An Approach to Blind Source Separation
Based on Temporal Structure of Speech Signals

Noboru Murata
Lab. for Information Synthesis
RIKEN Brain Science Institute
mura@brain.riken.go.jp

Shiro Ikeda
PRESTO, Japan Science and Technology Corporation (JST)
shiro@brain.riken.go.jp

Andreas Ziehe
GMD FIRST, Berlin
ziehe@first.gmd.de


Abstract

In this paper we introduce a new technique for blind source separation of speech signals. We focus on the temporal structure of the signals in contrast to most other major approaches to this problem. The idea is to apply the decorrelation method proposed by Molgedey and Schuster in the time-frequency domain. We show some results of experiments with both artificially controlled data and speech data recorded in the real environment.
1 Introduction

In this paper, we consider a blind source separation method for temporally structured signals, in particular speech signals. Blind source separation of speech signals is often called the “cocktail party problem”. The name comes from the fact that we can hold a conversation at a cocktail party even though we are surrounded by loud voices and boisterous music. This mechanism must involve higher order functions in human brain, but it is also an interesting problem of signal processing. We try to solve this problem in the framework of “blind source separation” or “independent component analysis”, but one of the difficulties of separating speech signals is due to delays and reflections of the real environment, that is, those mixed signals are not instantaneous mixtures but convolutive mixtures.

To solve the problem, we adopt time-frequency representation, which is obtained by applying a windowed Fourier transform. With this representation, the problem is reduced to subproblems of non-convolutive mixtures and the subproblems can be solved with simple blind source separation algorithms. In this paper, we use an algorithm of eliminating cross-correlation functions [7,12,15]. In the time-frequency approach, another difficulty arises when we reconstruct the separated sources by combining the separated components of different frequencies, because each frequency component is independently separated and we have to determine from which source signal the specific frequency component comes. Speech signals have a temporal structure that can be regarded as stationary for short time-scales, although for longer time-scales, it is non-stationary. This non-stationarity comes mainly from amplitude modulation and slight change of frequency distribution of signals. We propose to use this temporal structure in order to reconstruct source signals from frequency components.

This paper is organized as follows: in section 2, we formulate the problem of blind source separation and explain frequency-time domain representation and the de-correlation method of non-convolutive mixtures, which are used in our approach. In section 3, we describe an algorithm for separating convolutive mixtures of speech signals and, in section 4 some results of our algorithm are shown. Finally, we give a brief summary and concluding remarks in section 5.

2 Blind Source Separation

2.1 Statement of problem

In this section, we formulate the problem of blind source separation.

Source signals are denoted by a vector

\[ s(t) = (s_1(t), \ldots, s_n(t))^T, \quad (1) \]

at discrete time \( t = 0, 1, 2, \ldots \) and each component of \( s(t) \) is assumed to be independent of each other, i.e. the joint density function of the signals is factorized by their marginal density
functions.

$$p(s_1(t), \cdots, s_n(t)) = p(s_1(t)) \times \cdots \times p(s_n(t)).$$  \hspace{1cm} (2)

Without loss of generality, we assume the source signal $s(t)$ to be zero mean. Observations are represented by

$$x(t) = (x_1(t), \cdots, x_n(t))^T,$$  \hspace{1cm} (3)

and they correspond to the recorded signals at sensors or microphones. Let $a_{ij}(\tau)$ be a unit impulse response from source $j$ to sensor $i$ with time delay $\tau$. The observation at sensor $i$ can be represented as

$$x_i(t) = \sum_{j=1}^{n} \sum_{\tau} a_{ij}(\tau)s_j(t - \tau)$$

$$= \sum_{j=1}^{n} a_{ij} \ast s_j(t) \hspace{1cm} (4)$$

where $\ast$ denotes the convolution. We write this relation in matrix form as

$$x(t) = A \ast s(t),$$  \hspace{1cm} (5)

where

$$A(t) = \begin{pmatrix}
a_{11}(t) & \cdots & a_{1n}(t) \\
\vdots & \ddots & \vdots \\
a_{n1}(t) & \cdots & a_{nn}(t)
\end{pmatrix},$$  \hspace{1cm} (6)

is called a filter matrix.

The goal of blind source separation is usually to find a filter $B(t)$, such that the components of reconstructed signals

$$y(t) = B \ast x(t)$$  \hspace{1cm} (7)

are mutually independent, without knowing filter $A(t)$ and the probability distribution of source signal $s(t)$ (see for examples [6,10,14]). Ideally we expect $B(t)$ to be the inverse filter of $A(t)$, but there remains indefiniteness of scaling factors and permutation because of lack of information about the amplitude and the order of the source signals, that is, a desired $B(t)$ satisfies

$$B \ast (A \ast s)(t) = PDs(t),$$  \hspace{1cm} (8)

where $D$ is a diagonal matrix which represents scaling factors, and $P$ is a permutation matrix, i.e. all the elements of each column and row are 0 except for one element with value 1. In order to avoid the scaling ambiguity, we moderate the problem to find a decomposition

$$x(t) = v_1(t) + v_2(t) + \cdots + v_n(t)$$  \hspace{1cm} (9)
instead of estimating source signals, such that $v_i(t)$'s are mutually independent. By using the filter $B(t)$ and its inverse filter $B^{-1}(t)$, observations are decomposed as

