EFFICIENT BLIND SEPARATION OF CONVOLVED SOUND MIXTURES

Paris Smaragdis

Machine Listening Group
MIT Media Lab, Rm. E15-401C
20 Ames St., Cambridge, MA 02139
paris@media.mit.edu

ABSTRACT

In this paper we present an extension to recent approaches to Blind Source Separation. Bell and Sejnowski (1996) proposed a robust algorithm for separating instantaneous mixtures. Extensions were proposed by Torkkola (1996) and Lee *et al.* (1997) for separating convolved mixtures but the computational overhead and the convergence behavior of these algorithms were not ideal. A frequency domain extension is presented which improves the stability and the performance of these algorithms.

1. Introduction

Blind Source Separation is the problem of recovering independent sources given only mixes of these. This is a problem that occurs in a variety of disciplines ranging from telecommunications applications to artificial intelligence. Given the high profile of this problem it has received a generous amount of attention. Most implementations so far have been slow, demanding (in terms of accuracy) or not always successful, but thanks to recent advances in neural network research robust and efficient algorithms are starting to appear.

1.1. Instantaneous Mixture Separation

Bell and Sejnowski (1996) were the first to propose a robust algorithm for separating instantaneous mixtures which was later refined by Amari *et al.* (1996). Given N independent sources in the form of a vector $\mathbf{s}^T(t) = \left[s_1(t) \ \ldots \ s_N(t)\right]$, where t denotes time, we assume a mixing process defined as (see also Figure 1a):

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) \tag{1}$$

where $\mathbf{A} \in \Re^{N \times N}$ is an unknown matrix referred to as the mixing matrix. The vector $\mathbf{x}$ will be a linear mix of the original sources $\mathbf{s}$. This process is an idealized approximation of recording N sound sources in an anechoic room with N microphones[1]. The goal is to recover the $\mathbf{s}$ vectors, given only the $\mathbf{x}$ vectors. The way to do this is to invert the mixing process by implementing an unmixing equation which is (Figure 1b):

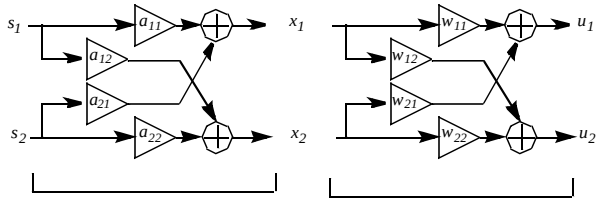

Figure 1: A 2 by 2 instantaneous mixing/unmixing process

$$\mathbf{u}(t) = \mathbf{W} \cdot \mathbf{x}(t) \tag{2}$$

and find a $\mathbf{W} = \mathbf{A}^{-1}$ so that $\mathbf{u}(t) = \mathbf{s}(t)$.

Finding this matrix $\mathbf{W}$ is a challenging task since we are not provided with any information about the $\mathbf{A}$ matrix nor the $\mathbf{s}$ vectors (hence the term blind separation). Theoretically the inverse of the mixing matrix can be estimated but with the columns permuted and scaled arbitrarily. This is not a problem for audio applications since the outputs can be scaled by post-processing and the output indexes are not of any major significance.

The approach suggested by Bell and Sejnowski (1996) is to implement the unmixing equation (2), and optimize with respect to the unmixing matrix, so as to minimize the statistical independence between the outputs $u_1(t), u_2(t), \ldots, u_N(t)$. It was shown that for Gaussian inputs this can happen by passing the outputs of the system through a sigmoid and maximizing the expected value of $\ln|\mathbf{J}|$, where $\mathbf{J}$ is the Jacobian of this transformation. Derivation yields the following update rule for the unmixing matrix:

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + 2 \cdot \mathbf{y}(t) \cdot \mathbf{x}(t)^T \tag{3}$$

where $\mathbf{y}(t) = \tanh\mathbf{u}(t)$. Alternative ways to derive similar learning rules are to use the Kullback-Leibler distance between the outputs (Amari (1996)) or maximum likelihood estimation (McKay (1996)).

[1] In this case we make the assumptions that sound transfers through media instantaneously and that all microphones are identical.

This update rule has also been refined by Amari (1996) who proposed a modification that performs natural gradient descent[1]. This algorithm exhibits very good performance since it is able to invert mixing matrices with a condition number in excess of 4000 (which corresponds to an amplitude difference in the inputs of roughly 72 dB). It has also been successfully used to separate up to 10 mixed sounds.

