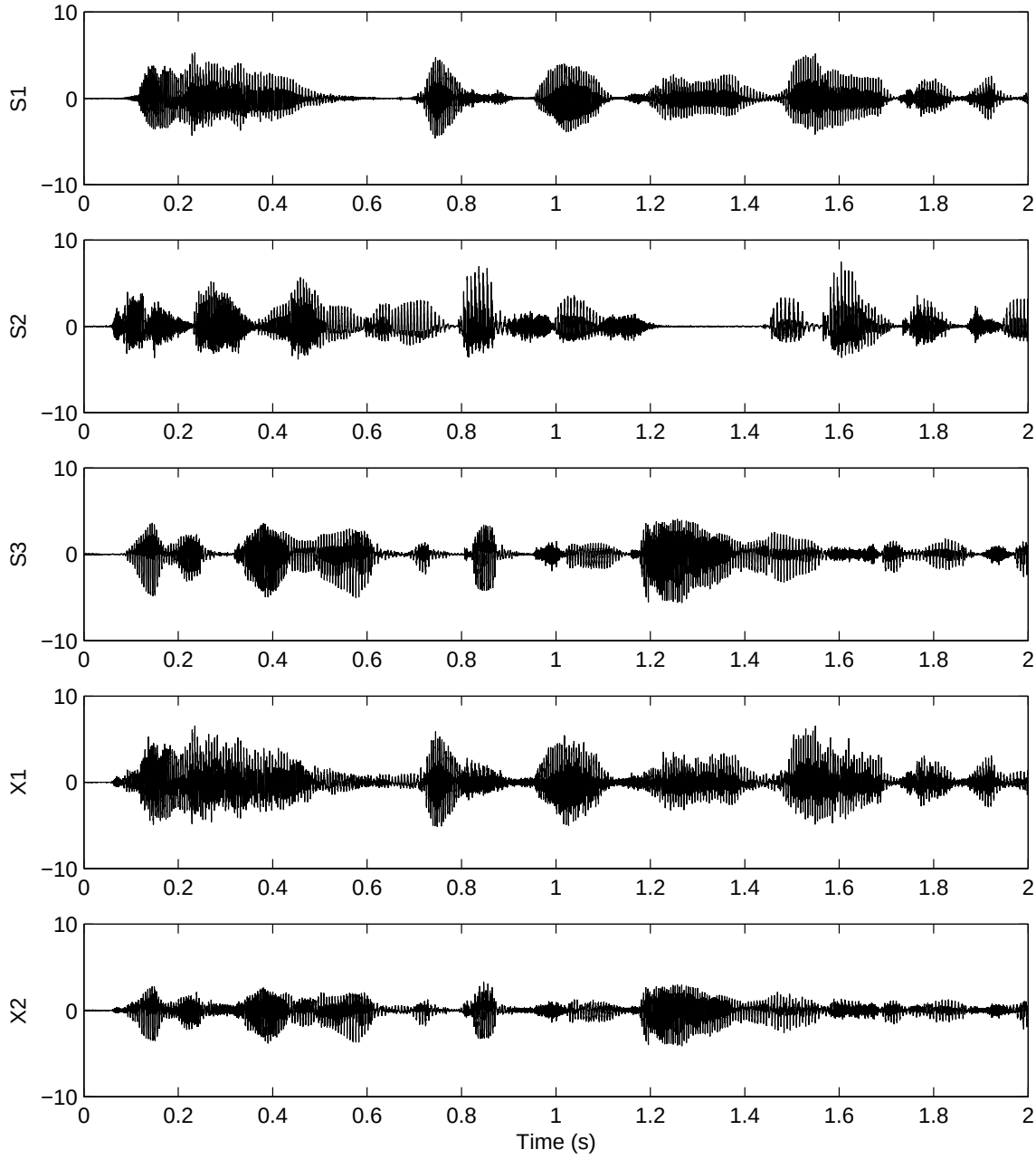

Fig. 2. Time sample plots for the 3 speech signals and the two mixture observations.

