COMMUN. MATH. SCI.
Vol. 7, No. 1, pp. 109–128

A TIME DOMAIN ALGORITHM FOR BLIND SEPARATION OF CONVOLUTIVE SOUND MIXTURES AND L_1 CONSTRAINED MINIMIZATION OF CROSS CORRELATIONS*

JIE LIU†, JACK XIN‡, YINGYONG QI§, AND FAN-GANG ZENG¶

Abstract. A time domain blind source separation algorithm of convolutive sound mixtures is studied based on a compact partial inversion formula in closed form. An l_1-constrained minimization problem is formulated to find demixing filter coefficients for source separation while capturing scaling invariance and sparseness of solutions. The minimization aims to reduce (lagged) cross correlations of the mixture signals, which are modeled stochastically. The problem is non-convex, however it is put in a nonlinear least squares form where the robust and convergent Levenberg-Marquardt iterative method is applicable to compute local minimizers. Efficiency is achieved in recovering lower dimensional demixing filter solutions than the physical ones. Computations on recorded and synthetic mixtures show satisfactory performance, and are compared with other iterative methods.

Key words. Convolutive mixtures, compact partial inversion, l_1 constrained decorrelation, blind source separation.

AMS subject classifications. 94A12, 65H10, 65C60.

1. Introduction

Humans with normal ears are able to pay attention to one speaker while others are talking at the same time, as typically happens at a cocktail party. This is an example of our brain's blind signal processing (BSP) capability. Here blindness refers to the estimation of sources from received signals without detailed knowledge of the transmission environment (room shapes, speaker locations, furnitures etc). A simplified problem is this. Suppose a person is talking in a room while some music is playing in the background. Two microphones are placed at different locations in the room for recording. Due to multi-pathing effects of sound wave propagation, each recorded signal is a *convolutive* mixture of the speech and the music signals. The two plots on the left of Fig. 1.1 show the recorded data at sampling rate 16000 Hz, which could be what a listener picks up if the two microphones are replaced by two ears. What to be discussed in this paper is how to recover the speech and music signals (the original source signals in general) by "inverting the received mixtures". The two plots on the right are the recovered (separated) speech and music signals obtained by the algorithm discussed later in this paper. The main part of the algorithm is to determine the *demixing filter coefficients*, which then convolute with the recorded mixtures to produce the estimated source signals. We will derive the system of equations satisfied by the demixing fitler coefficients. The *solution* of the "source separation" problem hereafter refers to the demixing filter coefficients.

Suppose signal transmission is modeled by the linear relation $x = As$, where $s \in R^n$ is the sources, x is the received data, and A is an invertible $n \times n$ transfer matrix. A standard inversion problem is to find s (the sources), given x and A. A standard system identification problem is to approximate A, given x and s. The BSP problem

*Received: July 3, 2008; accepted (in revised version): November 30, 2008. Communicated by Lenya Ryzhik.

†Department of Mathematics, UC Irvine, Irvine, CA 92697, USA (liuj@math.uci.edu).

‡Department of Mathematics, UC Irvine, Irvine, CA 92697, USA (jxin@math.uci.edu).

§Department of Mathematics, UC Irvine, Irvine, CA 92697, USA (yingyong_qi@yahoo.com). And School of Information Engineering, Shangdong University at Weihai, China.

¶Department of Biomedical Engineering, UC Irvine, Irvine, CA 92697, USA (fzeng@uci.edu).

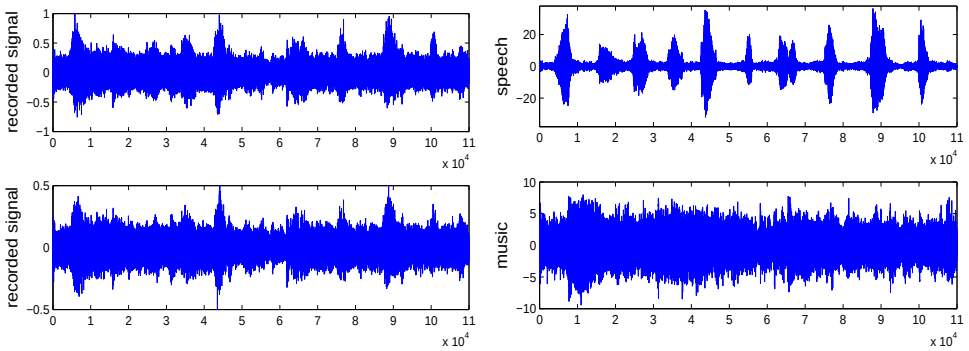


FIG. 1.1. *Left: recorded signals (input). Right: separated speech and music after processing (output). This is case(1)-1 in Sec. 4.*

is to estimate both s and A from x under certain assumptions of s. For example, if s and x are time dependent and A is time independent, then A can be estimated if different components of s are orthogonal time series or statistically independent when s is modeled as a random process [11]. The estimation is done without knowing s. Here, for simplicity we suppose that the data transmission has no time delay (direct-pathing only). The resulting BSP problem is called instantaneous.

A general BSP problem is to estimate an inverse system and source signals from observed data of a nonlinear dynamical system and certain a-priori properties (yet no access) of source signals or systems (Sec. 1.1 of [11]). The estimated source signals are usually a modified yet perceptually close version of the exact sources. BSP methods are widely used in biomedical engineering, medical imaging, communication systems, exploration seismology, geophysics, econometrics, unsupervised learning, and data mining [11], to name just a few.

Motivated by a human's ability to solve the cocktail party problem yet without knowing how the brain actually does it, blind source separation (BSS) methods aim to extract the original source signals from their mixtures based on the statistical independence of the source signals without knowledge of the mixing environment. The approach has been very successful for instantaneous mixtures [1, 7, 8]. However, realistic sound signals are unknown weighted sums of the signals and their delays. Separating such "convolutive" mixtures is a challenging problem, especially in realistic settings.

Time domain methods do not use Fourier transforms and have certain advantages. Phase and permutation indeterminacies commonly associated with frequency domain (Fourier transform based) methods are avoided. However, there are typically more parameters to estimate because of the nonlocality of convolutive mixtures, resulting in a high dimensional optimization problem. Also, the iterative solution methods may suffer from slow convergence, instability, and lack of robustness. Among the previously studied time domain methods, [29] introduced a feedback architecture and infomax learning rule (see [14] for recent development), and [21] proposed a feed-forward architecture and minimization of a logarithmic cost function (decorrelation) with preprocessing by a frequency domain method. As pointed out recently [10], to date time domain methods for convolutive mixtures are limited, and may converge slowly for statistically colored input signals such as speech.

In this paper, we formulate a time domain BSS method in an optimization framework to compute the demixing filter coefficients based on an exact partial inversion formula (2.7)–(2.8). We shall consider the case when the number of receivers is equal to the number N of sources. The computation is for $N=2$, where generally *a foreground signal is to be separated from a background signal*. The background signal may be a sound mixture with broad spectra. Though theoretically our framework is for any number of sources, we focus on $N=2$, separating a foreground source from the background source(s), a common scenario in practice.

