An Approach to Blind Source Separation of Speech Signals

Shiro Ikeda, Noboru Murata
RIKEN Brain Science Institute
Wako, Japan

Abstract

In this paper we introduce a new technique for blind source separation of speech signals. We focused on the temporal structure of signals which is not always the case in other major approaches. The idea is to apply the decorrelation method proposed by Molgedey and Schuster in time-frequency domain. We show some results of experiments with artificial data and speech data recorded in the real environment. Our algorithm needs considerably straightforward calculation and includes only a few parameters to be tuned.

1 Introduction

In this paper, we propose a blind source separation (BSS) method for speech signals recorded in a real environment. Speech signals have a temporal structure that it is stationary for a period shorter than 50∼60msec but not stationary anymore if it is longer than 50∼60msec and around 100msec. We use this time structure to build an algorithm.

The problem of BSS is defined as follows. Source signals are denoted by a vector $\boldsymbol{s}(t) = (s_1(t), \cdots, s_n(t))^T$ and it is assumed that each component of $\boldsymbol{s}(t)$ is independent to each other and mean 0. Recorded signals are $\boldsymbol{x}(t) = (x_1(t), \cdots, x_n(t))^T$. We usually simulate a real-room recording with FIR filters, s.t. the observations are convolutive mixtures of source signals,

$$\boldsymbol{x}(t) = A * \boldsymbol{s}(t) = \left(\sum_k a_{ik} * s_k(t) \right), \quad a_{ik} * s_k(t) = \sum_{\tau=0}^{\tau_{max}} a_{ik}(\tau) s_k(t-\tau) ,$$

where $A(t)$ is a function of time, $a_{ik} * s_k(t)$ is the convolution of $a_{ik}(t)$ and $s_k(t)$, where $a_{ik}(t)$ is the impulse response from source signal k to sensor i. The goal of BSS is to separate signals into the components which are mutually independent without knowing operator A and source signals $\boldsymbol{s}(t)$.

Basic BSS approaches have been developed for instantaneous mixtures. For convolutive mixtures, there are some trials[2]. We use the windowed-Fourier transform as in [3][5] to transform mixed source signals into the time-frequency domain. After that we apply Molgedey and Schuster's decorrelation algorithm[4] to the signals of each frequency independently. Most of the BSS approaches usually ignore the ambiguities of the amplitude and the permutation, but we have to remove these ambiguities to reconstruct the separated signals. Our idea is to use the inverse of the decorrelating matrices and the envelope of the speech signal.

2 Decorrelation Algorithm for Instantaneous Mixture

First we explain the decorrelation algorithm by Molgedey and Schuster [4] which was proposed for instantaneous mixtures, i.e. $\boldsymbol{x}(t) = A\boldsymbol{s}(t)$ where A is an $n \times n$ matrix.

The correlation matrix of observations is written as

$$\left\langle \boldsymbol{x}(t)\boldsymbol{x}(t+\tau)^T \right\rangle = R_{xx}(\tau) = A \left\langle \boldsymbol{s}(t)\boldsymbol{s}(t+\tau)^T \right\rangle A^T = AR_{ss}(\tau)A^T, \quad (1)$$

where $R_{xx}(\tau)$ and $R_{ss}(\tau)$ are correlation matrices. Since each component of $\boldsymbol{s}(t)$ is independent, $R_{ss}(\tau)$ is diagonal for any τ. Molgeday and Schuster showed that the BSS problem of finding B is reduced to solve the eigenvalue problem

$$\left(R_{xx}(\tau_1)R_{xx}(\tau_2)^{-1}\right) B = B\left(\Lambda_1\Lambda_2^{-1}\right). \quad (2)$$

This problem can also be solved by simultaneous diagonalization of matrices, where the number of the matrices doesn't have to be 2 but any number,

$$BR_{xx}(\tau_i)B^T = \Lambda_i, \quad i = 1, \ldots, r. \quad (3)$$

Although from the effect of the noise and small correlations among the source signals, (3) does not hold in practice. We implemented in the way to minimize the off-diagnal components of the matirices $BR_{xx}(\tau_i)B^T$. In order to obtain B, we use the algorithm which only needs straightforward calculations[6]. It consists of two procedures, sphering and rotation. (Fig.1)