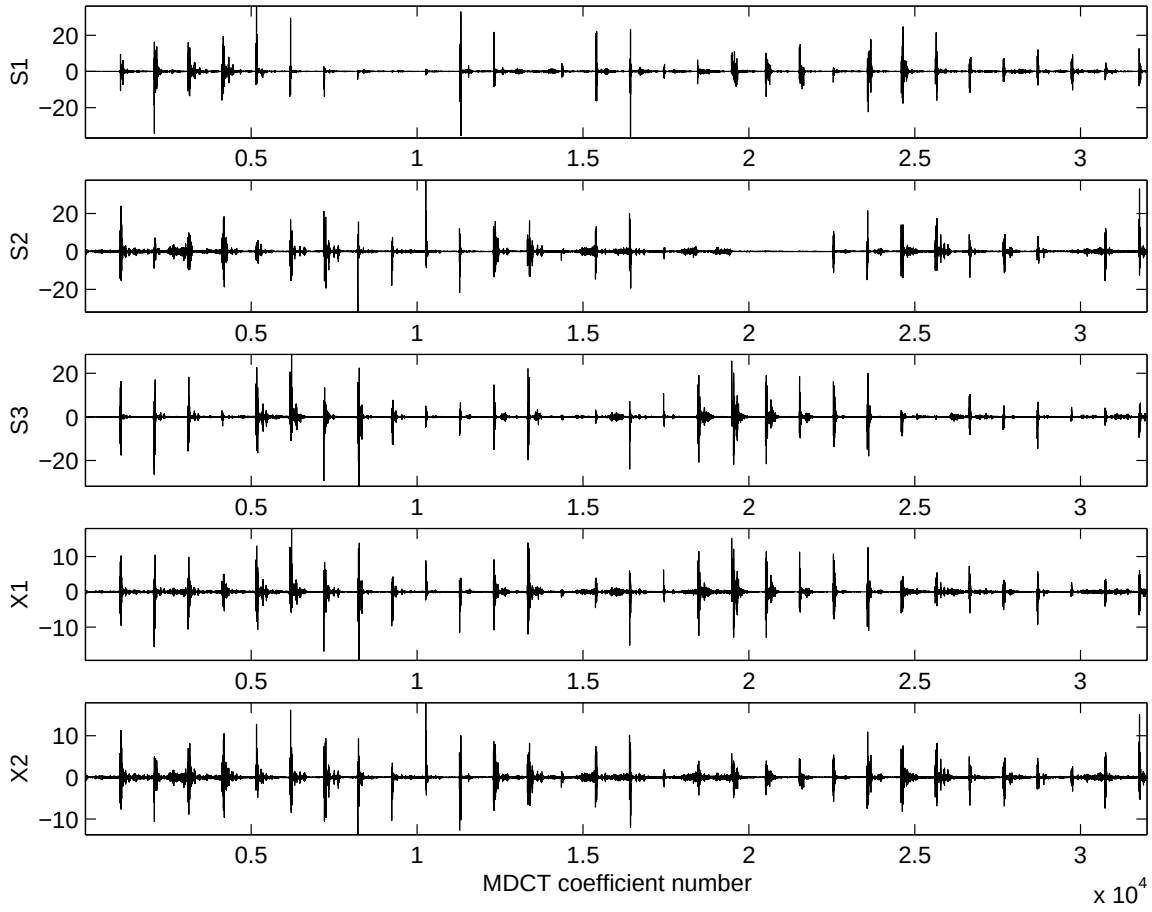

Fig. 3. The MDCT coefficients for the 3 speech signals and the two observation mixtures shown in figure 2.

by the WTA approximation. Indeed we can see that the WTA algorithm has learnt a very similar set of Gaussians and weights. Note that the WTA approximation tends to have a slightly high noise estimate due to the lack of the other four Gaussians. With both the full MoG model and the WTA approximation the correct directions of the mixing matrix are identified.

To reconstruct the source estimates we used MMSE estimators conditional on the learnt parameters. For the WTA approximation there were two options. We could either estimate the sources for the constrained model in equation (7) or produce a full reconstruction by substituting the learnt parameters, A, J, ω_{ij} into the full mixture model with the obvious additional computation (the cost of a single full MoG iteration). We applied both methods to the speech data and compared the sources to the full MoG model as well as the Lewicki algorithm [14]. Plots of the WTA source estimates and their errors are shown in figure 7. The resulting Signal to Noise