The first key idea is that there is a closed form compact partial inversion of the convolutive mixing (see (2.7)–(2.8) and Thm. 2.1). For example, if $x=As$, $x,s\in R^n$, and A is a nonsingular $n\times n$ matrix, a partial inversion is a square matrix B such that BA is diagonal. The partial inversion can be properly extended when the standard product As is replaced by a convolutive product $A\star s$. In other words, there is a convolutive variant of the cofactor formula in linear algebra. If $N=2$, the total number of demixing coefficients in the partial inversion is the same as that of the mixing coefficients. Let us denote the partially inverted signals by

$$v_j(t), \qquad j=1,2,\ldots,N,$$

and the source signals by $s_j(t)$, then each v_j depends only on s_j, and the v_j's are independent of each other. We will first obtain v_j's using the demixing coefficients. The s_j's can be obtained by further deconvolving the v_j's. Both the v_j's and the s_j's are acceptable results for the independent component analysis (ICA) and are in fact perceptually close, which reflects the non-uniqueness nature of the BSS problem. The v_j's require less computation than the s_j's. The partial inversion approach was based upon reformulating and developing a decorrelation method in the earlier work of Weinstein, Feder and Oppenheim [30]. These authors studied a special form of convolutive mixing and proposed an iterative method.

To impose independence on v_j's, we consider as in [30] the vanishing cross correlations (with time lags):

$$E[v_i(t)v_j(t-n)]=0 \qquad \forall n, \tag{1.1}$$

from which we derive algebraic equations for the demixing coefficients. As this condition is only theoretically meaningful and the number of equations may not be the same as the number of unknowns in general, we formulate a nonlinear least squares problem to minimize the square sum of cross correlation coefficients with time lags in a certain range. Such an objective function F_{in} is quartic in a high dimensional space R^{NMN}, with NMN on the order of hundreds, and M a positive integer equal to the length of mixing linear convolution.

Due to scale invariance of the BSS problem, zero demixing filter coefficients may be a solution. To remove such a trivial solution, one commonly introduces a normalization condition, by requiring that the l_k $(k\geq 1)$ norm of the demixing filter coefficients be 1. Our second idea is to choose $k=1$ (1-norm) because the resulting demixing coefficients have sparser multi-peak structures, and decay faster than those using $k=2$ (or in general $k>1$). The l_1 solutions are in a sense the minimum length (low dimensional) solutions among those that approximately satisfy $F_{in}=0$ (decorrelation). We shall illustrate this point later with examples. One may also think of sparse multiple peak structures as a compact (low dimensional) approximation of multiple reflections [14, 31] along the acoustic paths of sound propagation, as seen in simulated room acoustic data [29]. The computational advantage of a low dimensional

approximation is that the length of the demixing filter coefficients in our algorithm can be an order of magnitude shorter than that of the actual physical demixing filter coefficients. The latter is proved to exist by our compact partial inversion formula (2.7)–(2.8). Our experience with the current method is that l_1 normalization performs consistently better than l_2, even though the improvement may not be perceived distinctly sometimes.

We shall introduce a new objective function F equal to the sum of the decorrelation function F_{in} and a penalty function to control the l_1 norm of solution *so that it is near a fixed value, say 1.* Because of the scale-invariance of the BSS problem, we make use of the l_1 norm as a constraint, different from existing sparse solution problems [12, 5, 32] where the l_1 norm may be *directly minimized*. Nevertheless, we found that l_1-constraint also leads to "sparse solutions" as compared with those from the l_2-constraint. The l_1 norm is known to be a convex relaxation of sparseness measure, and an efficient quantity for computing sparse solutions [16, 27]; see also the related total variation norm [26, 9, 22] for feature enhancement and blind deconvolution in image processing. The l_1 norm may also be applicable in the spectral domain for formant enhancement and speech recognition [24]. The l_1 minimization problem has been extensively studied in the context of compressive sampling and basis pursuit [12, 5, 32]. Under certain optimal conditions, l_1 minimization is equivalent to minimizing the sparseness of solutions subject to linear under-determined constraints, leading to exact and stable signal recovery [5, 6, 13].

The two ideas above, partial inversion and the l_1 constraint, allow us to develop an efficient time domain BSS method. Encouraging results on satisfactory separation of recorded and synthetic sound mixtures of different kinds are reported.

The objective function F (l_1 constrained decorrelation) will be treated as deterministic where expectations are approximated by sufficient data streams. We use the Levenberg-Marquardt method (LM) to solve this approximately "deterministic" minimization problem (see (3.2)). LM is a hybrid method of gradient descent and Newton iteration. Its step size is variable and is controlled by the method, and it is well known for its robustness and efficiency [18, 19]. The advantage is that the convergence to a minimizer is guaranteed. This is different from stochastic gradient descent algorithms [14, 15], where convergence is unknown and depends on source signals' probability distribution function, the initial condition and choice of step size [14]. Another difference is that the stochastic gradient descent method [14, 15] uses higher order statistics while our method relies on non-stationarity of signals and second order statistics. As we shall see, the same initial condition of the form $(1,0,...,0)$ is fine for all three real room recorded data. Because we seek low dimensional solutions under l_1 normalization, the length of demixing filter coefficients is set at an order of magnitude smaller than that in the actual physical measurements [14]. Computations on recorded and synthetic mixtures show satisfactory performance, comparable to the stochastic descent (infomax) method [15] however at much shorter demixing filter lengths.

Related speech enhancement or separation methods have been studied in the literature. One is the microphone array beamforming method based on maximizing directionality ([2] and references therein), another is auditory scene analysis, which models various aspects of human perceptual representation and separation of sources, [3, 4, 25] etc. Earlier work on echo cancellation [28] touches upon decorrelation as well; see also reference on adaptive noise cancelling in [30]. These early works used least mean square estimation methods, which may not be effective for the convolutive

BSS problems.

The paper is organized as follows. In Sec. 2, the compact partial inversion formula of mixtures and the resulting algebraic system of equations of mixing coefficients are derived. In Sec. 3, the l_1 constrained objective function is introduced and the minimization method is discussed. In Sec. 4, numerical results are shown and analyzed to demonstrate the capability of our method to separate speech and music mixtures in both real room and synthetic environments. Conclusions are in Sec. 5.

2. Compact partial inversion and independence

Consider the convolutive mixing model with N sources and N receivers:

$$y_n(t)=\sum_{k=1}^{q}a_k^{n1}s_1(t+1-k)+\cdots+\sum_{k=1}^{q}a_k^{nN}s_N(t+1-k) \quad \text{for } n=1,...,N, \tag{2.1}$$

where s_i's are the independent source signals, the y_i's are the received mixtures, and the a_k^{ij}'s are the mixing coefficients, $i,j=1,...,N$, $k=1,...,q$. We wish to recover $s_i(t)$ from $y_i(t)$ without knowing a_k^{ij}.

We will derive a compact partial inversion formula for (2.1) and derive a system of equations for $\{a_k^{ij}\}$ based on the assumption that $s_1,...,s_N$ are mutually independent. To make the notation simple so that the derivation can be easily understood, we will first consider the $N=2$ case before we discuss the more general $N\geq 2$ case.