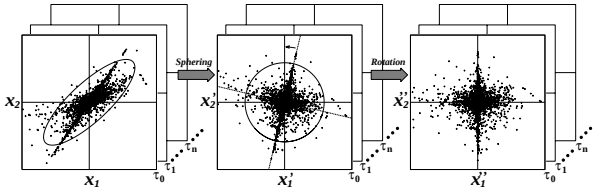


Figure 1: Sphering and Rotation

Sphering is a procedure to obtain a matrix V which satisfies,

$$VR_{xx}(0)V^T = I \quad (4)$$

and rotation is a procedure to remove off-diagonal elements of correlation matrices with an orthogonal transformation. The implementation is to find an orthogonal matrix C which minimizes

$$\sum_{l=1}^{r} \sum_{i \neq k} \left| (CVR_{xx}(\tau_l)V^TC^T)_{ik} \right|^2, \quad (5)$$

where $(*)_{ik}$ is the ik-element of a matrix. Cardoso and Souloumiac gave an implementation [1] with Jacobi-like algorithm to obtain C. Finally, matrix B is given by $B = CV$. An advantage of this method is that it uses only the second order statistics and fixed amount of computation.

3 Proposed Method

In this section, the detail of the algorithm is shown along with the flow of it. First, the windowed-Fourier transform is applied to convolutive mixed signals,

$$\begin{aligned} \hat{\boldsymbol{x}}(\omega, t_s) &= \sum_t e^{-j\omega t} \boldsymbol{x}(t) w(t - t_s), \\ \omega &= 0, \tfrac{1}{N}2\pi, \ldots, \tfrac{N-1}{N}2\pi, \quad t_s = 0, \Delta T, 2\Delta T, \ldots \end{aligned} \tag{6}$$

where ω denotes the frequency and N denotes the number of points of the discrete Fourier transform, t_s denotes the window position, w is a window function (we used Hamming window) and ΔT is the shifting interval of moving windows. Let us redefine a $\hat{\boldsymbol{x}}(\omega, t_s)$ for a fixed frequency ω as $\hat{\boldsymbol{x}}_\omega(t_s) = \hat{\boldsymbol{x}}(\omega, t_s)$. If the window length is long enough compared to the impulse response of $A(t)$, the relationship between observations and sources can be approximated as,

$$\hat{\boldsymbol{x}}_\omega(t_s) = \hat{A}(\omega)\hat{\boldsymbol{s}}(\omega, t_s), \tag{7}$$

where $\hat{A}(\omega)$ is the Fourier transform of operator $A(t)$, and $\hat{\boldsymbol{s}}(\omega, t_s)$ is the windowed-Fourier transform of $\boldsymbol{s}(t)$. This shows that for fixed ω, a convolutive mixture is simply an instantaneous mixture. We extend the algorithm in §2 to complex values by substituting a Hermite matrix for a symmetric matrix, and apply it for each frequency. As the result, we have a separated time sequence,

$$\hat{\boldsymbol{u}}_\omega(t_s) = B(\omega)\hat{\boldsymbol{x}}_\omega(t_s). \tag{8}$$

Since BSS algorithms cannot solve the ambiguity of amplitude and permutation, even if we put each component of $\hat{\boldsymbol{u}}_\omega(t_s)$ along with ω, amplitudes are irregular and different independent sources will be mixed up. The problem of irregular amplitude can be solved by putting back the separated independent components to the sensor input with the inverse matrices $B(\omega)^{-1}$. Let us define $\hat{\boldsymbol{v}}_\omega(t_s; i)$ as,

$$\hat{\boldsymbol{v}}_\omega(t_s; i) = B(\omega)^{-1}(0 \ldots 0, \hat{u}_{i,\omega}(t_s), 0 \ldots 0)^T, \quad i = 1, \ldots, n \tag{9}$$

where $\hat{v}_{k,\omega}(t_s; i)$ represents the input of i-th independent component of $\hat{\boldsymbol{u}}_\omega(t_s)$ into the k-th ($k = 1, \ldots, n$) sensor. We applied $B(\omega)$ and $B(\omega)^{-1}$ to obtain $\hat{\boldsymbol{v}}_\omega(t_s; i)$, therefore $\hat{\boldsymbol{v}}_\omega(t_s; i)$ has no ambiguity of amplitude.