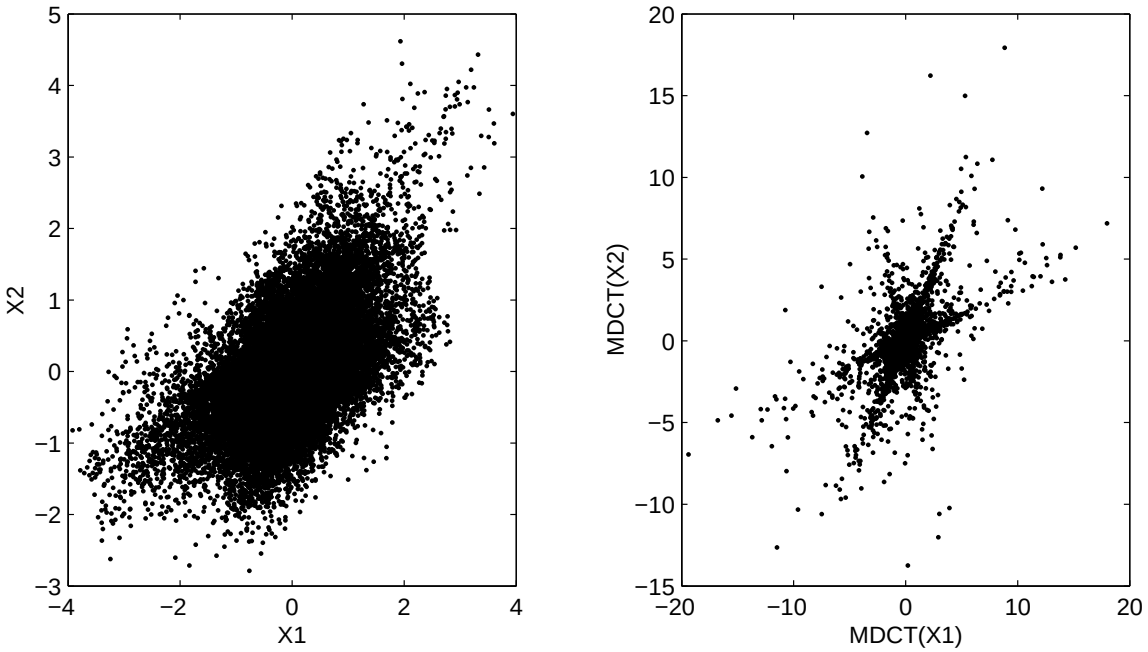

Fig. 4. Scatter plots for the time domain mixtures (left) and the MDCT coefficients of the same mixtures (right).

Ratios for all methods are given in table III below.

SNR(dB)	$s^{(1)}$	$s^{(2)}$	$s^{(3)}$
Full MoG	5.0	5.9	7.4
WTA	5.0	4.4	6.8
WTA (full recon.)	6.2	6.0	9.3
Lewicki	6.9	9.0	11.6

TABLE III

SNR (dB) MEASUREMENTS FOR THE THREE SOURCE ESTIMATES USING THE FULL MOG MODEL; THE WTA; THE WTA WITH A FULL RECONSTRUCTION; AND THE LEWICKI ALGORITHM.

In all cases the separated speech signals were comprehensible, though not perfectly recovered. From the SNR results it is clear that the WTA algorithm performs slightly worse than the full MoG model (1 dB on average), though when the WTA is used with a full reconstruction the the results are as good.

Interestingly none of the MoG solutions here performed as well as the Lewicki algorithm. This