2.1. Two sources and two receivers case. Consider the convolutive mixing model of two independent sources:

$$y_1(t)=\sum_{k=1}^{q}a_k^{11}s_1(t+1-k)+\sum_{k=1}^{q}a_k^{12}s_2(t+1-k) \tag{2.2}$$

$$y_2(t)=\sum_{k=1}^{q}a_k^{21}s_1(t+1-k)+\sum_{k=1}^{q}a_k^{22}s_2(t+1-k), \tag{2.3}$$

which can be written as

$$\begin{pmatrix}y_1\\ y_2\end{pmatrix}=\begin{pmatrix}a^{11}*,\ a^{12}*\\ a^{21}*,\ a^{22}*\end{pmatrix}\begin{pmatrix}s_1\\ s_2\end{pmatrix}, \tag{2.4}$$

where $*$ is the convolution operation and a^{ij} or s_i or y_i is an infinite sequence (any finite sequence is extended into an infinite sequence by zero padding). For example, s_1 is $(\cdots,s_1(k-1),s_1(k),s_1(k+1),\cdots)$. The collection of all the infinite sequences forms a semigroup with the binary operation $*$ being associative and commutative, namely

$$(f*g)*h=f*(g*h), \qquad f*g=g*f. \tag{2.5}$$

For the purpose of inversion, we shall consider multiplying a matrix by its adjoint matrix. To this end, let us define

$$\begin{pmatrix}v_1\\ v_2\end{pmatrix}=\begin{pmatrix}a^{22}*,\ -a^{12}*\\ -a^{21}*,\ \ a^{11}*\end{pmatrix}\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \tag{2.6}$$

namely,

$$v_1(t)=\sum_{k=1}^{q}a_k^{22}y_1(t+1-k)-\sum_{k=1}^{q}a_k^{12}y_2(t+1-k) \tag{2.7}$$

$$v_2(t)=-\sum_{k=1}^{q}a_k^{21}y_1(t+1-k)+\sum_{k=1}^{q}a_k^{11}y_2(t+1-k). \tag{2.8}$$

This is a convolutional variant of the cofactor formula in linear algebra.

Now plugging (2.4) into (2.6), we obtain

$$\begin{aligned}\begin{pmatrix}v_1\\v_2\end{pmatrix}&=\begin{pmatrix}a^{22}*, & -a^{12}*\\ -a^{21}*, & a^{11}*\end{pmatrix}\begin{pmatrix}a^{11}*, & a^{12}*\\ a^{21}*, & a^{22}*\end{pmatrix}\begin{pmatrix}s_1\\s_2\end{pmatrix}\\&=\begin{pmatrix}a^{22}*a^{11}*-a^{12}*a^{21}*, & a^{22}*a^{12}*-a^{12}*a^{22}*\\ -a^{21}*a^{11}*+a^{11}*a^{21}*, & -a^{21}*a^{12}*+a^{11}*a^{22}*\end{pmatrix}\begin{pmatrix}s_1\\s_2\end{pmatrix}\\&=\begin{pmatrix}a^{11}*a^{22}*-a^{12}*a^{21}*, & 0\\ 0, & a^{11}*a^{22}*-a^{12}*a^{21}*\end{pmatrix}\begin{pmatrix}s_1\\s_2\end{pmatrix}.\end{aligned}\tag{2.9}$$

Unfolding the convolutions, we have:

$$v_1(t)=\sum_{\tau=1}^{2q-1}\left(\sum_{k=1}^{\tau}a_k^{11}a_{\tau+1-k}^{22}-a_k^{12}a_{\tau+1-k}^{21}\right)s_1(t+1-\tau),\tag{2.10}$$

$$v_2(t)=\sum_{\tau=1}^{2q-1}\left(\sum_{k=1}^{\tau}a_k^{11}a_{\tau+1-k}^{22}-a_k^{12}a_{\tau+1-k}^{21}\right)s_2(t+1-\tau).\tag{2.11}$$

Since v_i only depends on s_i, we conclude that v_1 and v_2 are independent. In particular, we have

$$E(v_1(t)v_2(t-n))=0\qquad\forall n.\tag{2.12}$$

Substituting (2.7) and (2.8) into (2.12), we have

$$\begin{aligned}\sum_{k,m=1}^{q}&-a_k^{22}a_m^{21}E(y_1(t+1-k)y_1(t+1-m-n))\\&+a_k^{12}a_m^{21}E(y_2(t+1-k)y_1(t+1-m-n))\\&+a_k^{22}a_m^{11}E(y_1(t+1-k)y_2(t+1-m-n))\\&-a_k^{12}a_m^{11}E(y_2(t+1-k)y_2(t+1-m-n))=0,\end{aligned}$$

which can be written as

$$\begin{aligned}\sum_{k,m=1}^{q}&-a_k^{22}a_m^{21}C_{k,m,n}^{11}+a_k^{12}a_m^{21}C_{k,m,n}^{21}\\&+a_k^{22}a_m^{11}C_{k,m,n}^{12}-a_k^{12}a_m^{11}C_{k,m,n}^{22}=0\end{aligned}\tag{2.13}$$

for any n, where

$$C_{k,m,n}^{ij}=E(y_i(t+1-k)y_j(t+1-m-n)).\tag{2.14}$$

Let $\boldsymbol{C}_n^{ij}=(C_{k,m,n}^{ij})_{k,m=1,\dots,q}$ be a $q\times q$ matrix. Let $\boldsymbol{a}^{ij}=(a_1^{ij},\dots,a_q^{ij})^T$. Then, (2.13) can be written in vector form:

$$\left(\boldsymbol{a}^{22};\boldsymbol{a}^{12}\right)^T\begin{pmatrix}-\boldsymbol{C}_n^{11} & \boldsymbol{C}_n^{12}\\ \boldsymbol{C}_n^{21} & -\boldsymbol{C}_n^{22}\end{pmatrix}\begin{pmatrix}\boldsymbol{a}^{21}\\ \boldsymbol{a}^{11}\end{pmatrix}:=\boldsymbol{u}^T\boldsymbol{C}_n\boldsymbol{w}=0\tag{2.15}$$

for any n. We note that equations (2.7)–(2.8) decouple the mixtures exactly as seen in (2.10)–(2.11), so that the independence of the sources is passed onto that of (v_1,v_2).

This is different from the approximate independence at the output of a feedforward network (Sec. 10 of [10]) whose filter length and precise relation to the mixing filter are unknown (even if the mixing filter length is known).

Note that

$$C^{ij}_{k,m,n} = C^{ij}_{k',m',n'} \tag{2.16}$$

whenever $k-m-n = k'-m'-n'$. Because of this property, part of the matrices that we encounter in computation are Toeplitz or Hankel matrices, which can be utilized to save storage and computation time.

2.2. The more general N sources and N receivers case. Let us define a $N \times N$ matrix $A = (a^{ij})$, and its determinant by $\alpha(\{a^{kl}\};\cdot)$. The adjoint matrix of A is denoted by $B = (b^{ij})$ where each entry b^{ij} is a function of $\{a^{kl}\}$, denoted by $b^{ij} = \beta^{ij}(\{a^{kl}\};\cdot)$. For example, when $N=3$, $\alpha(\{a^{kl}\};\cdot) = \det A = a^{11}\cdot a^{22}\cdot a^{33} - a^{11}\cdot a^{23}\cdot a^{32} - a^{21}\cdot a^{12}\cdot a^{33} + a^{21}\cdot a^{13}\cdot a^{32} + a^{31}\cdot a^{12}\cdot a^{23} - a^{31}\cdot a^{13}\cdot a^{22}$ and $\beta^{12}(\{a^{kl}\};\cdot) = \frac{\partial \det A}{\partial a^{21}} = a^{13}\cdot a^{32} - a^{12}\cdot a^{33}$. The reason we introduce $\alpha(\{a^{kl}\};\cdot)$ and $\beta^{ij}(\{a^{kl}\};\cdot)$ is that later on we will use a^{ij} to denote an infinite sequence and then the notation $\alpha(\{a^{kl}\};*) = a^{11} * a^{22} * a^{33} - a^{11} * a^{23} * a^{32} - a^{21} * a^{12} * a^{33} + a^{21} * a^{13} * a^{32} + a^{31} * a^{12} * a^{23} - a^{31} * a^{13} * a^{22}$ and $\beta^{12}(\{a^{kl}\};*) = a^{13} * a^{32} - a^{12} * a^{33}$ will be infinite sequences, where $*$ is the convolution.