$$x(t) = B^{-1} * B * x(t)$$

$$= B^{-1} * IB * x(t)$$

$$= B^{-1} * (E_1 + \cdots + E_n) B * x(t)$$

$$= B^{-1} * E_1 B * x(t) + \cdots + B^{-1} * E_n B * x(t),$$  \hspace{1cm} (10)

where $I$ is the identity matrix, $E_i$ is a matrix with 1 for the $i$-th diagonal element and 0 for the other elements and satisfy $E_1 + \cdots + E_n = I$. Therefore by putting

$$v_i(t) = B^{-1} * (E_i B) * x(t),$$  \hspace{1cm} (11)

we obtain the desired decomposition. It is easy to check that the representation of $v_i(t)$ does not depend on the rescaling of $B(t)$ as follows. Let $D$ be an arbitrary non-singular diagonal matrix, then

$$(DB)^{-1} * (E_i(DB)) * x(t) = B^{-1} * (D^{-1} E_i DB) * x(t)$$

$$= B^{-1} * (E_i B) * x(t).$$

The physical meaning of each component $v_i(t)$ is a signal vector generated by one independent component which is observed on sensors.

Though there is still an ambiguity of the order of decomposed components, i.e. the index $i$ can be arbitrary, our goal of blind separation of speech signal is to obtain this decomposition.

### 2.2 Time-frequency representation

There is another representation of convolutive mixtures by using the windowed Fourier transform, which allows us to formulate the problem in time-frequency domain (cf. [16], see, for example, for the same approach in speech separation [14]). Let us define the discrete Fourier transform with moving windows as

$$\hat{f}(\omega, t_s) = \sum_t e^{-j\omega t} f(t) w(t - t_s),$$  \hspace{1cm} (12)

$$\omega = 0, \frac{1}{N} 2\pi, \ldots, \frac{N-1}{N} 2\pi, \quad t_s = 0, \Delta T, 2\Delta T, \ldots$$

where $\omega$ denotes the frequency, $N$ denotes the number of points in the discrete Fourier transform, $t_s$ denotes the window position, $w$ is a window function such as Hamming, Hanning or Kaiser, and $\Delta T$ is the shifting time of the moving windows. This type of representation is commonly used in speech processing and $\hat{f}(\omega, t_s)$ is often referred as spectrogram. The inversion of Equation (12) is

$$f(t) = \frac{1}{2\pi} \cdot \frac{1}{W(t)} \sum_{t_s} \sum_\omega e^{j\omega(t-t_s)} \hat{f}(\omega, t_s),$$  \hspace{1cm} (13)
where

\[ W(t) = \sum_{t_s} w(t - t_s). \]  

(14)

Applying the windowed Fourier transform, the relationship between observations and sources is approximated by

\[ \hat{x}(\omega, t_s) = \hat{A}(\omega) \hat{s}(\omega, t_s), \]  

where \( \hat{A}(\omega) \) is the Fourier transform of filter matrix \( A(t) \), and \( \hat{s}(\omega, t_s) \) is the windowed Fourier transform of source signals \( s(t) \). For fixed frequency \( \omega \) in Equation (15), \( \hat{x}(\omega, t_s) \) can be regarded as a non-convolutive or instantaneous mixture of complex-valued time series \( \hat{s}(\omega, t_s) \). Therefore, by adopting the time-frequency representation, the convolutive mixture problem is divided into subproblems of non-convolutive mixtures at frequency \( \omega \).

Note that in the above representation, the filter matrix is parameterized in the frequency domain and matrix \( \hat{A}(\omega) \) for one frequency can be estimated independently of other frequencies. On the other hand, the filter matrix, which is parameterized in time domain as Equation (6), is transformed into frequency domain as

\[ \hat{A}(\omega) = \sum_{\tau} A(\tau) e^{j\omega \tau}. \]  

(16)

This parameterization includes some sort of continuity condition across different frequencies, however it doesn’t allow us to decompose the problem into simple subproblems.

2.3 Eliminating Cross-correlation

In this section, we explain the method of separating non-convolutive mixture of source signals, which we use in the following experiments. There are some approaches of blind source separation problems, such as minimizing mutual information (e.g. [1, 2, 5]), eliminating cross-moments or cross-cumulants (e.g. [3, 8]), and de-correlating simultaneously in different time slices (e.g. [11]). In this paper we focus on the method of eliminating the cross-correlation of the reconstructed signals, because of the following two advantages of this method,

- it uses only the second order statistics, hence the estimation is generally robust,
- and there exists an algorithm which doesn’t includes iterative operations.