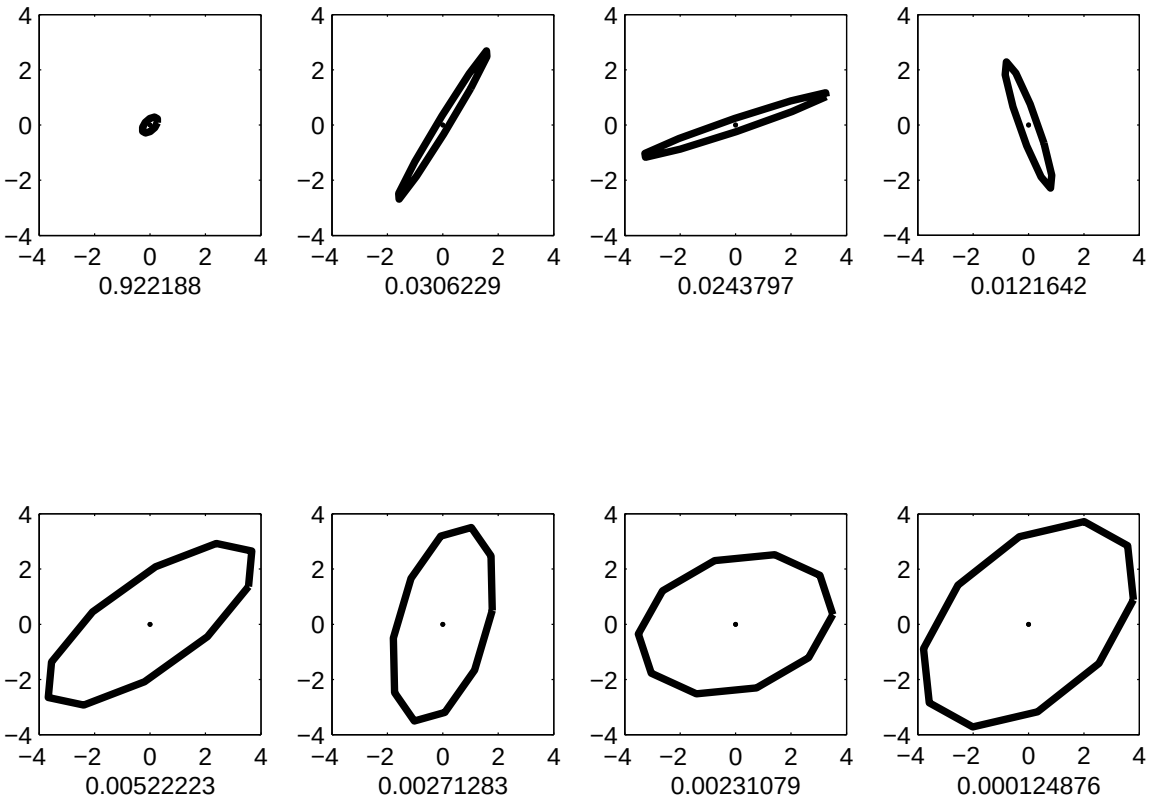

Fig. 5. The plots show the eight Gaussians (one standard deviation contour) resulting from the full model with the weights for each Gaussian are listed below.

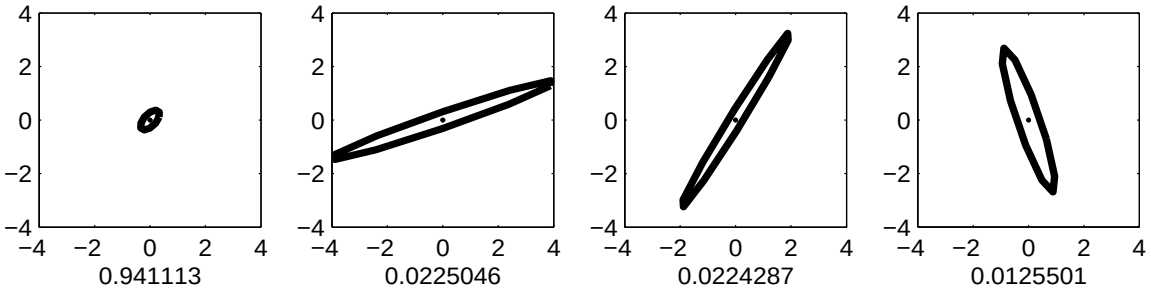

Fig. 6. The plots show the four Gaussians (one standard deviation contour) resulting from the winner-takes-all approximation with the associated weights for each listed below.