Now we consider the more general N receivers and N sources case (2.1) which can be written as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} a^{11}* & a^{12}* & \cdots & a^{1N}* \\ a^{21}* & a^{22}* & \cdots & a^{2N}* \\ \vdots & \vdots & \vdots & \vdots \\ a^{N1}* & a^{N2}* & \cdots & a^{NN}* \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} := \mathbb{A} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} \tag{2.17}$$

where $\mathbb{A}$ is the $N \times N$ *operator matrix* $(a^{ij}*)$. Now we introduce the adjoint matrix $\mathbb{B}$ whose (i,j) entry is the convolution operator

$$\beta^{ij}(\{a^{kl}\};*)*, \tag{2.18}$$

where $\beta^{ij}(\{a^{kl}\};*)$ by itself is an infinite sequence. For example, when $N=3$,

$$\mathbb{B} = \begin{pmatrix} (a^{22}*a^{33} - a^{23}*a^{32})*, & (a^{32}*a^{13} - a^{33}*a^{12})*, & (a^{12}*a^{23} - a^{13}*a^{22})* \\ (a^{23}*a^{31} - a^{21}*a^{33})*, & (a^{33}*a^{11} - a^{31}*a^{13})*, & (a^{13}*a^{21} - a^{11}*a^{23})* \\ (a^{21}*a^{32} - a^{22}*a^{31})*, & (a^{31}*a^{12} - a^{32}*a^{11})*, & (a^{11}*a^{22} - a^{12}*a^{21})* \end{pmatrix}. \tag{2.19}$$

Now define

$$\begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} = \mathbb{B} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \tag{2.20}$$

THEOREM 2.1. *Suppose s_i and y_j $(i,j=1,...,N)$ are related by Equ. (2.17). If $\{s_i : i=1,...,N\}$ are uncorrelated (or independent), then $\{v_i : i=1,...,N\}$ defined by (2.20) are uncorrelated (or independent).*

Proof. Because of (2.5) and because of the arithmetic cancellation in the product between the matrix and its adjoint matrix, we know the product between the two

operator matrices $\mathbb{B}$ and $\mathbb{A}$ is a diagonal operator matrix with each diagonal entry being the convolution operator $\alpha(\{a^{kl}\};*)*$. Plugging (2.17) into (2.20), we obtain

$$\begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} = \mathbb{B}\mathbb{A} \begin{pmatrix} s_1 \\ \vdots \\ s_N \end{pmatrix} = (\alpha(\{a^{kl}\};*)*) \begin{pmatrix} s_1 \\ \vdots \\ s_N \end{pmatrix}. \tag{2.21}$$

So the decorrelation or independence between $\{s_i : i=1,...,N\}$ is passed to that of $\{v_i : i=1,...,N\}$. □

REMARK 2.1. If the support of a^{kl} in (2.17) is q, then the support p of sequences $\beta^{ij}(\{a^{kl};*\})$ (the demixing coefficients) is equal to the length of linear convolution product of $N-1$ mixing coefficients, or

$$p=(N-1)q-N+2.$$

In particular, when $N\geq 3$, the number of filter coefficients in the partial inversion is larger than that in mixing.

Denote

$$\beta^{ij}(\{a^{kl};*\})=\{b_1^{ij},...,b_p^{ij}\}. \tag{2.22}$$

The BSS problem is reduced to finding pN^2 number of demixing filter coefficients b_n^{ij}, for which one imposes a proper set of covariance conditions and formulates an optimization problem, similar to what we have done before. The details are omitted.

REMARK 2.2. The above theorem proved the *existence* of a *finite length* solution $\{b_n^{ij} : n=1,...,p;i,j=1,...,N\}$ for the BSS problem. In particular, if we are not blind, and can measure the impulse response a^{kl} which is independent of the sources, then given any mixture recorded in the same environment (same source and receiver locations) we immediately obtain the v_i's in the theorem. Each of them contains exactly one source, and "source separation" is achieved.

In BSS, one relies on independence of the v_i's to write down (theoretically infinitely many) equations that a^{ij} should satisfy. For example $E(f(v_i(t))g(v_j(t-k)))=0$, $\forall\ i\neq j$, for infinitely many choices of f and g. It seems unknown whether these equations or a subset of these equations uniquely determine a^{ij}.

3. l_1 Constrained minimization

Now, we let n vary and solve the decorrelation equations (2.13) or (2.15) to obtain a_k^{ij}. Since we may have more equations than unknowns in general, it is natural to solve (2.15) (with n running from $-N$ to N) in the least squares sense. To avoid the trivial solution $\boldsymbol{u}=\boldsymbol{w}=0$ and fix scaling invariance, we impose the constraint $\|\boldsymbol{u}\|=1$ and $\|\boldsymbol{w}\|=1$ for some vector norm $\|\cdot\|$. To capture peaks in the solution, we use the l_1 norm in a penalty term. The mixing and demixing filter coefficients typically have multi-peaks due to multiple reflections experienced by the acoustic waves in the environment. The l_1 norm is good at distinguishing peaks. In contrast, l_2 norm tends to spread solutions. Moreover, the l_1 norm maintains certain sparseness between peaks, which is also desired for avoiding spurious oscillations. A numerical example will be presented later to illustrate this point.

The objective function to be minimized is

$$F(\boldsymbol{u},\boldsymbol{w}) := \sum_n \left|\boldsymbol{u}^T \boldsymbol{C}_n \boldsymbol{w}\right|^2 + \sigma^2(\|\boldsymbol{u}\|_{l_1}^2 - 1)^2 + \sigma^2(\|\boldsymbol{w}\|_{l_1}^2 - 1)^2, \tag{3.1}$$

where $\sigma > 0$ is a positive relaxation parameter. So

$$[\boldsymbol{u};\boldsymbol{w}] = \operatorname{argmin} \sum_n \left|\boldsymbol{u}^T \boldsymbol{C}_n \boldsymbol{w}\right|^2 + \sigma^2(\|\boldsymbol{u}\|_{l_1}^2 - 1)^2 + \sigma^2(\|\boldsymbol{w}\|_{l_1}^2 - 1)^2 \tag{3.2}$$

where the Matlab notation $[\boldsymbol{u};\boldsymbol{w}]$ defines a vector as a concatenation of two vectors. The relaxation parameter σ is to be properly chosen so that the minimizing sequence can evolve from a sparse initial condition. If it is too small, the constraint is too weak to be effective. If it is too large, the minimizing sequence evolves too slowly. For a proper choice of σ, the l_1 norm will vary in a small neighborhood of one, and $[\boldsymbol{u};\boldsymbol{w}]$ evolves effectively to a nonzero limit. In our numerical examples, we take $\sigma = 0.002$ and have found that once we found this σ after a few tests, it works for all different mixtures. In principle, σ depends on the size of the optimization problem. However, we do not know an automated method to find the optimal σ. The good values of σ are pre-determined by hearing tests. Alternatively, one may minimize the following Rayleigh quotient with more overheads of computing ratios and their gradients:

$$[\boldsymbol{u};\boldsymbol{w}] = \operatorname{argmin} F(\boldsymbol{u},\boldsymbol{w}) = \operatorname{argmin} \sum_n \left(\frac{\boldsymbol{u}^T \boldsymbol{C}_n \boldsymbol{w}}{\|\boldsymbol{u}\|_{l_1} \|\boldsymbol{w}\|_{l_1}} \right)^2. \tag{3.3}$$

In numerical tests that have been performed, the results are comparable.

