COMBINING TIME-DELAYED DECORRELATION AND ICA: TOWARDS SOLVING THE COCKTAIL PARTY PROBLEM

Te-Won Lee

The Salk Institute, CNL
La Jolla, CA 92037
tewon@salk.edu

Andreas Ziehe

GMD, FIRST
Berlin, Germany
ziehe@first.gmd.de

Reinhold Orglmeister

Berlin University of Technology,
10603 Berlin, Germany,
orglm@tubife1.ee.tu-berlin.de

Terrence Sejnowski

The Salk Institute, CNL
La Jolla, CA 92037
terry@salk.edu

ABSTRACT

We present methods to separate blindly mixed signals recorded in a room. The learning algorithm is based on the information maximization in a single layer neural network. We focus on the implementation of the learning algorithm and on issues that arise when separating speakers in room recordings. We used an infomax approach in a feedforward neural network implemented in the frequency domain using the polynomial filter matrix algebra technique. Fast convergence speed was achieved by using a time-delayed decorrelation method as a preprocessing step. Under minimum-phase mixing conditions this preprocessing step was sufficient for the separation of signals. These methods successfully separated a recorded voice with music in the background (cocktail party problem). Finally, we discuss problems that arise in real world recordings and their potential solutions.

1. INTRODUCTION

In a *cocktail party*, the problem is to focus one's listening attention on a single talker among a din of conversations and background noise and extract one voice. We model this as a linear mixing and filtering of independent sound sources. Assuming that the original signals are independent we can apply an Independent Component Analysis (ICA) algorithm to blindly recover the unknown sources. Bell and Sejnowski [2] have shown that information maximization can be used to separate many independent sources. Torkkola [15] extended this approach to a feedback system with only cross filters. A full filter feedback system is presented in [11] and [4]. Since feedback systems are limited to minimum-phase mixing systems, the general assumption of non-minimum-phase systems can be overcome by using a feedforward unmixing system [10, 12, 8, 5]. The infomax algorithm has been used to separate voices recorded in real environments [10, 11, 6].

A simple time-delayed decorrelation (TDD) algorithm [14] has been shown to be highly effective under the minimum-phase constraint. The TDD algorithm can in some circumstances achieve the same separation quality much faster which is important for online implementations. The convergence of the infomax algorithm can be improved by using the TDD algorithm as a preprocessing step. In this paper, we show that this method increases the convergence speed and may allow for online use of the algorithm. Regarding applications, the recognition rate in an automatic speech recognition system can be increased by using these methods as a preprocessing step [12, 6].

2. PROBLEM STATEMENT AND ASSUMPTIONS

Assume that there is an M dimensional zero-mean vector $\mathbf{s}(t)$ such that the components of $\mathbf{s}(t) = [s_1(t), \cdots, s_M(t)]^T$ are mutually independent. The M signals are transmitted through a medium so that an array of N sensors picks up a set of signals $\mathbf{x}(t) = [x_1(t) \ldots x_N(t)]^T$, each of which has been mixed, delayed and filtered as follows:

$$x_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{P-1} a_{ijk} s_j(t - D_{ij} - k). \qquad (1)$$

D_{ij} are entries in a matrix of delays and there is an P-point filter, $\mathbf{a}_{ij}$, between the the jth source and the ith sensor. The problem is to recover the original signals, $\mathbf{s}(t)$ given only the sensory outputs $\mathbf{x}(t)$. The infomax approach depends on the following assumptions: (1) The number of sensors is

greater or equal to the number of sources $N \geq M$. (2) The sources $\mathbf{s}(t)$ are at each time instant mutually independent and each source is white, ie: there are no dependencies between time points. Assumption 1 is needed to make $\mathbf{A}(z)$ a full rank matrix of filters which holds for most physical situations. Assumption 2 is the basis of ICA. However, this is not true for natural signals. The algorithm will whiten: it will remove dependencies across time which already existed in the original source signals, s_i. There are two ways to overcome this problem: (1) to omit the direct filters and hence recover a filtered version of the original signal [15] or (2) to restore the characteristic autocorrelations (amplitude spectra) of the sources by post-processing [9].