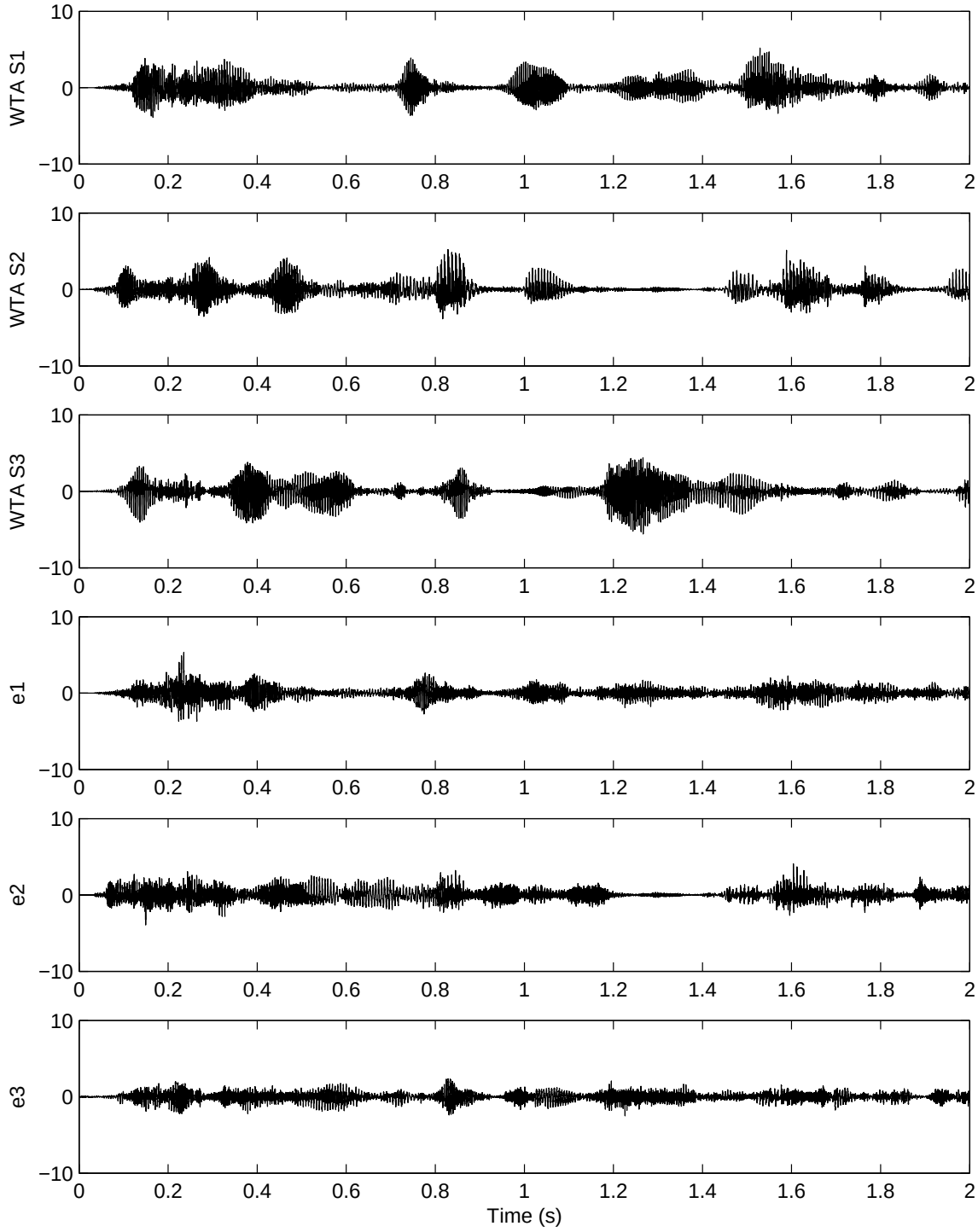

Fig. 7. Time sample plots for the 3 speech signal estimates using the winner-takes-all approximation (top three) and their estimation errors (bottom three).

is because the MoG formulation assumes the presence of noise and because the 2-state source priors are quite severe. This results in small MDCT coefficients being modelled as noise and hence suppressed.

To validate this assertion we repeated the experiment with the addition of a small amount of additive Gaussian noise with a variance of 0.25. The SNR results for the noisy source separation are shown in table IV and we see that now the MoG models have the highest SNR. Notice that now there is very little difference between any of the MoG based solutions. They all also estimated the noise covariance to be about $[0.3, 0; 0, 0.3]$: a slight over-estimate. This implies that the WTA approximation is particularly suitable for noisy overcomplete mixtures. The Lewicki algorithm, which does not model the noise, performs significantly worse in the noisy environment (1-4 dB worse than the MoG solutions).

SNR(dB)	$s^{(1)}$	$s^{(2)}$	$s^{(3)}$
Full MoG	4.9	5.3	7.6
WTA	4.5	5.0	6.7
WTA (full recon.)	4.9	5.2	7.5
Lewicki	1.6	4.2	4.6

TABLE IV

SNR (dB) measurements for the three source estimates with the addition of Gaussian noise (variance=0.25) using the full MoG model; the WTA; the WTA with a full reconstruction; and the Lewicki algorithm.