1.2. Convolved Mixture Separation

The assumptions that were set in order to solve the instantaneous mixture problem are not realistic at all. In real recordings sounds are delayed before they reach the microphones, they are convolved with room responses, and they are altered by the microphone characteristics. The model developed above is not adequate to separate sounds mixed this way.

Robust solutions to this problem were introduced by Torkkola (1996), who assumed the mixing process to be:

$$x_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{K} s_j(t-k) a_{ij}(k) \tag{4}$$

where N is the number of the sources s_i and a_{ij} are the K length mixing filters (which describe the delays and the microphone and room responses).

This problem is invertible by the equation:

$$u_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{M} x_j(t-k) h_{ij}(k) \tag{5}$$

where h_{ij} are the unmixing filters we need to estimate and $M > K$ is their length[2]. The derivation used here is the similar to Bell and Sejnowski (1996) (see Torkkola (1996)), and provides the following update rules for the 2 by 2 case:

$$\Delta h_{11}(0) \propto \frac{h_{22}(0)}{D} - 2 \cdot y_1(t) \cdot x_1(t) \tag{6}$$

$$\Delta h_{12}(0) \propto \frac{-h_{21}(0)}{D} - 2 \cdot y_1(t) \cdot x_2(t) \tag{7}$$

$$\Delta h_{21}(0) \propto \frac{-h_{12}(0)}{D} - 2 \cdot y_2(t) \cdot x_1(t) \tag{8}$$

$$\Delta h_{22}(0) \propto \frac{h_{22}(0)}{D} - 2 \cdot y_2(t) \cdot x_2(t) \tag{9}$$

$$\Delta h_{ij}(k) \propto -2 \cdot y_i(t) \cdot x_j(t-k) \tag{10}$$

where $y_i(t) = \tanh u_i(t)$ and $D = h_{11}(0) \cdot h_{22}(0) - h_{12}(0) \cdot h_{21}(0)$.

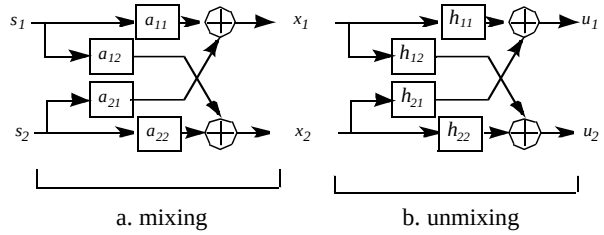

Figure 2: A 2 by 2 convolved mixing/unmixing process.

Unfortunately this algorithm exhibits very strong local minima which interfere with the adaptation. Due to the cost function there is optimization credit for decorrelating contiguous time samples from the same output as well as samples from different outputs. Since the length of the separating filters is most likely larger than the number of sources, the algorithm is quickly stuck at a position where the outputs are whitened and hardly separated.

Due to this property of this approach Torkkola (1996) proposed another unmixing procedure, defined as:

$$u_i(t) = \sum_{k=0}^{M} x_i(t) \cdot h_{ii}(t-k) + \sum_{j=1, j \neq i}^{N} \sum_{k=0}^{M} x_j(t) \cdot h_{ij}(t-k) \tag{11}$$

which is shown for the 2 by 2 case in the following figure:

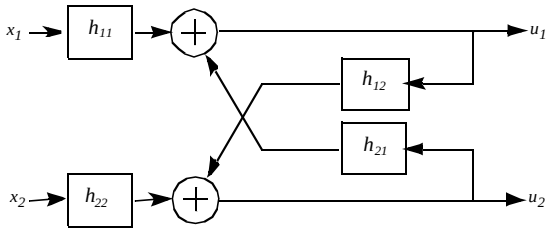

Figure 3: An alternate unmixing procedure

Optimizing with respect to the unmixing filters we get the following update rules for the 2 by 2 case:

$$\Delta h_{ii}(0) \propto -2 \cdot y_i(t) \cdot x_i(t) + \frac{1}{h_{ii}(0)} \tag{12}$$

$$\Delta h_{ii}(k) \propto -2 \cdot y_i(t) \cdot x_i(t-k), \quad k > 0 \tag{13}$$

---

1. This problem has a Riemannian structure and Bell proposed a Euclidean update rule. By 'warping' the rule to the cost surface it is possible to get better performance.
2. This is actually an approximation of the solution since we are using FIR filters to invert FIR filtering. The solution can be also derived using IIR unmixing filters but the performance will not differ much for real world mixing and the instability of IIR filters is better avoided.

$$\Delta h_{ij}(k) = -2 \cdot y_i(t) \cdot u_j(t-k) \quad (14)$$

This algorithm doesn't have as strong local minima and is reported to perform better. It is also possible to use IIR filters for unmixing for both of the above algorithms.