Remaining problem is permutation. We made an assumption that even for different frequencies, if the original souce is the same, the envelopes are similar. We utilize this idea for solving the permutation. Let us define an operator $\mathcal{E}$ to take the envelope as,

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s; i) = \frac{1}{2M} \sum_{t'_s = t_s - M}^{t_s + M} \sum_{k=1}^{n} |\hat{v}_{k,\omega}(t'_s; i)|, \tag{10}$$

where M is a positive constant and $\hat{v}_{k,\omega}(t_s;i)$ denotes the k-th element of $\hat{\boldsymbol{v}}_\omega(t_s;i)$. Inner product and norm are defined as

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\cdot\mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(k)=\sum_{t_s}\mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s;i)\mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(t_s;k), \tag{11}$$

$$\|\mathcal{E}\boldsymbol{v}_\omega(i)\|=\sqrt{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\cdot\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)}, \tag{12}$$

and we define the similarity between two envelopes as the following,

$$\mathrm{sim}(\omega)=\sum_{i\neq k}\frac{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\cdot\mathcal{E}\hat{\boldsymbol{v}}_\omega(k)}{\|\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\|\|\mathcal{E}\hat{\boldsymbol{v}}_\omega(k)\|}. \tag{13}$$

Using these operations, we solve the permutation as follows:

Solve the Permutation

$\boldsymbol{\omega}=$ **sort**(ω,sim) *sorting* ω *to be* $\mathrm{sim}(\omega_1)<\cdots<\mathrm{sim}(\omega_N)$
for $i=1$ **to** n **do**
  $\hat{\boldsymbol{y}}_{\omega_1}(t_s;i):=\hat{\boldsymbol{v}}_{\omega_1}(t_s;i)$
done
for $l=2$ **to** N **do**
  for $i=1$ **to** n **do**
    $\sigma(i):=$ **argmax**$_{i'}\left\{\mathcal{E}\hat{\boldsymbol{v}}_{\omega_k}(i')\cdot\left(\sum_{k=1}^{l-1}\mathcal{E}\hat{\boldsymbol{y}}_{\omega_k}(i)\right)\right\}$
    $\hat{\boldsymbol{y}}_{\omega_l}(t_s;i):=\hat{\boldsymbol{v}}_{\omega_l}(t_s;\sigma(i))$
  done
done

As a result, we obtain separated spectrograms as $\hat{\boldsymbol{y}}_\omega(t_s;i)$. Applying inverse Fourier transform, finally we get a set of separated sources

$$\boldsymbol{y}(t;i)=\frac{1}{2\pi}\cdot\frac{1}{W(t)}\sum_{t_s}\sum_{\omega}e^{j\omega(t-t_s)}\hat{\boldsymbol{y}}_\omega(t_s;i),\quad i=1,\ldots,n \tag{14}$$

where $W(t)=\sum_{t_s}w(t-t_s)$. Note that each $y_k(t;i)$ represents a separated independent component i on sensor k, and $\sum_i\boldsymbol{y}(t;i)=\boldsymbol{x}(t)$ holds.

4 Experimental Results

4.1 Artificial Data

We show a result of an experiment on a set of data which were recorded separately and mixed on a computer. Since we wanted to simulate the general problem of recording sounds in a real environment, we built a virtual room as Fig.2 and calculated reflections and delays. We supposed that each wall, floor and ceil reflect the sound once and the strength of sounds varies in proportion to the inverse square of the distance. The strength of the reflection is 0.1 in power for any frequency.

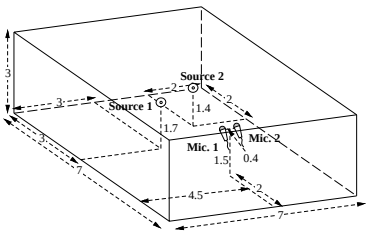
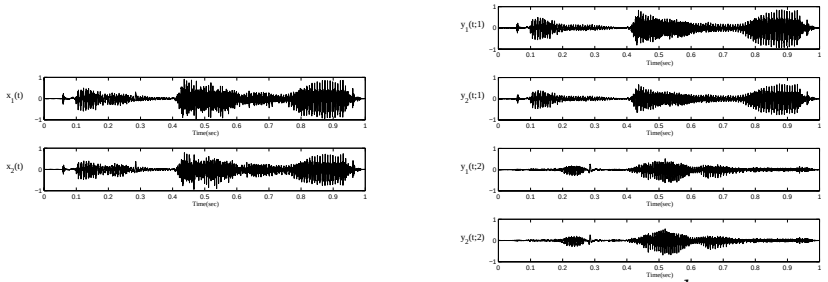