Finally we compare the convergence rates for the EM algorithm with our proposed accelerated AECM algorithm (table II). Applying both algorithms to the WTA approximation for the noiseless speech data above, figure 8 shows the rate of convergence for the mixing coefficients. In both cases the algorithms converged to the same solution (ignoring a sign ambiguity).

As noted in section V-B the EM algorithm has a large quantity of missing data thus we can expect its convergence to be slow (indeed Bermond and Cardoso [2] have shown that the EM algorithm tends to a small gradient step in the low noise limit). This is confirmed in practice since even after 200 iterations the EM algorithm has not fully converged. In contrast to this the AECM algorithm converges extremely rapidly (here in about 10 iterations).

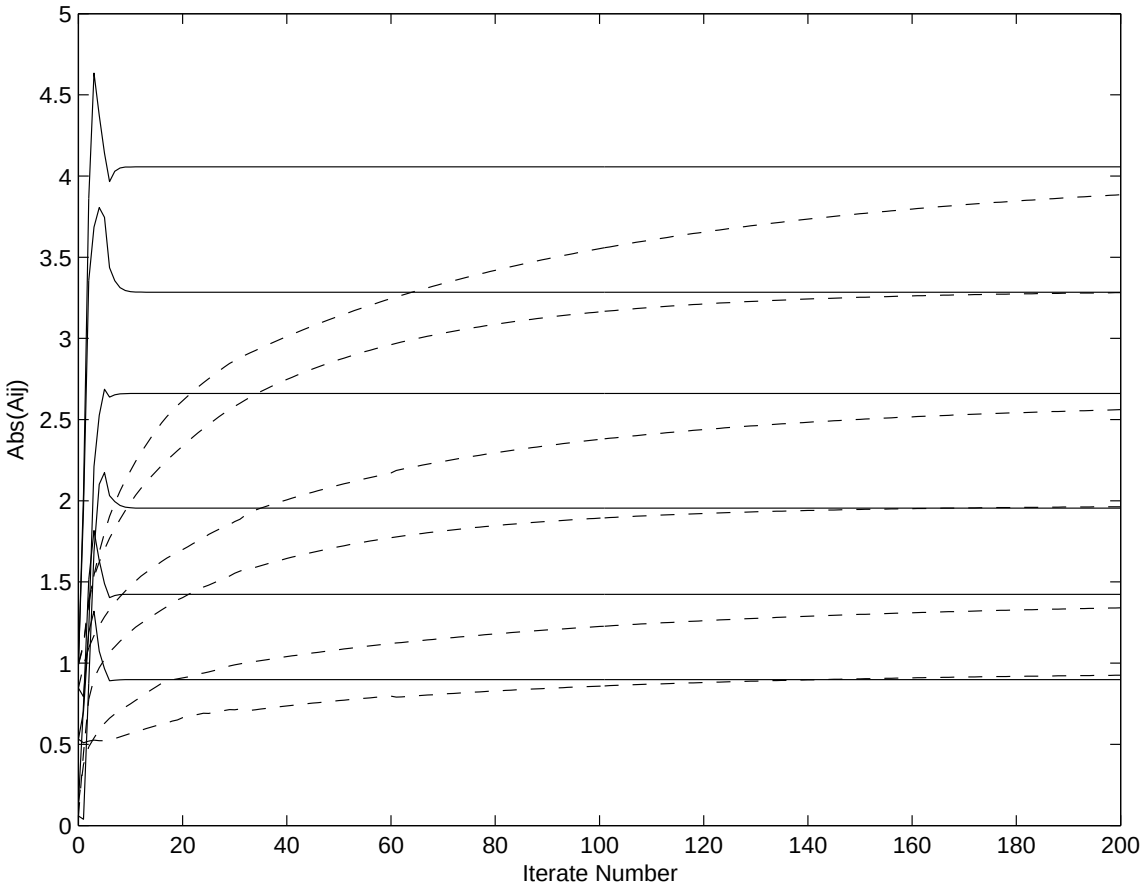

Fig. 8. Estimates for the absolute value of the mixing coefficients against iteration number, using the full EM algorithm (dashed) and the ACEM (solid).

VII. Discussion

We have presented an approximation scheme for dealing with the exponential growth in complexity that occurs when formulating overcomplete ICA within a MoG framework. This is appropriate when the source distributions are sparse and most of the MoG states have a low probability of occurrence. We have concentrated on the simplest possible 2-state MoG source representation that results in an algorithm with linear complexity that converges faster than a previously proposed gradient-based methods. It is surprising that such a simple source model can be successfully used to separate out overcomplete mixtures.