Let us assume that source signals are weakly stationary and observations are non-convolutive mixtures, i.e. \( A \) is a constant matrix. In this case, the relationship between sources and observations are simply written as

\[ x(t) = As(t). \]  

(17)
The correlation matrix of observations is
\[
\langle x(t)x(t+\tau)^* \rangle = A \langle s(t)s(t+\tau)^* \rangle A^*,
\]
\[
= A \begin{pmatrix} R_{s_1}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{s_n}(\tau) \end{pmatrix} A^*, \tag{18}
\]
where * denotes taking complex conjugate and transpose, \(\langle \cdot \rangle\) denotes taking the average, and \(R_{s_i}(\tau)\) is the auto-correlation function of source signal \(s_i(t)\). With a desired matrix \(B\), the reconstructed signals are represented by
\[
y(t) = Bx(t) = BAs(t) = PDs(t), \tag{19}
\]
and the correlation matrix of the reconstructed signals becomes
\[
\langle y(t)y(t+\tau)^* \rangle = \langle (PDs(t))(PDs(t+\tau))^* \rangle
\]
\[
= \begin{pmatrix} |\lambda_1'|^2 R_{s_1'}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & |\lambda_n'|^2 R_{s_n'}(\tau) \end{pmatrix} \tag{20}
\]
where \(1', 2', \ldots, n'\) denotes a permutation of the indices \(1, 2, \ldots, n\) determined by matrix \(P\), and \(\lambda_i\) is the \(i\)-th diagonal element of matrix \(D\). Hence, except for the ambiguity of permutation \(P\) and scaling \(D\), an optimal \(B\) can be characterized as a matrix that diagonalizes the correlation matrices at any time difference \(\tau\).

This concept is simplified as the simultaneous diagonalization of the correlation matrix of observations at several time delays \([7, 12, 15]\), i.e. the objective is to find \(B\) in a certain class of matrices, which satisfies
\[
B\langle x(t)x(t+\tau_i)^* \rangle B^* = A_i, \quad i = 1, \ldots, r, \tag{21}
\]
where \(A_i\)'s are diagonal matrices.

Some algorithms are proposed for solving Equation (21). For instance, in [12], Molgedey and Schuster mentioned an algorithm solving the eigenvalue problem of the two correlation matrices
\[
M_1^{-1}M_2 B^* = B^* A_1^{-1} A_2, \tag{22}
\]
where
\[
M_k = \frac{1}{T} \sum_{t=0}^{T-1} x(t)x(t+\tau_k)^*, \quad k = 1, 2
\]
however, practically this procedure is sensitive to the estimation error of the correlation matrices. Also they proposed an algorithm minimizing the error function which consists of several time delayed correlation matrices

\[
L(B) = \sum_{i=1}^{r} \sum_{j \neq k} \left| (B \langle x(t)x(t+\tau_i) \rangle B^*)_{jk} \right|^2,
\]

with a certain regularization for matrix \( B \) such as \( |\text{det}B| = 1 \), where \((B \langle x(t)x(t+\tau_i) \rangle B^*)_{jk}\) denotes the \(jk\)-element of the matrix \( B \langle x(t)x(t+\tau_i) \rangle B^* \). This is solved by a gradient-based iterative method.

In this paper, we use a non-iterative method [17, 18]. The algorithm consists of two stages, “sphering” and “rotation” (see Figure 1).

Sphering (or pre-whitening) is an operation for orthogonalizing the source signals in a new coordinate. Let us define a decomposition of a covariance matrix of observations

\[
V = SAS^*,
\]

where \( V \) is a covariance matrix

\[
V = \frac{1}{T} \sum_{t=0}^{T-1} x(t)x(t)^*,
\]

where \( S \) and \( A \) are a unitary matrix and a diagonal matrix respectively, and define a square root inverse of the covariance matrix with

\[
\sqrt{V^{-1}} = \sqrt{A^{-1}}S^*,
\]

where \( \sqrt{A^{-1}} \) denotes a diagonal matrix, each of whose elements is the square root of \( A^{-1} \)'s corresponding diagonal element, i.e.,

\[
\sqrt{A^{-1}} = \begin{pmatrix} \sqrt{\lambda_1^{-1}} & & \\ & \ddots & \\ & & \sqrt{\lambda_n^{-1}} \end{pmatrix} \text{ for } A = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}
\]

Note that from weak stationarity, the covariance matrix is time independent. By transforming the observation vector with

\[
x'(t) = \sqrt{V^{-1}}x(t),
\]

the covariance matrix of the new vector \( x'(t) \) is diagonalized,

\[
\frac{1}{T} \sum_{t=0}^{T-1} x'(t)x'(t)^* = \sqrt{V^{-1}}V\sqrt{V^{-1}} = I.
\]
Intuitively speaking, source signals are distributed over certain directions in the coordinate based on the observation, and in general the directions are not orthogonal. Sphering rearranges these directions to be orthogonal to each other in the new coordinate (Figure 1). After applying transform (27), however, there remains an ambiguity of rotation, because the decomposition of Hermite matrix which satisfies

\[ V = \sqrt{V^{-1}} * \sqrt{V^{-1}} \]

is not uniquely determined and definition (26) is one of the decomposition. It is easily seen that with any unitary matrix \( C \), rotated signals \( Cx'(t) \) doesn’t change the covariance matrix,

\[
\frac{1}{T} \sum_{t=0}^{T-1} (Cx'(t))(Cx'(t))^* = C \left( \frac{1}{T} \sum_{t=0}^{T-1} x'(t)x'(t)^* \right) C^* \\
= CIC^* \\
= I.
\]

The correct rotation is determined by removing the off-diagonal elements of the correlation matrix at several time delays. A possible implementation is to find a unitary matrix \( C \) which minimizes

\[
\sum_{k=1}^{r} \sum_{i \neq j} |(CM_kC^*)_{ij}|^2 ,
\]

where \((CM_kC^*)_{ij}\) denotes the \(ij\)-element of matrix \(CM_kC^*\) and

\[
M_k = \frac{1}{T} \sum_{t=0}^{T-1} x'(t)x'(t+\tau_k)^* \quad k = 1, \ldots, r.
\]

To solve this approximate simultaneous diagonalization problem, we use a Jacobi-like algorithm proposed by Cardoso and Souloumiac [4]. It is an extension of Givens unitary rotation transform and the problem is reduced to combination of subproblems of the \(2 \times 2\) case that can be solved analytically (see [4] for more details, and see [3] for its application).