Next, we proceed to find the minimizer of F in (3.1) or (3.3). We have writen F as a sum of the squares of functions so that the Levenberg-Marquardt (LM) method applies. The LM method minimizes a function of the form

$$g(\boldsymbol{x}) = \frac{1}{2} \sum_{i=1}^{m} (f_i(\boldsymbol{x}))^2,$$

or a sum of squares of functions of $\boldsymbol{x} = (x_1, ..., x_n)$. If we use steepest descent, the searching direction is $-\nabla g = -\boldsymbol{J}^T \boldsymbol{f}$, where $\boldsymbol{f} = (f_1, ..., f_m)^T$ and $\boldsymbol{J} = (\partial f_i / \partial x_j)$ is the Jacobian. If we apply the standard Gauss-Newton method, then $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\boldsymbol{J}^T \boldsymbol{J})^{-1} \boldsymbol{J}^T \boldsymbol{f}$. When $m = n$, $\boldsymbol{J}$ is a square matrix this is the Newton method for solving $\boldsymbol{f}(x) = 0$. The LM method is an interpolation between steepest descent and Gauss-Newton, namely

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\boldsymbol{J}^T \boldsymbol{J} + \mu_k \boldsymbol{I})^{-1} \boldsymbol{J}^T \boldsymbol{f} \tag{3.4}$$

with $\mu_k > 0$. When $\mu_k = 0$, it becomes Gauss-Newton; and when μ_k is large, it moves a small step along a direction very close to the steepest descent direction. The matrix being inverted is always nonsingular, even when $\boldsymbol{J}^T \boldsymbol{J}$ is singular. The LM method has a strategy to choose μ_k to guarantee the reduction of g. For more exposition of the LM method, see [18, 19] among others. The web site http://www.ics.forth.gr/ lourakis/levmar/ provides a public C/C++ code by M. Lourakis, and a link to the Matlab code by H. B. Nielsen.

Though the l_1 norm seems less differentiable, it is convex and can be easily regularized. The derivatives $\nabla_{u_j} \|\boldsymbol{u}\|_{l_1} = \nabla_{u_j} |u_j|$ are approximated by $\nabla_{u_j} |u_j| = \frac{u_j}{\epsilon + |u_j|}$ with a small positive parameter ϵ. This turns out to be sufficient for computation.

When the $\boldsymbol{a}^{ij}$'s are computed, the decorrelation of v_1 and v_2 follow from equation (2.12) with $n=-N,-N+1,...,N$. Furthermore, if the $\boldsymbol{a}^{ij}$'s are close to the exact $\boldsymbol{a}^{ij}$'s, then v_i only contains information from the exact s_i alone, for $i=1,2$ (see (2.10)–(2.11)). At this point, we have a set of BSS solutions, which are often perceptually good enough. The source signals s_i $(i=1,2)$ are recovered from (2.10) and (2.11) by a FFT-based deconvolution method.

4. Computational results

The computations reported here are for 2 sources and 2 receivers. The data are either recorded mixtures in an office size room or conference room or synthetic convolutive mixtures, as listed in Table 4.1. They will be called case (1)-1, (1)-2, (1)-3 and case (2) in the following discussion. The three real room recorded data in case (1) are from [15].

case #	description
(1)-1	A speaker has been recorded with two distance talking microphones in a normal office room with loud music in the background. The distance between the speaker, cassette player and the microphones is about 60cm in a square ordering.
(1)-2	Two speakers have been recorded speaking simultaneously. Speaker 1 says the digits from one to ten in English and speaker 2 counts at the same time the digits in Spanish. The recording has been done in a normal office room. The distance between the speakers and the microphones is about 60cm in a square ordering.
(1)-3	Two speakers have been recorded speaking simultaneously. The recording was in a conference room (5.5 m by 8 m). The conference room had some air-conditioning noise. Both speakers were reading a section from the newspaper for 16 sec. Microphones were placed 120 cm away from the speakers.
(2)	Synthetic mixture of two female speeches. The mixing coefficients $\boldsymbol{a}^{ij}$ in (2.2)–(2.3) are given by the empirical formula of [29, (8)], and renormalized so that $\left\|[\boldsymbol{a}^{22};\boldsymbol{a}^{12}]\right\|_{l_1}=1=\left\|[\boldsymbol{a}^{21};\boldsymbol{a}^{11}]\right\|_{l_1}$. The exact $\boldsymbol{a}^{ij}$ can be read from Table 4.6-$\boldsymbol{a}^{ij}$. Mixtures and the clean sources are in Table 4.5.

TABLE 4.1. *Description of real room recorded data from [15] and the synthetic mixture, at sampling rate 16 kHz.*

Table 4.2 plots the three pairs of mixtures of case (1). The associated computation results are listed in Table 4.3 and Table 4.4. See also Table 4.10 (to be discussed later) for results from an "economical" version of the algorithm. We have heard both our separation results and those posted on [15]. Perceptually, they are very close. For case (1)-1 and (1)-2, the separation results are very good. For case (1)-3, distinct improvement can be heard.

For the synthetic mixtures in case (2), the mixing coefficients are known, and hence by our compact partial inversion formula (2.7)–(2.8), we know exactly the demixing filter coefficients. We compare computed demixing filter coefficients with the exact values. Computational results of case (2) are listed in Table 4.6. From the plot of $\boldsymbol{a}^{ij}$'s, we see that the numerical solution is sparse, which agrees with the

exact $\boldsymbol{a}^{ij}$. One can also compare the computed s_i's in Table 4.6 with the exact s_i's in Table 4.5.