In future, it would be interesting to adapt this framework to deal with the delays and convolutions present in real world audio mixtures. In such circumstances it is preferable to transform into the frequency domain [8] since, firstly, convolutive mixing becomes complex multiplication.

Secondly, sources tend to be significantly sparser than in the time domain. In the framework presented here it should be straight forward to replace the mixtures of real valued Gaussians for mixtures of complex Gaussians. The update algorithms (of both algorithms) will remain the same with transpose replaced by conjugate transpose. We also note that such complex source priors will be naturally phase invariant, which is generally desirable [8].

Another possible avenue of research would be to incorporate order selection within this framework, either for the individual source MoGs or more interestingly to identify varying numbers of sources.

Acknowledgments

The authors would particularly like to thank Laurent Daudet for providing the Matlab routines for the MDCT and the reviewers for their constructive comments.

References

[1] H. Attias, 1998 Independent Factor Analysis. *Neural Comp.*, 11, 803-851.

[2] O. Bermond and J-F. Cardoso 2000 Approximate Likelihood for noisy mixtures, *Proc. ICA '99.*

[3] K. Brandenburg 1999 MP3 and AAC explained, Proc. AES 17th Int. conf. High Quality Audio Coding (Florence, September 1999.

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

[5] J.-F. Cardoso, 1998 Blind signal separation: statistical principles, *Proceedings of the IEEE*, 9(10), 2009-2025.

[6] S. Chen and D.L. Donoho, 1999 Atomic decomposition by basis pursuit, *SIAM J. Sci. Computation, Vol. 20, No 1, pp 33-61, 1999.*

[7] M.S.Crouse, R.D.Nowak,and R.G. Baraniuk, 1998 Wavelet-Based Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing (Special Issue on Wavelets and Filter-banks)*, April 1998, 886-902.

[8] M.E. Davies 2002 Audio Source Separation. Chapter in *Mathematics in Signal Processing V*, Oxford University Press.

[9] A.P. Dempster, N.M. Laird and D. B. Rubin, 1977 Maximum Likelihood from incomplete data via the EM algorithm *J. Roy. Stat. Society* No. 1 pp 1-21.

[10] A. Gersho and R. M. Gray, 1992 *Vector Quantization and Signal Compression*, Kluwer Academic Publishers.

[11] M. Girolami, 2002 A Variational Method for Learning Sparse and Overcomplete Representations. *Neural Computation*, 13(11), pp 2517 - 2532.

[12] A. Hyvarinen, 1998 Independent Component Analysis in the presence of noise by maximizing the joint likelihood. *Neurocomputing.*

[13] T.W. Lee, M. Lewicki, M. Girolami and T. Sejnowski, 1999 Blind Source Separation of more sources than mixtures using overcomplete representations, *IEEE Signal Processing Letters*, 6, pp 87-90.

[14] M.S. Lewicki and T.J. Sejnowski, 2000 Learning overcomplete representations. *Neural Computation*, 12:337-365.

[15] S. Mallat, 1998 *A wavelet tour of signal processing*, Academic Press.

[16] G.J. McLachlin and T. Krishnan, 1997 *The EM Algorithm and Extensions*, Wiley Series in Probability and Statistics.

[17] X.L. Meng and D. van Dyk, 1997 The EM algorithm - an old folk song sung to fast new tune. *Journal of the Royal Statistical Society*, Ser. B, 59, 511-567.

[18] E. Moulines, J.-F. Cardoso, E. Gassiat, 1997 Maximum Likelihood for blind signal separation and deconvolution of noisy signals using mixture models. *ICASSP-97*.

[19] B. A. Olshausen and K.J. Millman, 2000 Learning sparse codes with a mixture-of-Gaussians prior. *in Advances in Neural Information Processing Systems, 12*, MIT Press, pp. 841-847.

[20] A. Taleb and C. Jutten 1999, On underdetermined source separation. In *Proc. 24th Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3 pp. 1445-1448.

[21] M.E. Tipping and C. M. Bishop 1999 Mixtures of probabilistic principal component analyzers. *Neural Comp.* 11(2), 443-482.

[22] M. Zibulevsky and B. A. Pearlmutter 2001, Blind separation of sources with sparse representations in a given signal dictionary, *Neural Computation*, vol. 13(4) pp. 863-882.