With these two operations, matrix \( B \) is given by

\[
B = C\sqrt{V^{-1}}.
\]

As mentioned before, one of the advantages of this method is that it uses only the second order statistics, therefore it is less sensitive to out-liers than the other methods which use higher order statistics. In derivation of this method, it is assumed that the sources are weakly stationary signals with different auto-correlation functions, however, this method is applicable to non-stationary signals if the non-stationarity is not strong. A required condition is that the averaged correlation functions can be well approximated by the observations

\[
\frac{1}{T} \sum_{t=0}^{T-1} \langle x'(t)x'(t+\tau_k)^* \rangle \sim \frac{1}{T} \sum_{t=0}^{T-1} x'(t)x'(t+\tau_k)^*
\]

for appropriately chosen \( T \) and \( \tau_k \).
3 Proposed Method on Convolutive Mixtures

3.1 Properties of speech signals

In this section, we propose an algorithm for blind source separation for convolutive mixtures of time-structured signals, such as speech signals. We first transform the observations to spectrograms in order to handle them in the time-frequency domain, then apply blind source separation for each frequency, and finally reconstruct the separated signals from the separated spectrograms. The properties of speech signals which we focus on are:

- signals are supposed to be stationary within a short time-scale,
- and signals are intrinsically non-stationary in a long range mainly because of amplitude modulation.

It is said that the human voice is stationary for a period shorter than a few 10msecs [9]. If it is longer than a few 10msecs and around 100msec, the frequency components of the speech will change its structure. This means that speech signals must be dealt with differently in microscopic and macroscopic viewpoints.

The first property allows us to apply the windowed Fourier transform at a short time range and the length of the time window should be determined with the knowledge of microscopic stationarity. The second property is used to combine the decomposed frequency components properly in our approach. As mentioned in previous section, there is an ambiguity of determining the order of the separated signals in blind source separation algorithms, and we apply an algorithm to every frequency independently, therefore we have to carefully choose frequency components when we recompile them. As well known, when the signal is stationary, any Fourier components at different frequencies are uncorrelated, therefore it is impossible to find an appropriate combination. On the other hand, based on the non-stationarity, i.e. the second property of the speech signals, it is natural to assume that components at different frequencies from the same source signals are under the influence of a similar modulation in amplitude. Let us rewrite $\hat{s}_i(\omega, t_s)$ with radius and phase as

$$\hat{s}_i(\omega, t_s) = a_i(\omega, t_s)e^{j\phi_i(\omega, t_s)}.$$  \hspace{1cm} (33)

Because of non-stationarity, radius $a_i(\omega, t_s)$ changes in time, and it corresponds to the envelop of signal $\hat{s}_i(\omega, t_s)$. As $\hat{s}_i(\omega, t_s)$ and $\hat{s}_j(\omega, t_s)$ are independent, the correlation between envelopes $a_i(\omega, t_s)$ and $a_j(\omega, t_s)$ vanishes,

$$\text{corr}(a_i(\omega, t_s), a_j(\omega, t_s)) = \frac{1}{T} \sum_{s=1}^{T} a_i(\omega, t_s)a_j(\omega, t_s) - \frac{1}{T} \sum_{s=1}^{T} a_i(\omega, t_s) \cdot \frac{1}{T} \sum_{s=1}^{T} a_j(\omega, t_s)$$

$$= 0, \quad i \neq j,$$
if $T$ is sufficiently large. Similarly, correlation between different frequency components from
different source signals also vanishes
\[
\text{corr}(a_i(\omega, t_s), a_j(\omega', t_s)) = 0, \quad i \neq j, \quad \omega \neq \omega'.
\]
However, for different frequency components from the same source signal, we can assume
\[
\text{corr}(a_i(\omega, t_s), a_i(\omega', t_s)) \neq 0,
\]
from the second property. Intuitively speaking, frequency components of speech signals won’t change the distributions drastically in time, but they are similarly affected by the amplitude modulation of the vocal chords. Therefore, the correlation coefficient of their envelopes
\[
r(a_i(\omega, t_s), a_j(\omega', t_s)) = \frac{\text{corr}(a_i(\omega, t_s), a_j(\omega', t_s))}{\sqrt{\text{corr}(a_i(\omega, t_s), a_i(\omega, t_s))\text{corr}(a_j(\omega', t_s), a_j(\omega', t_s))}}
\]
would be a natural measure for estimating appropriate combination of frequency elements.

### 3.2 Description of algorithm

The followings are whole the procedures of our algorithm.

**windowed Fourier transform** we apply the Fourier transform with moving windows (see Figure 2). The size of the window should be shorter than stationary duration of the speech signals. In our experiments, we mainly adopt 32msec (512 taps with 16kHz sampling) Hamming window. Then we obtain spectrogram representation $\hat{x}(\omega, t_s)$.