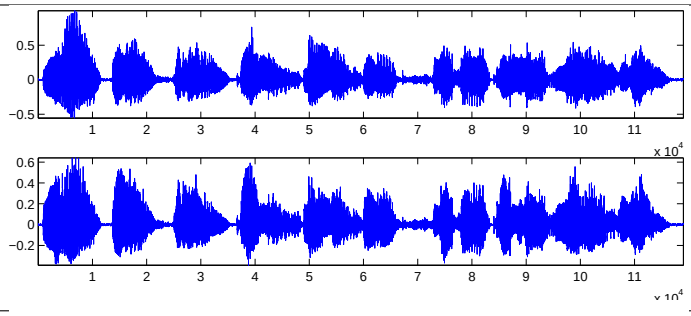
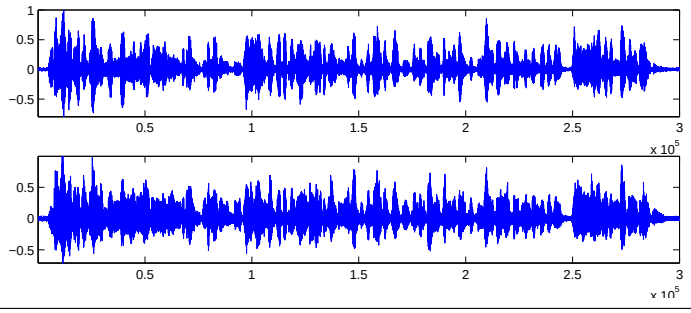
case #	Mixture
(1)-2	
(1)-3	

TABLE 4.2. *Plot of the mixture for case (1). See Fig. 1.1 for case (1)-1.*

The $\boldsymbol{C}_n$'s in the objective function are computed from (2.14) with $n=-N,...,N$, while k and m range from 1 to q. We take demixing filter length $q=50$ for all the cases. In case (1), q is unknown; in case (2), the exact q is 46. We take $N=150$, which means the range of n in (2.13), (2.15), (3.1) or (3.2) is from -150 to 150. So, there are a total of 301 equations in (2.13) or (2.15) and $4q=200$ unknowns. A segment of $y_i(t)$ of length L is used to approximate the expectation. To reduce the statistical error, in all cases we have used all the available data stream to estimate the $C^{ij}_{k,m,n}$ in (2.14). The maximum of $C^{ij}_{k,m,n}$ from the data are typically on the order of $O(10^{-2})$. To avoid loss of significant digits, we multiply $C^{ij}_{k,m,n}$ by 100 so that the adjusted maximum of $C^{ij}_{k,m,n}$ is order 1. The value of σ in (3.1) is fixed at 0.002 for all the cases.

In all the cases, we take the initial value to be $(1,0,...,0)$ for $\boldsymbol{a}^{ii}$, $i=1,2$, and $(0,0,...,0)$ for $\boldsymbol{a}^{ij}$ when $i\neq j$. When the LM method is applied to the objective function (3.1) that contains the l_1 norm, the derivative terms like $\nabla_{u_j}\|\boldsymbol{u}\|_{l_1}=\nabla_{u_j}|u_j|$ are numerically approximated by $\nabla_{u_j}|u_j|=\frac{u_j}{\epsilon+|u_j|}$ with $\epsilon=10^{-16}$.

The LM method is implemented in Matlab [18, 19], with the stopping parameters for iterations set to be opts=$[10^{-3}, 10^{-7}, 10^{-12}, 1000, 10^{-15}]$. These numbers have the following meaning: 10^{-3} and 10^{-15} are related to the initial value and lower bound of μ_k in LM method (see (3.4)). LM iteration will stop when the l_∞ norm of the gradient is less than 10^{-7}, or when $\|[\boldsymbol{u};\boldsymbol{w}]^{m+1}-[\boldsymbol{u};\boldsymbol{w}]^m\|_{l_2}\leq 10^{-12}(10^{-12}+\|[\boldsymbol{u};\boldsymbol{w}]^m\|_{l_2})$, or when the number of iterations exceeds 1000. The superscript m refers to the m-th iterate. In all cases above, LM method reduces the gradient of the objective function

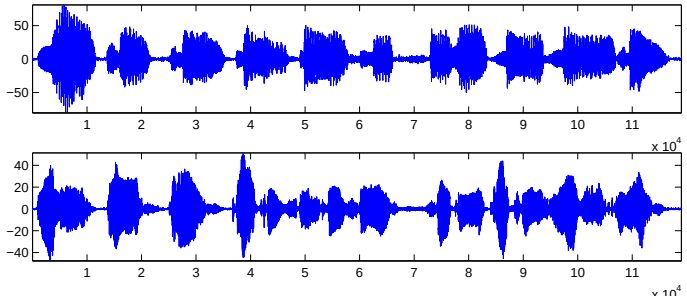
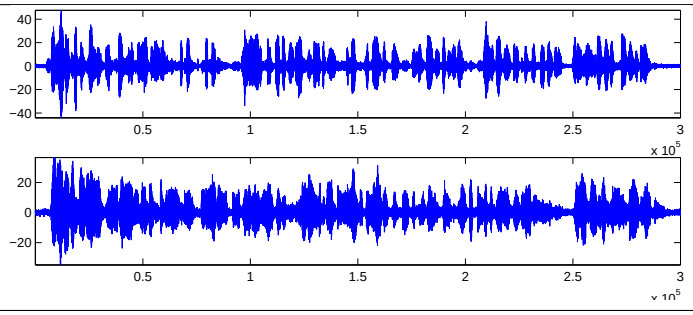
case #	calculated s^i using (2.7)–(2.8) and then (2.10)–(2.11).
(1)-2	
(1)-3	

TABLE 4.3. *Plot of the computed source signals s for case (1). See Fig. 1.1 for results of case (1)-1.*

below the preset value 10^{-7}.

The partial derivatives of $f_n := \boldsymbol{u}^T \boldsymbol{C}_n \boldsymbol{w}$ with respect to a_k^{ij} are as follows. Let $J_{n,k}^{ij} = \frac{\partial f_n}{\partial a_k^{ij}}$, and $\boldsymbol{J}^{ij} = (J_{n,k}^{ij})$ with $n = -N, ..., N, k = 1, ..., q$. Then

$$J_{n,k}^{11} = \sum_{i=1}^{q} a_i^{22} C_{i,k,n}^{12} - a_i^{12} C_{i,k,n}^{22} \tag{4.1}$$

$$J_{n,k}^{12} = \sum_{i=1}^{q} a_i^{21} C_{k,i,n}^{21} - a_i^{11} C_{k,i,n}^{22} \tag{4.2}$$

$$J_{n,k}^{21} = \sum_{i=1}^{q} a_i^{12} C_{i,k,n}^{21} - a_i^{22} C_{i,k,n}^{11} \tag{4.3}$$

$$J_{n,k}^{22} = \sum_{i=1}^{q} a_i^{11} C_{k,i,n}^{12} - a_i^{21} C_{k,i,n}^{11}. \tag{4.4}$$

We have (2.16):

$$C_{k,m,n}^{ij} = C_{k',m',n'}^{ij} \tag{4.5}$$

whenever $k - m - n = k' - m' - n'$. So $\boldsymbol{J}^{12}$ and $\boldsymbol{J}^{22}$ are Toeplitz matrices while $\boldsymbol{J}^{11}$ and $\boldsymbol{J}^{21}$ are Hankel matrices. These matrix structures lead to savings in storage and computation time.