3. FEEDFORWARD ARCHITECTURE

The feedforward architecture can be described as:

$$u_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{P-1} w_{ijk} x_j(t-k), \qquad (2)$$

where the filters, $\mathbf{w}_{ij}$, reproduce, at the u_i, the original uncorrupted source signals, s_i. This was the architecture implicitly assumed in [2]. Although a feedback architecture requires less parameters, it is unstable for non-minimum-phase of $\mathbf{A}(z)$. The advantage of the feedforward system is that it can approximate a more general inverse system. For example, a non-minimum phase system will occur when a microphone picks up an echo that is stronger than the direct signal. Then the increase in negative phase is directly related to the amount of temporal delay of a narrowband component at that frequency. Hence, the minimum phase lag property or the minimum group delay property of a non-minimum-phase system is not guaranteed. Since we cannot obtain prior knowledge about the mixing properties in room recordings we have to assume a non-minimum phase system which may have a non-causal filter system inverse. Strictly non-causal filters (dependency on an infinite number of past time-samples) cannot be implemented. However, any non-minimum or true phase system can be expressed as $W(z) = W_{min}(z)W_{AP}(z)$ where $W_{min}(z)$ is a minimum phase system and $W_{AP}(z)$ is an *all-pass* system. $W_{min}(z)$ has all its poles and zeros inside the unit circle and $W_{AP}(z)$ represents a time delay with a unit frequency magnitude response. Therefore, $W_{AP}(z)$ preserves the amplitude frequency spectrum and imposes a time delay on $W(z)$ by reflecting the zeros outside the unit circle to their conjugate reciprocal location inside the unit circle. By time-delaying the inverting system up to $P/2$ taps, P being the size of the inverting filter, we introduce a $P/2$-order $W_{AP}(z)$ which can realize non-causal systems.

4. LEARNING ALGORITHM

Learning in this feedforward architecture is performed by maximizing the joint entropy, $H(\mathbf{y}(t))$, of the random vector $\mathbf{y}(t) = g(\mathbf{u}(t))$, where g is a bounded monotonic nonlinear function (e.g. a sigmoid function). The relation and theory between ICA and infomax is further explained in [2, 13]. The general learning rule is:

$$\Delta \mathbf{W} \propto \frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} = \left[\mathbf{I} + \left(\frac{\frac{\partial p(\mathbf{u})}{\partial \mathbf{u}}}{p(\mathbf{u})} \right) \mathbf{u}^T \right] \mathbf{W} \qquad (3)$$

where $\mathbf{I}$ is the identity matrix and $p(\mathbf{u}) = \frac{\partial \mathbf{y}}{\partial \mathbf{u}}$. This is the learning rule in [2] using the natural gradient extension by [1, 3]. We may keep the form of the equation in eq.3 for the full filter system by moving into the frequency domain representation where the elements of the matrices are filters. Then the multiplication operation replaces the convolution property. Lambert [10] showed that FIR polynomial matrix algebra can be used as an efficient tool to elegantly solve problems for the multichannel source separation. The goal of using the FIR polynomial matrix algebra is to extend the algebra of scalar matrices to the algebra of matrices of filters (time-domain) or polynomials (frequency domain). The methods for computing functions of an FIR filter, such as an inverse, involve the formation of a circulant data matrix. Due to this nature we move to the frequency domain representation where eigencolumns of the circulant matrix are the discrete Fourier basis functions of the FFT of corresponding length. The filters now become polynomials of the Laurent series extension (z-transform) and the convolution and deconvolution of filters is reduced to multiplication and division of polynomials. The prepending of post-pending of zeros is needed to produce a good estimate of the double-sided Laurent series expansion to allow for non-causal expansions of non-minimum phase roots. The circular reordering in the time domain shifts the zeroth lag to the center of the filter (FFTSHIFT). Lambert [10] presents a complete proof and justification of FIR polynomials. The learning algorithm for the two sources and two sensors problem can be reformulated from eq.3 as follows:

$$\Delta \mathbf{W}(z) = \qquad (4)$$
$$\left(\begin{bmatrix} \bar{1} & \bar{0} \\ \bar{0} & \bar{1} \end{bmatrix} + \begin{bmatrix} \mathrm{FFT}(\hat{y}_1) \\ \mathrm{FFT}(\hat{y}_2) \end{bmatrix} \begin{bmatrix} \mathrm{FFT}(u_1) & \mathrm{FFT}(u_2) \end{bmatrix}^* \right) \times \begin{bmatrix} W_{11}(z) & W_{21}(z) \\ W_{12}(z) & W_{22}(z) \end{bmatrix}$$

where $\bar{1}$ and $\bar{0}$ denote vectors (of the length of the FFT operation) of ones and zeros respectively. Note that the neural processor $\hat{y}_i = \frac{\partial p(u_i)/\partial u_i}{p(u_i)}$ still operates in the time domain and the FFT is applied at the output and $*$ denotes the complex conjugate form. Eq.4 is of the form of the least

mean squared (LMS) adaptive filters. A fast implementation of the LMS adaptive filters in the frequency domain can be achieved by employing the *overlap and save* block LMS technique, i.e. two blocks are processed simultaneously and x_k is shifted by one block after each iteration.

$$X(z) = \text{FFT}[x_{(k-1)n} \cdots x_{kn-1} x_{kn} \cdots x_{kn+n-1}]. \quad (5)$$

For a block size of 1024 FFT-points the method is 16 times faster than the conventional LMS method [7].