**blind separation at each frequency** We apply the algorithm explained in section 2.3 upon spectrogram $\hat{x}(\omega, t_s)$ at each frequency, then we obtain estimation of de-mixing matrices $\hat{B}(\omega)$, which give independent components
\[
\hat{u}(\omega, t_s) = \hat{B}(\omega)\hat{x}(\omega, t_s).
\]

Note that any blind source separation algorithm for non-convolutive mixtures of real-valued signals can be applied with small extension. Usually this can be easily done by substituting a Hermite matrix and a unitary matrix for a symmetric matrix and an orthogonal matrix respectively.

**decomposition of spectrograms** The decomposition of spectrograms is performed by
\[
\hat{v}(\omega, t_s; i) = B(\omega)^{-1}E_i B(\omega)\hat{x}(\omega, t_s) = B(\omega)^{-1} \begin{pmatrix} 0 \\ \vdots \\ \hat{u}_i(\omega, t_s) \\ \vdots \\ 0 \end{pmatrix},
\]
where \( \hat{u}_i(\omega, t_s) \) denotes the \( i \)-th element (the \( i \)-th independent component) of \( \hat{u}(\omega, t_s) \).

Note that \( i \) is a function of the frequency \( \omega \) implicitly, i.e. \( i = i(\omega) \).

**clustering frequency components**  Let us define a moving average operator \( \mathcal{E} \) for estimating the envelope of time series by

\[
\mathcal{E} \hat{v}(\omega, t_s; i) = \frac{1}{2M + 1} \sum_{t'_s = t_s - M}^{t_s + M} \sum_{j=1}^{n} |\hat{v}_j(\omega, t'_s; i)|,
\]

where \( M \) is a positive constant and \( \hat{v}_j(\omega, t_s; i) \) denotes the \( j \)-th element of \( \hat{v}(\omega, t_s; i) \).

We solve the permutation by sorting based on the correlation of envelopes as follows (see Figure 3),

1. sort \( \omega \) in order of low correlation between independent components in \( \omega \). This is done by sorting in increasing order of similarity defined by

\[
sim(\omega) = \sum_{i \neq j} r(\mathcal{E} \hat{v}(\omega, t_s; i), \mathcal{E} \hat{v}(\omega, t_s; j)),
\]

\[
sim(\omega_1) \leq \sim(\omega_2) \leq \cdots \leq \sim(\omega_N).
\]

2. for \( \omega_1 \), assign \( \hat{v}(\omega_1, t_s; i) \) to \( \hat{y}(\omega_1, t_s; i) \) as it is,

\[
\hat{y}(\omega_1, t_s; i) = \hat{v}(\omega_1, t_s; i), \quad i = 1, \ldots, n
\]

3. for \( \omega_k \), find a permutation \( \sigma(i) \) which maximizes the correlation between the envelope of \( \omega_k \) and the aggregated envelope from \( \omega_1 \) through \( \omega_{k-1} \). This is achieved by maximizing sum of correlation coefficients

\[
\sum_{i=1}^{n} r(\mathcal{E} \hat{v}(\omega_k, t_s, \sigma(i)), \sum_{j=1}^{k-1} \mathcal{E} \hat{y}(\omega_j, t_s; i))
\]

within all the possible permutations of \( i = 1, \ldots, n \).

4. assign the appropriate permutation to \( \hat{y}(\omega_k, t_s; i) \):

\[
\hat{y}(\omega_k, t_s; i) = \hat{v}(\omega_k, t_s; \sigma(i)), \quad i = 1, \ldots, n.
\]

5. go to 3 until \( k = N \).

As a result, we obtain separated spectrograms

\[
\hat{y}(\omega, t_s; i), \quad i = 1, \ldots, n.
\]
Applying the inverse Fourier transform defined by Equation (13) to the separated spectrograms \( \hat{y}(\omega, t_s; i) \), we finally obtain a set of separated signals,

\[
y(t; i) = \frac{1}{2\pi} \cdot \frac{1}{W(t)} \sum_{t_s} \sum_{\omega} e^{j\omega (t - t_s)} \hat{y}(\omega, t_s; i),
\]

(44)

\[
\sum_{i=1}^{n} y(t; i) = x(t).
\]

4 Experimental Results

In this section, we show some results of the proposed algorithm. First, the sources are mixed on the computer and our algorithm was applied to those mixed data. Since the true sources were available, we can evaluate the performance of the algorithm. Also another result with the data recorded in the real environment will be shown. In this case, we cannot know the true sources. We show the result with graphs. The data is available on our web page:


4.1 Artificial Data

4.1.1 Separating instantaneous mixtures with the basic de-correlation approach

In this subsection, we show the result of an experiment on a set of data which was mixed instantaneous on a computer. Figure 4 shows the sources which were recorded separately on the computer.