The q value in our computation is much smaller than the dimension of a typical room impulse response (on the order of 1000), [14]. In fact, the length of the demixing

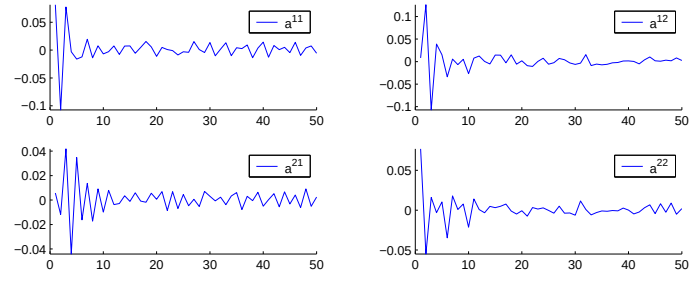
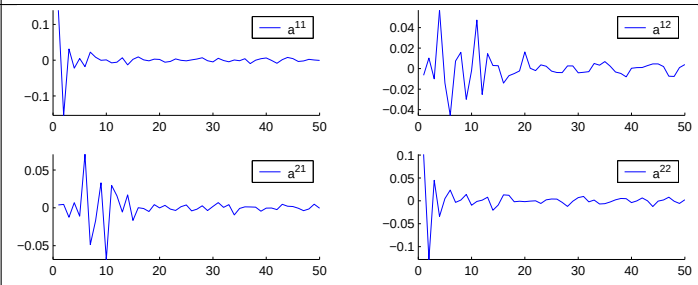
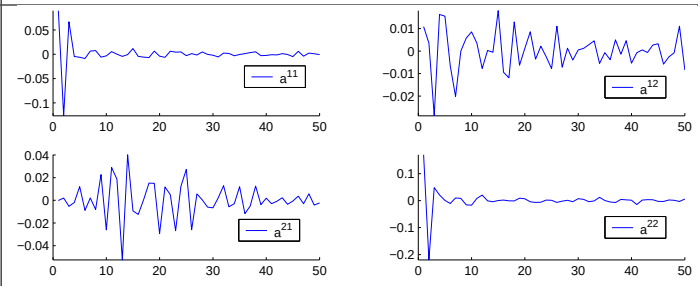
case #	$\boldsymbol{a}^{ij}$ obtained from solving (3.2), (2.15)
(1)-1	
(1)-2	
(1)-3	

TABLE 4.4. *Plot of the computed* $\boldsymbol{a}^{ij}$ *for case (1).*

filter coefficients for case (1)-3 is 2048 in [15] while ours is 50. Our results on case (1) shows that a low dimensional solution can separate sound mixtures. A more systematic study of this phenomenon will be presented in a subsequent paper.

The computation times are affected by the following factors: (I) how much data are used to estimate the $C^{ij}_{k,m,n}$ in (2.14); (II) size of q and N; (III) stopping criteria for LM or the number of iterations in LM. For the results in Table 4.3, we have used all the data shown in Table 4.2 to estimate $C^{ij}_{k,m,n}$; we have taken $q=50$ and $N=150$, and we have waited for LM to converge with stopping criteria described above. The CPU times for case (1)-1, (1)-2, (1)-3 and case (2) are 47, 53, 119, and 20 seconds respectively. The computation is done with Matlab on a Compaq laptop with 1.6G Hz AMD 64 bit dual core CPU.

However, much less data still yields quite good separation. As demonstrated later in Table 4.10, where 5 iterations of LM and 12800 samples (or 0.8 second of data stream at 16 kHz sampling frequency) are sufficient for a quality separation of the recorded mixtures. To avoid the initial silent period in the recorded data, we use the data stream from 1.0 sec to 1.8 sec. Moreover, we also take $q=50$ and $N=50$, which means that the number of equations $(2N+1=101)$ is less than the number of un-

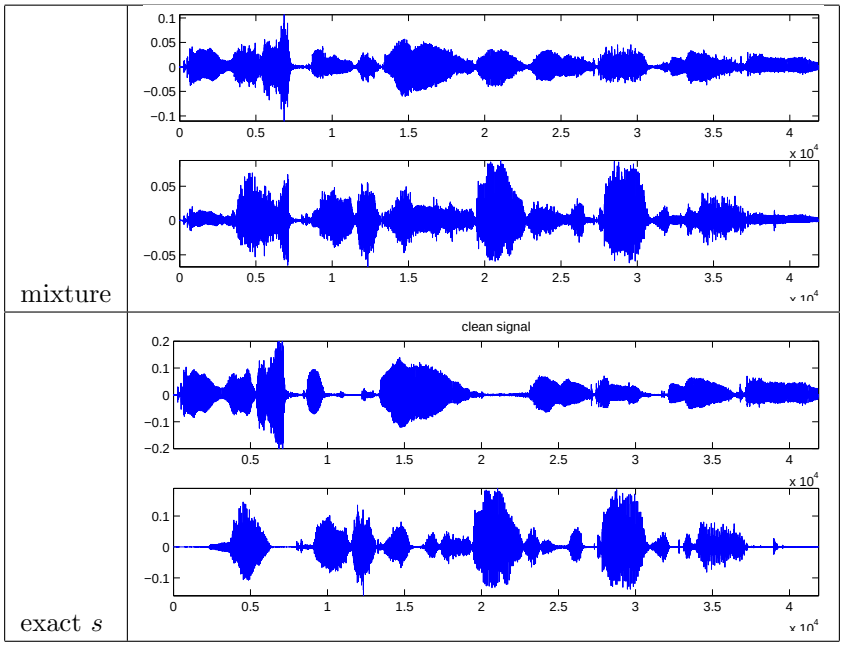


TABLE 4.5. *Case (2). Mixtures and exact source signals.*

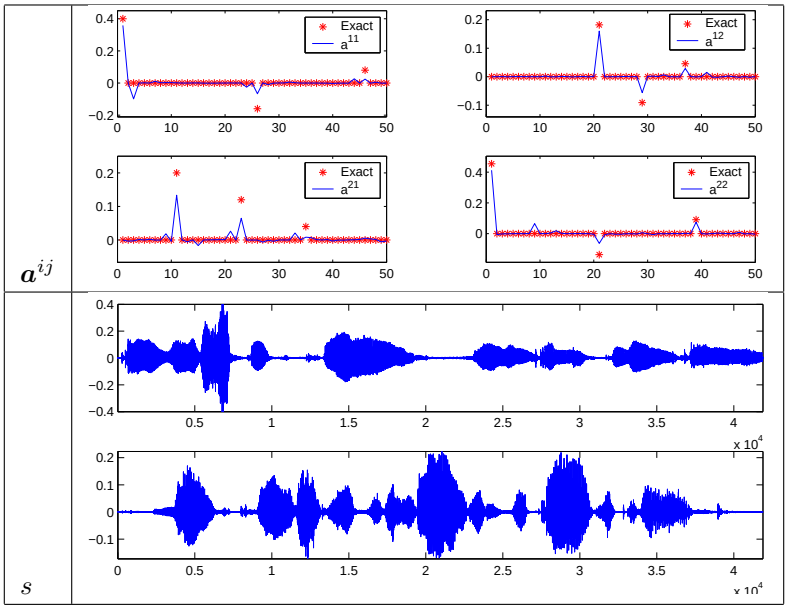


TABLE 4.6. *Case (2) with l_1 norm constrained minimization ((3.2)). (top) Computed $\boldsymbol{a}^{ij}$ and exact $\boldsymbol{a}^{ij}$; (bottom) computed source signals s.*

knowns ($4q = 200$). The LM method (3.4) can handle such a degenerate case because $\mu_k > 0$ and the matrix to be inverted is nonsingular. To further save computational