However there is still a problem of efficiency, since for N sources we require N convolutions. Given that the deconvolving filters need to have a considerable length for real world problems we are posed with a very expensive algorithm.

2. Frequency Domain Extension

The algorithms that were introduced in the previous section exhibit problems which are inherent in the time domain, mainly the statistical dependence between the unmixing filter taps (which hinders convergence) and the heavy computation which is required for the unmixing convolutions. In this section we present a frequency domain formulation of these algorithms which bypasses these problems.

2.1. The Algorithm

We can use the Short Time Fourier Transform (STFT) to decompose a time series x to X, where $X^{(t)}$ is the t th spectrum that the STFT gives and $X^{(t)}(f)$ its f th frequency bin.

Looking again at equation (4) we can rewrite it in terms of the STFT as:

$$X_i^{(t)}(f) = \sum_{j=1}^{N} S_j^{(t)}(f) H_{ij}(f) \quad (15)$$

where H_{ij} is the Fourier transform of h_{ij}, $S_j^{(t)}$ the Fourier transform of s_i and $X_i^{(t)}$ the Fourier transform of x_i.

By closer inspection of equation (15) we notice a resemblance to equation (1). In fact every frequency bin of the resulting mixtures x_i, is an instantaneous mix of the corresponding bins of the original sources s_j. This refers us back to the original problem with the added complexity that the input sequences and the mixing matrix are now complex valued. In order to invert the mixing in the frequency domain we need to apply the instantaneous unmixing algorithm on every frequency bin track we get from the STFT. The implementation of an unmixing system is depicted in figure 4.

The boxes labeled with W are implementations of the instantaneous unmixing algorithm presented in section 1.1. The only difference is that the learning rule is derived for the complex number domain. The only two adjustments to be made are:

- The $y_i(t) = \tanh u_i(t)$ used in the real number domain case has to be changed to $y_i(t) = \tanh Re(u_i(t)) + \tanh Im(u_i(t))$, since in the complex domain $\tanh(\cdot)$ is unbounded thus inhibiting the gradient descent (see Smaragdis (1997) for a more detailed explanation).
- The matrix transpositions in the learning rule have to be substituted with Hermitian transpositions.

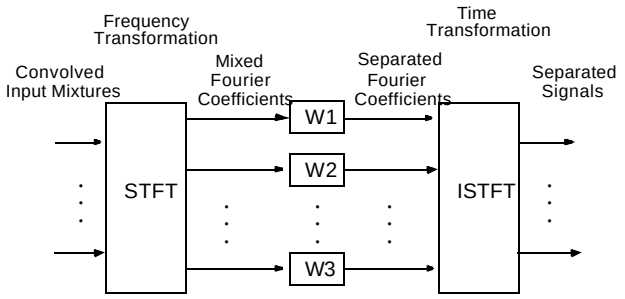

Figure 4: The frequency domain separation algorithm

Applying these two rules on a modified version of Amari's natural gradient rule we derive the following update equation:

$$\Delta \mathbf{W}_f \propto (\mathbf{I} - \mathbf{Y}_f \cdot \mathbf{U}_f^H) \cdot \mathbf{W}_f \quad (16)$$

where f is the frequency bin index and $\mathbf{Y}_f = \tanh \mathbf{U}_f$.

The frequency domain algorithm has a computational complexity of $O(M \cdot \log M)$, where M is the length of the unmixing filters, whereas the time domain approaches are $O(M^2)$. Given that FIR filters are used to invert FIR filters, a considerable filter length is usually required. The time domain algorithms are unable to provide an efficient algorithm which prohibits their use in a real-time situation.

In addition to dramatic performance improvements this approach eliminates the problems of local minima. With the time domain approaches, an update of a filter tap would influence the taps following it. In this case the filter parameters are lying on an orthogonal space and updates of one parameter have no influence over the rest. Also due to this decomposition to smaller independent problems the length of the deconvolving filters does not complicate convergence, since it only raises the number of smaller problems but not necessarily their complexity.