We mixed these source signals without delay as in Equation (17) where the matrix \( A \) is shown below.

\[
x(t) = As(t) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} s(t) = \begin{pmatrix} 1 & 0.7 \\ 0.3 & 1 \end{pmatrix} s(t)
\]

Since \( x(t) \) is not a convolutive mixture, we can apply the original technique in Subsection 2.3. We used 30 matrices for simultaneous diagonalization in the experiment. Since we know the true sources and the mixing rates, we can evaluate the performance using the SNR (Signal to Noise Ratio) which is defined as

\[
\text{SNR}_{ij} = 10 \log_{10} \frac{\sum_t \text{signal}_i(t; j)^2}{\sum_t \text{error}_i(t; j)^2}.
\]

SNR\(_{ij}\) for this experiment is shown in Table 1. Every SNR\(_{ij}\) is more than 27.0 which means crosstalk is less than 1/500.
4.1.2 Separating instantaneous mixtures with the proposed method

In this subsection, our algorithm described in section 3 was applied to the same problem in the last subsection.

In order to use our algorithm, we have to first define some parameters. Our algorithm needs to apply the windowed Fourier transform which was defined in Equation (12). There are two parameters for a windowed Fourier transform, one is the window length and the other is the shifting time $\Delta T$. We also have another parameter $r$ in Equation (30) which is the number of matrices to be diagonalized simultaneously. We made some preliminary experiments to set these parameters.

From some trials, we found that $\Delta T$ and $r$ are strongly related, but that window length is relatively independent of these values. The SNR$_{ij}$’s are measured by changing the window length from 4msec to 32msec and $\Delta T$ from 0.0625msec to 2.5msec. We used the Hamming window for the window function. Results of SNR$_{11}$ is shown in Figure 5. It is clear from the graph that the window length with 8msec gave results better than the others and we confirmed this fact for other $ij$ and $r$ with experiments not shown here. Therefore, we defined window length as 8msec for this experiment. Signals were all recorded with a sampling rate of 16kHz and 8msec corresponds to 128 points of samples.

This set the window length, but we still had to define $\Delta T$ and $r$. Theoretically, $r$ can be 2 or any larger number. However, small $r$ gives an unstable solution, and large $r$ leads to a wrong solution because time difference between correlation matrices will be too larger for the stationarity of speech signals. We changed $\Delta T$ and $r$ and calculated SNR$_{ij}$. Figure 6 shows the result of SNR$_{11}$ with changing $\Delta T$ from 0.0625msec to 2.5msec and $r$ from 2 to 70. From this graph, we see that there is a peak on each row. To see this more clearly, we replot SNR$_{11}$ versus $\Delta T \times r$. $\Delta T \times r$ is the interval of time within which the matrices are diagonalized. This value should not go beyond the stationarity of the speech signals. The result is shown in Figure 7. We can see there is a peak between 30 and 50msec. It is said that speech signals are stationary around 40msec, and this matches the result we obtained here. This feature is also true for other $ij$’s. We can see that the combination of $\Delta T = 1.25$msec and $r = 40$ is the best. The position of the peaks are almost the same for other $ij$’s, and we decided to use these values for $\Delta T$ and $r$. SNR$_{11}$ for this combination is 18.9 (crosstalk is 1/77.6).

The parameter $M$ in Equation (37) is used to make the moving average of signals to make the envelopes. We used $\Delta T \times (2M + 1)$ to be around 40msec. We defined $M$ as 15, because $\Delta T$ was defined as 1.25msec.

Finally, we show the separated signals in Figure 8, and SNR’s in Table 2. Crosstalk is small and it is hard to see them in the graph.
4.1.3 Separating convolutive mixtures

Our main aim of this paper is not to separate instantaneous mixtures, but to separate convolutive mixtures. We also made convolutive mixture signals on the computer and used these signals for experiment to assess how our algorithm works. As in Equation (5), a convolutive mixture is defined as,

\[ x(t) = A * s(t) = \left( \sum_j a_{1j} * s_j(t) \right) \]
\[ a_{ij} * s_j(t) = \sum_{\tau=0}^{\infty} a_{ij}(\tau)s_j(t - \tau). \]

\( a_{ij} * s_j(t) \) is the convolution of \( a_{ij}(t) \) and \( s_j(t) \).

We wanted to simulate the general problem of recording sounds in a real environment. When sound signals are recorded, the major factors causing convolutions are reflections and delays. In order to simulate these factors, we built a virtual room as Figure 9 and calculated reflections and delays.

We supposed that each wall, floor and ceiling reflects the sound. The strength of the reflection is 0.1 in power for any frequency. We also supposed the strength of the sounds varies in proportion to the inverse square of the distance. Because the second reflection of a sound is really small, we only counted the first reflection. In Figure 10, the impulse response from source 1 to microphone 2 is shown. Also we show the window function with different lengths in the graph.

Apparently, the impulse response is rather long and if the window length is 8msec, all the reflections within one window cannot be included. If all the reflections are not included within a window, our new approach won’t work well. Hence we have to set the window length longer than the impulse response. But as shown in Subsection 4.1.2, if we make the window length long, the SNR will be worse. There is a trade-off between the window length and SNR for the convolutive mixture.

The source signals are the same as Figure 4. The convolutive mixtures in this virtual room are shown in Figure 11.

For the separation, we first applied the original de-correlation algorithm. Of course it didn’t work for a convolutive mixture. We also applied our algorithm, changing the window length from 8msec to 32msec. The SNRs of these results are shown in Table 3. Our approach with the window length of 32msec gave the best SNRs. Separated signals are shown in Figure 12(window length was 32msec, \( \Delta T \) was 1.25msec, \( r \) was 40.).