Figure 2: Virtual Room

Since we know the true sources and the mixing rates, we evaluate the performance with SNR (Signal to Noise Ratio) which is defined as,

$$\text{signal}_i(t;k) = a_{ik}s_k(t), \quad \text{error}_i(t;k) = y_i(t;k) - \text{signal}_i(t;k)$$
$$\text{SNR}_{ik} = 10\log_{10}\frac{\sum_t \text{signal}_i(t;k)^2}{\sum_t \text{error}_i(t;k)^2}.$$

We applied our algorithm, changing the window length from 8msec to 32msec.



a b

Figure 3: a: Mixed Signals in a Virtual Room b: Separated Signals

The SNRs of these results are shown in Tab. 1. Our approach with the window length of 32msec gave the best SNRs. Mixed inputs (Fig.3.a) and separated signals (Fig.3.b) are shown. Signals were clearly separated.

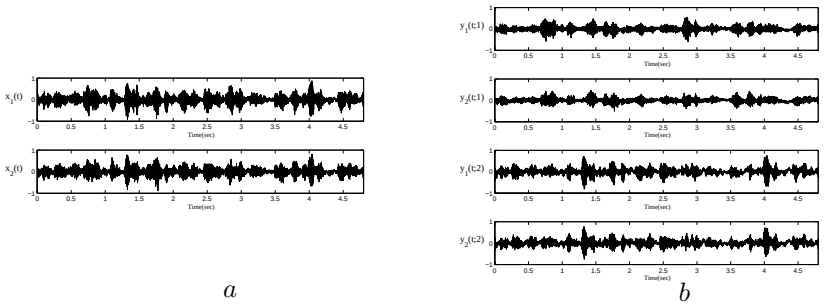
Table 1: SNRs (dB) for Separated Signals (ΔT is 1.25msec and r in (3)is 40)

		SNR_{11}	SNR_{12}	SNR_{21}	SNR_{22}
Window length	8msec	4.36	6.32	11.94	11.66
	16msec	4.72	6.65	12.66	12.52
	32msec	6.47	7.30	14.40	13.19

4.2 Real-room Recorded Data

Finally, we applied the algorithm to the data recorded in a real environment. The data was provided by Prof. Kota Takahashi in the University of Electro-Communications. Two males were repeating different phrases simultaneously in a room and their voices were recorded with two microphones with 44.1kHz for 5sec then down-sampled to 16kHz. Inputs are shown in Fig.4.a.

We applied our algorithm to this data. Window length was 32msec (512 points), ΔT was 1.25msec and r was 40. The result is shown in Fig.4.*b*. We show the separated signals in the graphs. We heard them and they were separated clearly.



a *b*

Figure 4: *a*: Recorded Signals in a Real Room *b*: Separated Signals

5 Conclusion

We proposed a BSS algorithm for speech signals. Our algorithm only uses straightforward calculation, and it includes only a few parameters to be tuned. Through the experiments, the algorithm worked very well for the data mixed on the computer and also for the real-room-recorded data. We haven't shown other results but they are available at

`http://www.islab.brain.riken.go.jp/~shiro/blindsep.html`

This algorithm is easy for hardware implementation, and we are now working for it. We are also working for realizing its on-line version. An on-line algorithm will make it possible to track walking speakers. In this article, we used the envelope of the signals to construct independent spectrograms form spectrograms form separated frequency components. It may be possible to use the continuity of de-mixing matrices between close frequency channels.

References

[1] J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, 1996.

[2] S. C. Douglas and A. Cichocki. Neural networks for blind decorrelation of signals. *IEEE Trans. Signal Processing*, 45(11):2829–2842, 1997.

[3] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. Sejnowski. Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem. In *Proceedings of ICASSP'98*, 1998.

[4] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72(23):3634–3637, 1994.

[5] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. In *International Workshop on Independence & Artificial Neural Networks*, University of La Laguna, Tenerife, Spain, 1998.

[6] A. Ziehe. Statistische verfahren zur signalquellentrennung. Master's thesis, Humboldt Universität, Berlin, 1998. (in German).