5. TIME-DELAYED DECORRELATION AS A PREPROCESSING STEP

Iteratively updating the weights for the filter is crucial when considering online learning where source signals are non stationary. However, the convergence speed may be increased by using a fairly computationally inexpensive time-delayed decorrelation algorithm as a preprocessing step [14]. The main point of TDD is to diagonalize the covariance matrix $\mathbf{C}_0 = \langle \mathbf{x}(t)\mathbf{x}(t)^T \rangle$ for $\tau = 0$ (no time-delay) and at the same time diagonalize the covariance matrix for a given delay $\mathbf{C}_\tau = \langle \mathbf{x}(t)\mathbf{x}(t-\tau)^T \rangle$. This leads to an eigenvalue problem as described in [14]:

$$(\mathbf{C}_0\mathbf{C}_\tau^{-1})\mathbf{A} = \mathbf{A}(\mathbf{\Lambda}_0\mathbf{\Lambda}_\tau^{-1}) \quad (6)$$

where $\mathbf{\Lambda}$ is the diagonal matrix with elements that are the eigenvalues of the corresponding covariance matrix. The TDD algorithm can be extended to a matrix of filters [6]. The main extension consists of transforming the signals $x_i(t)$ into the frequency domain $X_i(z)$ and hence creating a spectrogram. A correlation matrix can be computed as in eq.6 for each frequency bin. The inverse of $\mathbf{A}(z)$ multiplied with the spectrogram results in the frequency domain decorrelated signals which can be reconstructed using an IFFT and overlap and zero-padding technique. The unmixing filters in the time-domain are obtained by IFFT of $\mathbf{A}(z)$. There are two optimizing steps improving the separation performance: (a) setting the direct filters to identity and therefore avoiding the whitening problem (b) optimizing a decorrelation-based cost function [6] (c) optimizing τ as a function of decorrelation cost function. Point (c) is crucial to achieve good separation results and therefore requires a secondary optimization step. The main advantage of the TDD algorithm is the computational efficiency in computing the cross-filters since no adaptation is necessary. An online-version of this algorithm could be implemented in a block mode in which successive blocks of data points (e.g. 128, 256) are processed.

6. EXPERIMENTAL RESULTS

In Figure 1 we show an example of a recording in a room obtained by Yellin[1] and Weinstein (1996). Here, a music signal and a voice signal that was played by an audio system and the signals were recorded with two microphones located close to the sources (60 cm). Figure 1 (a) and (b) show the recorded signals. Two cross filters with 128 taps each were computed using the TDD algorithm. The unmixed signals were obtained after 10 seconds on a Sparc10 workstation using MATLAB. Figure 1 (c) shows the recovered speech signal and Figure 1 (d) shows the music signal using the TDD algorithm by Molgedey and Schuster [14]. For the same recording we used the learning rule in eq.4 and obtained slightly better separating results shown in Figure 1 (e) and (f) with the same set of parameters but slow convergence speed (about 5 min with annealing the learning rate). The infomax results are very similar to the results obtained by Yellin and Weinstein (1996) using a fourth-order cumulant-based method. Unfortunately, the signal to noise ratio is not measurable due to the unavailability of the original speech and music signals. The use of the TDD algorithm as a preprocessing step for infomax approximately doubled the convergence speed. In many experiments such as the recordings in [6] and [12] the TDD algorithm by itself gave results similar in quality to infomax. However, in experiments performed in a large con-