4.2 Real-room Recorded Data

In this subsection, we will show a result of our algorithm applied to data recorded in a real environment. This data was recorded by Dr. Te-Won Lee in Salk Institute. In this recorded
data, a speaker has been recorded counting digits from one to ten with two microphones (sampling rate 16kHz) in a normal office room with music in the background. The distance between the speaker, cassette player and the microphones is about 60cm in a square ordering. Inputs to the microphones were shown in Figure 13.

We applied our algorithm to this data. The parameters are set as in the last subsection. Window length was 32msec (512 points), $\Delta T$ was 1.25msec and $r$ was 40. The result is shown in Figure 14.

In this experiment, we don’t know the source signals precisely, and we cannot calculate the SNRs. The only way to evaluate the performance is to see the graphs and to listen to the results. These two signals seem to be independent in the graphs. We listened to them and they were separated clearly.

5 Conclusion

We proposed a blind source separation algorithm based on the temporal structure of speech signals. Our algorithm has a feature that it only uses straightforward calculations, and it includes only a few parameters to be tuned. On the experiments, the algorithm worked well for the data mixed on the computer and also for the real-room-recorded data. We haven’t shown other results but we have also applied our algorithm to other data, and they are available at the URL which we showed in section 4.

There are still some problems to be solved in our algorithm. We have three major parameters, the window length, $\Delta T$ and $r$. We showed experimentally that window length can be defined independent of the other two parameters, but the window length has a strong relation to the impulse response of the mixing process. If the mixing process has a long impulse response, window length has to be correspondingly longer, but it will make the performance worse because it will go beyond the stationary range of the source signals. In order to choose good parameters, it is important to know more about the statistics of the speech signals and other sources (music or noises) and also the environment (reflections of floor and walls, size of the room). We are now trying to know more on those statistics by recording real voices in real environment, but those are one of our future works. Also the sampling rate involves the performance of the algorithm. In our experiments, we only used a sampling rate of 16kHz. From the sampling theorem, it follows that the data includes signals whose frequency component is below 8kHz. Usually, speech signals have some power for every component under 8kHz. Our algorithm applies the de-correlation algorithm for every frequency component, but if even one component doesn’t have any power, the de-correlation algorithm fails. Therefore, if we use 44.1kHz for the sampling rate of speech.

\footnote{For the detail of the signals, please visit http://www.cnl.salk.edu/ tewon/}
signals, there will be a lot of components which cannot be separated correctly. We need some other technique to solve this problem.

Another problem is the computational cost. Since separation is done in each frequency channel, the complexity of calculation is proportional to the number of the points of the discrete Fourier transform, but computational complexity increases when the number of sources increases. As separation based on extended Givens rotation transform and solving permutation based on pair-wise correlation are calculation between two independent components, for \( n \) sources the computational cost is evaluated as \( O\left( \frac{n(n-1)}{2} \right) = O(n^2) \). The improvement of the computational cost remains as a future work.

Finally, we would like to say this is the first step of this approach. The algorithm is easy for hardware implementation and we are now working for it. We are also working on realizing its on-line version [13]. An on-line algorithm will make it possible to follow the changing environment, such as tracking walking speakers. Moreover our algorithm is based on the windowed Fourier transform, but filter banks and Wavelet transforms are also applicable, and we are investigating the possibility of these modifications. Our method presented here is based on the strict independence of source signals, but also extension for correlated source signals are needed for more realistic situations. In this case, it may be helpful to use the continuity of de-mixing matrices between close frequency channels, but this is also one of the problems which have not been solved.

Acknowledgment

We thank Shun-ichi Amari, Klaus-Robert Müller, Andrew Cichocki, and Colin Fyfe for very useful discussions on this work. We are grateful to Te-Won Lee for a set of data and comments on the manuscript. We also thank Kota Takahashi, Paris Smaragdis, and Scott C. Douglas for comments on our work and demonstrations which can be found in the web contents. We appreciate Daishi Harada for thorough proofreading our manuscript. A.Z. was partly funded by DFG contracts JA 379/51 and 379/71.

References


List of Figures

1 Decorrelation method using temporal structure of the signals ........................................ 20
2 The windowed Fourier transformation (We use notations $\hat{x}_t(\omega)$ and $\hat{x}_\omega(t_s)$ in the figure, to explicitly indicate that they are functions of $\omega$ and $t_s$ respectively.) ........ 20
3 Solving permutation problem based on the correlation of the envelopes ............................ 21
4 The source signals: each signal was spoken by a different male and recorded with sampling rate of 16kHz. $s_1(t)$ is a recorded word of “good morning” and $s_2(t)$ is a Japanese word “konbanwa” which means “good evening”. .................................................. 22
5 The SNR for instantaneous mixtures: the window lengths and $\Delta T$ were varied, $r$ (the number of the matrices for simultaneous diagonalization) was fixed to 30. .................. 22
6 The SNR for instantaneous mixtures: the window lengths was fixed to 8msec, $\Delta T$ and $r$ were varied. ................................................................. 23
7 The SNR: data in Figure 6 were replot against $\Delta T \times r$. ........................................... 23
8 The separated signals using the proposed algorithm: the window length was 8msec, $\Delta T = 1.25$msec and $r = 40$. ................................................................. 24
9 Virtual room for making convolutive mixtures: the length in the figure is m, and the sonic speed is 340m/sec. The strength of the reflection is 0.1 in power for any frequency, and the strength of sounds varies in proportion to the inverse square of the distance. We only counted first reflection. .................................................. 25
10 The impulse response from the source 1 to the microphone 2 in a virtual room .................. 25
11 The mixed signals in a virtual room .............................................................................. 26
12 The separated signals: the proposed algorithm was applied with the window length of 32msec, $\Delta T = 1.25$msec and $r = 40$. ................................................................. 26
13 The recorded signals in a real room: a speaker was counting digits from one to ten with a music in the background. ................................................................. 27
14 The separated signals: the proposed algorithm was applied with the window length of 32msec, $\Delta T = 1.25$msec and $r = 40$. ................................................................. 28
List of Tables