2.2. Implementations

This algorithm can run in two forms, on-line or off-line. The off-line version computes the unmixing matrix of every frequency bin for the entire input and then proceeds to the next bin. In order to assure that we have the same permutation at every bin we use the weights of the previous frequency bin as the initial state of the unmixing matrix of the current bin. The data is zero padded before

the frequency transformation, so that spectra are interpolated and we ensure that the unmixing matrices of adjacent bins are numerically close. In order to assure proper scaling, the unmixing matrix is scaled to have a determinant of 1. This ensures volume conservation for every frequency bin and avoids random boosts or cuts in different frequency bands.

The main limitation of the off-line version is that it can only deal with static mixtures. Consequently, it is impossible to separate moving sources or time varying room responses.

The on-line version was implemented to deal with this problem. In this implementation the unmixing matrices are updated every time a new spectrum is computed. With this approach we have no way of controlling the permutation of the unmixing matrices. In general using low learning rates in adaptation results in uniform permutation and the algorithm relies on this. The measures used for scaling in the off-line case are used here too.

The on-line version is efficient enough to run in real time on common high-end computers at high sampling rates for the 2 by 2 case. Unlike the off-line version it is capable of separating non-static mixes and requires an adaptation time of 10 to 100 spectra depending on the amount of the mixing changes.

2.3. Results

Preliminary results were collected by both the off-line and the on-line algorithms. Two sources were synthetically mixed with various filters. The off-line algorithm had an overall good performance. Separation using sparse filters of small length yielded almost perfect results. In addition to this the algorithm was robust enough to attenuate interference of other sources by up to 6dB for sources mixed with 100 tap filters of Gaussian noise. Problems were introduced when the mixing filters were non-minimum phase in which case non-causal unmixing filters were required. Due to the causality of the design of the algorithm, separation was impossible.

The on-line algorithm, due to the permutation inconsistencies among frequency bins, has not been as successful, it was however tested with mixing filters of 20 taps of sparse filters, in which case the interfering sources were forced to be inaudible.

3. Conclusions

An extension to robust algorithms for blind source separation was formulated in the frequency domain. By this domain shift, certain convergence problems inherent in the time domain have been eliminated and efficiency was improved dramatically.

This algorithm can also support on-line learning, for cases of dynamic mixtures and real-time implementations, but certain algorithm characteristics can hinder performance. In the off-line form this algorithm has proved to be more satisfactory.

Future work will include ways to make the on-line version more robust and modifications to deal with non-minimum phase mixing filters.

4. Acknowledgments

The author would like to thank the Machine Listening Group at the MIT Media Lab for valuable input and support in this project, as well as Kari Torkkola and V. Michael Bove for helpful comments.

REFERENCES

1. Bell, A.J. and T.J. Sejnowski. 1995. An Information Maximization Approach to Blind Separation and Blind Deconvolution", *Neural Computation 7.* MIT Press, Cambridge, MA.

2. Torkkola, K. 1996. "Blind Separation of Convolved Sources Based on Information Maximization", *IEEE Workshop on Neural Networks for Signal Processing,* Kyoto Japan.

3. Lee, T., A. Bell, and R. Lambert. 1997. "Blind Separation of Convolved and Delayed Sources", *Advances in Neural Information Processing Systems 9.* MIT Press, Cambridge, MA.

4. Amari, S., A. Cichocki, and H.H. Yang. 1996. "A New Learning Algorithm for Blind Signal Separation", *Advances in Neural Information Processing systems 8.* MIT Press, Cambridge, MA.

5. Cichocki A., S. Amari, M. Adachi, and W. Kasprzak. 1996. "Self-Adaptive Neural Networks for Blind Source Separation of Sources", *1996 IEEE International Symposium on Circuits and Systems, ISCAS'96, Vol. 2*, IEEE, Piscataway, NJ.

6. McKay, D. 1996. "Maximum Likelihood and Covariant Algorithms for Independent Component Analysis", *Draft paper available at:*

   `ftp://wol.ra.phy.cam.ac.uk/pub/mackay/ica.ps.gz`

7. Smaragdis, P. 1997. "Information Theoretic Approaches to Sound Separation", *Masters Thesis,* MIT Media Arts and Sciences Dept. Cambridge, MA.