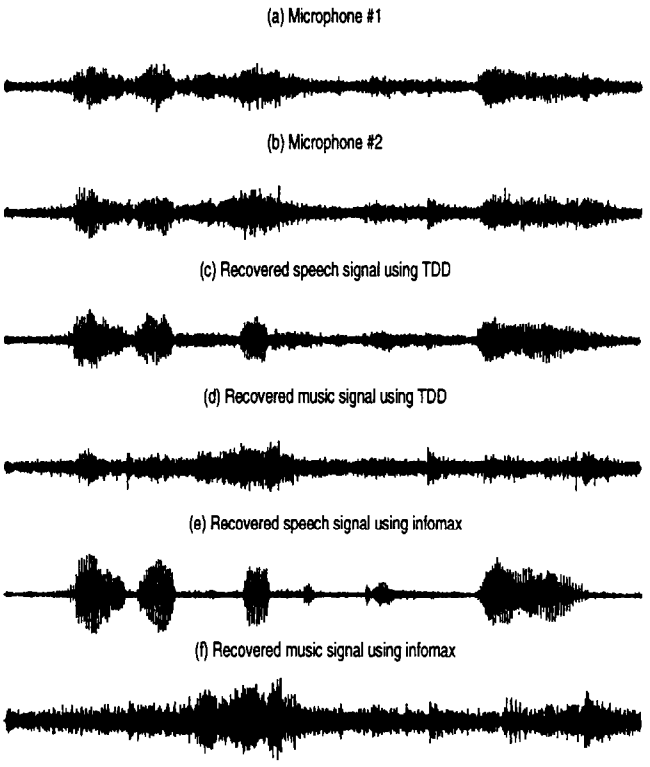

Figure 1: Room recordings from Yellin and Weinstein (1996): (a) microphone 1, (b) microphone 2. The separated signals using the TDD algorithm are shown for speech and music in (c) and (d). Slightly better results were obtained with eq.4 as shown in (e) and (f).

[1] We are grateful to Dr. Yellin for making the data available.

ference room with microphones located 3 m from the sources, the decorrelation algorithm performed poorly in separating the signals.

7. DISCUSSION

We have presented separation results of room recordings using the TDD algorithm and the infomax ICA algorithm. While in general infomax achieved better separation results than the TDD algorithm, the convergence speed was slow. The TDD algorithm, however, may allow for online implementations for real-time applications such as speech recognition and may be used as a preprocessing step for infomax to speed up convergence. Additional improvements can be made in optimizing the TDD algorithm and its combination with infomax. The infomax algorithm has several limitations that have not yet been resolved: (1) The number of sensors must be greater or equal the number of sources. (2) A noisy ICA model formulation for recorded signals has not been addressed. (3) For many experiments, we observed that the algorithm failed to clearly separate non-stationary signals such as recordings from people with slight movements while they talk. In contrast, humans can track non-stationary sources and extract signals from a high number of sources with only two sensors. Another source of improvement may be found in the use of cochlear filter banks to compute an frequency spectrum, which may be less sensitive to non-stationary sources.

Acknowledgments

We are grateful to Klaus-Robert Müller, Tony Bell and Russ Lambert for discussions.

8. REFERENCES

[1] Amari, S. (1997). Natural gradient works efficiently in learning. *Neural Computation*, in press.

[2] Bell, A. and Sejnowski, T. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7:1129–1159.

[3] Cardoso, J.-F. and Laheld, B. (1996). Equivariant adaptive source separation. *IEEE Trans. on S.P.*, 45(2)434–444.

[4] Cichocki, A., Amari, S., and Cao, J. (1997). Neural network models for blind separation of time delayed and convolved signals. *Japanease IEICE Transaction on Fundamentals*, E-82-A(9).

[5] Douglas, S., Cichocki, A., and Amari, S. (1997). Multichannel blind separation and deconvolution of sources with arbitrary distribiutions. In *Proc. IEEE Workshop on NNSP*, 436–445.

[6] Ehlers, F. and Schuster, H. (1997). Blind separation of convolutive mixtures and an application in automatic speech recognition in noisy environment. *IEEE Transactions on Signal processing*, 45(10):2608–2609.

[7] Ferrara, E. (1980). Fast implementation of lms adaptive filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):474–478.

[8] Girolami, M. and Fyfe, C. (1997). An extended exploratory projection pursuit network with linear and nonlinear anti-hebbian connections applied to the cocktail party problem. *Neural Networks*, in press.

[9] Haykin, S. (1991). *Adaptive filter theory*. Prentice-Hall.

[10] Lambert, R. (1996). Multichannel blind deconvolution: Fir matrix algebra and separation of multipath mixtures. Thesis, University of Southern California, Department of Electrical Engineering.

[11] Lee, T.-W., Bell, A., and Lambert, R. (1997). Blind separation of convolved and delayed sources. In *Advances in Neural Information Processing Systems 9*, 758–764. MIT Press.

[12] Lee, T.-W., Bell, A., and Orglmeister, R. (1997). Blind source separation of real-world signals. In *Proc. ICNN*, 2129–2135, Houston, USA.

[13] Lee, T.-W., Girolami, M., Bell, A., and Sejnowski, T. (1997). A unifying framework for independent component analysis. *International Journal on Mathematical and Computer Models*, to appear.

[14] Molgedey, L. and Schuster, H. (1994) Separation of independent signals using time-delayed correlations. *Physical Review Letters*, 72(23):3634–3637.

[15] Torkkola, K. (1996). Blind separation of convolved sources based on information maximization. In *IEEE Workshop on Neural Networks for Signal Processing*, 423–432, Kyoto, Japan.

[16] Yellin, D. and Weinstein, E. (1996). Multichannel signal separation: Methods and analysis. *IEEE Transactions on Signal Processing*, 44(1):106–118.