1. The SNRs (dB) for linear mixture using the basic decorrelation algorithm: 30 matrices were used for simultaneous diagonalization. .................................................. 29
2. The SNRs (dB) for linear mixture using the proposed algorithm: the window length was 8msec, \(\Delta T = 1.25\text{msec}\) and \(r = 40\). ................................. 29
3. The SNRs (dB) for convolutive mixtures in a virtual room: \(\Delta T = 1.25\text{msec}\) and \(r = 40\). .......................................................... 29
Figure 1: Decorrelation method using temporal structure of the signals

Figure 2: The windowed Fourier transformation (We use notations $\hat{x}_{t_s}(\omega)$ and $\hat{x}_{\omega}(t_s)$ in the figure, to explicitly indicate that they are functions of $\omega$ and $t_s$ respectively.)
Figure 3: Solving permutation problem based on the correlation of the envelopes
Figure 4: The source signals: each signal was spoken by a different male and recorded with sampling rate of 16kHz. $s_1(t)$ is a recorded word of “good morning” and $s_2(t)$ is a Japanese word “konbanwa” which means “good evening”.

Figure 5: The SNR for instantaneous mixtures: the window lengths and $\Delta T$ were varied, $r$ (the number of the matrices for simultaneous diagonalization) was fixed to 30.
Figure 6: The SNR\textsubscript{11} for instantaneous mixtures: the window lengths was fixed to 8msec, $\Delta T$ and $r$ were varied.

Figure 7: The SNR\textsubscript{11}: data in Figure 6 were replot against $\Delta T \times r$. 

23
Figure 8: The separated signals using the proposed algorithm: the window length was 8msec, $\Delta T = 1.25\text{msec}$ and $r = 40$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure8}
\caption{The separated signals using the proposed algorithm: the window length was 8msec, $\Delta T = 1.25\text{msec}$ and $r = 40$.}
\end{figure}
Figure 9: Virtual room for making convolutive mixtures: the length in the figure is m, and the sonic speed is 340m/sec. The strength of the reflection is 0.1 in power for any frequency, and the strength of sounds varies in proportion to the inverse square of the distance. We only counted first reflection.

Figure 10: The impulse response from the source 1 to the microphone 2 in a virtual room
Figure 11: The mixed signals in a virtual room

Figure 12: The separated signals: the proposed algorithm was applied with the window length of 32msec, $\Delta T = 1.25$msec and $r = 40$. 

26
Figure 13: The recorded signals in a real room: a speaker was counting digits from one to ten with a music in the background.
The separated source of counting digits.

The separated source of background music.

Figure 14: The separated signals: the proposed algorithm was applied with the window length of 32msec, $\Delta T = 1.25\text{msec}$ and $r = 40$. 
Table 1: The SNRs (dB) for linear mixture using the basic decorrelation algorithm: 30 matrices were used for simultaneous diagonalization.

<table>
<thead>
<tr>
<th>SNR_{11}</th>
<th>SNR_{12}</th>
<th>SNR_{21}</th>
<th>SNR_{22}</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.98</td>
<td>27.45</td>
<td>41.54</td>
<td>41.56</td>
</tr>
</tbody>
</table>

Table 2: The SNRs (dB) for linear mixture using the proposed algorithm: the window length was 8msec, $\Delta T = 1.25$ msec and $r = 40$.

<table>
<thead>
<tr>
<th>SNR_{11}</th>
<th>SNR_{12}</th>
<th>SNR_{21}</th>
<th>SNR_{22}</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.90</td>
<td>11.13</td>
<td>19.71</td>
<td>21.14</td>
</tr>
</tbody>
</table>

Table 3: The SNRs (dB) for convolutive mixtures in a virtual: $\Delta T = 1.25$ msec and $r = 40$.

<table>
<thead>
<tr>
<th></th>
<th>SNR_{11}</th>
<th>SNR_{12}</th>
<th>SNR_{21}</th>
<th>SNR_{22}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original de-correlation</td>
<td>-1.87</td>
<td>-1.22</td>
<td>2.93</td>
<td>2.74</td>
</tr>
<tr>
<td>Our approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(window length)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8msec</td>
<td>4.36</td>
<td>6.32</td>
<td>11.94</td>
<td>11.66</td>
</tr>
<tr>
<td>16msec</td>
<td>4.72</td>
<td>6.65</td>
<td>12.66</td>
<td>12.52</td>
</tr>
<tr>
<td>32msec</td>
<td>6.47</td>
<td>7.30</td>
<td>14.40</td>
<td>13.19</td>
</tr>
</tbody>
</table>