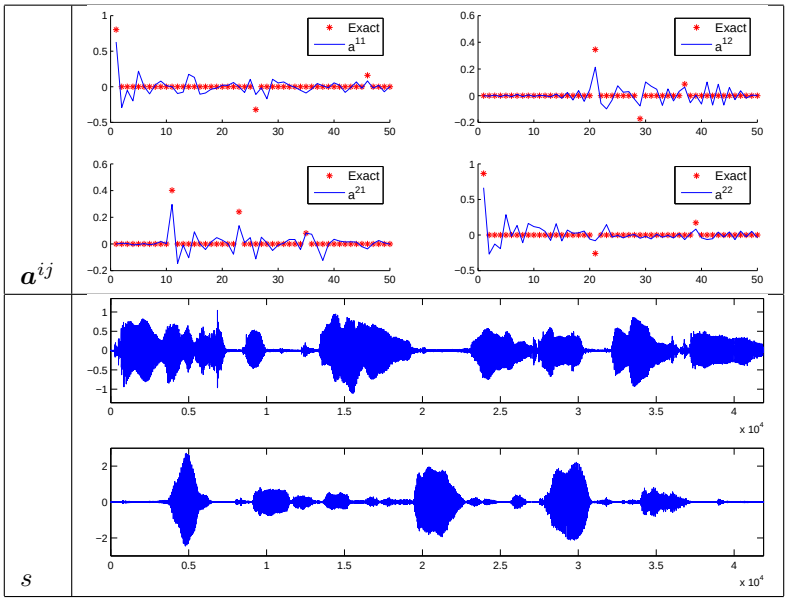


TABLE 4.7. *Case (2) with l_2 norm constrained minimization (replace $\|\cdot\|_{l_1}$ by $\|\cdot\|_{l_2}$ in (3.2)). (top) Computed $\boldsymbol{a}^{ij}$ and exact $\boldsymbol{a}^{ij}$; (bottom) computed source signals s.*

	$\bar{\rho}(s_1^{ex}, s_2^{ex})$	$\bar{\rho}(y_1, y_2)$	$\bar{\rho}(v_1, v_2)$	$\bar{\rho}(s_1, s_2)$	$\bar{\rho}(s_1^{Lee}, s_2^{Lee})$
case (1)-1	N/A	8.31e-1	5.62e-3	1.09e-2	3.51e-2
case (1)-2	N/A	7.75e-1	8.81e-3	1.83e-2	1.12e-2
case (1)-3	N/A	5.46e-1	6.28e-3	3.36e-3	7.93e-3
case (2)	2.69e-3	3.19e-1	1.87e-3	4.51e-3	N/A

TABLE 4.8. *Values of the correlation coefficient $\bar{\rho}(\cdot,\cdot)$. The s_1^{ex} and s_2^{ex} are the original sources. The s_1^{Lee} and s_2^{Lee} are the results from [15].*

case #	I_1	I_2	F
(1)-1	6.47e-4	4.71e-5	3.04e-9
(1)-2	7.70e-4	5.79e-5	3.52e-9
(1)-3	1.58e-3	1.19e-4	1.44e-8
(2)	4.21e-4	3.06e-5	3.24e-9
(2)-Ex	1.23e-3	8.93e-5	7.98e-9

TABLE 4.9. *Values of I_1, I_2 of independence measure (4.9) and of the objective function F evaluated at the computed $\boldsymbol{a}^{ij}$'s as well as at the exact $\boldsymbol{a}^{ij}$'s. (2)-EX means evaluating I_1, I_2 and F at exact $\boldsymbol{a}^{ij}$'s. The exact $\boldsymbol{a}^{ij}$ is not the global minimizer of the objective function, though it gives the best separation perceptually.*

time, we directly use the $v^i(t)$ obtained from (2.7)–(2.8) as the separation results, which is perceptually good enough. So, we do not have to go further to find s_i from each v_i. The results shown in Table 4.10 are plots of those v_i's. We have heard the results and they are already quite good. The total computation time for Table 4.10

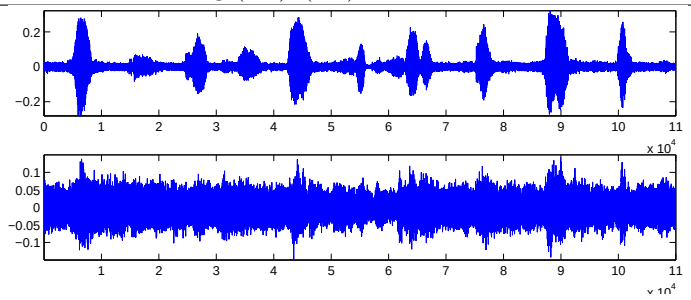
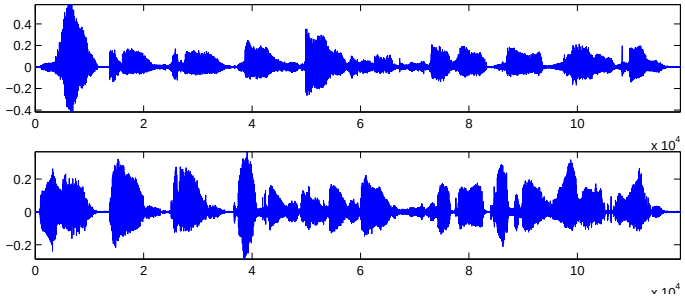
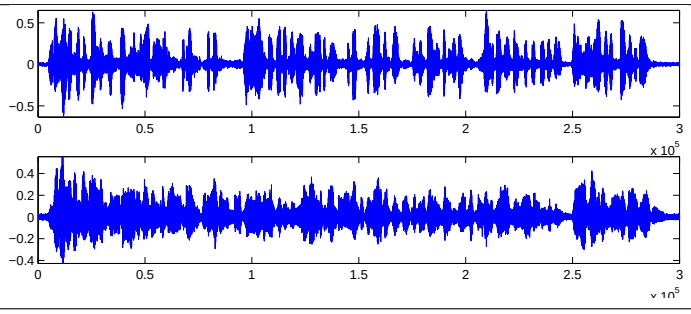
case #	calculated v^i using (2.7)–(2.8).
(1)-1	
(1)-2	
(1)-3	

TABLE 4.10. *Plots of the v for case (1) after 5 LM iterations. A piece of 0.8 sec of the data stream (starting from 1.0 sec after the initial to avoid initial silence) is used to estimate the demixing filter coefficients.* $q=50$, $N=50$.

is 4.7, 4.9, and 4.7 seconds for case (1)-1, (1)-2 and (1)-3 respectively.

As a quantitative measure of separation, we compute the maximal correlation coefficient over multiple time lags:

$$\bar{\rho}(a,b) = \max_{k\in\{-K,\ldots,K\}} |\rho(a(t),b(t+k))|, \tag{4.6}$$

where ρ is the correlation coefficient defined by

$$\rho(a(t),b(t)) = \frac{\mathrm{cov}(a(t),b(t))}{\sqrt{\mathrm{cov}(a(t),a(t))\,\mathrm{cov}(b(t),b(t))}} \tag{4.7}$$

with

$$\mathrm{cov}(a(t),b(t)) = L^{-1}\sum_{t=1}^{L} a(t)b(t) - L^{-2}\sum_{t=1}^{L} a(t)\sum_{t=1}^{L} b(t)$$

being the estimation of covariance of a and b when there are L samples. The values of L for computing ρ equals to the size of the total available data stream, which can be read from the plots in Table 4.2. The value K in (4.6) is 20 in our calculations. The $\bar{\rho}$ is computed for the mixtures, sources (in case 2), and the separated v and s. For comparison, we also computed $\bar{\rho}$ for the separated signals of [15]. The results are listed in Table 4.8 which shows that the $\bar{\rho}$ values of the mixtures are much larger than those of the separated signals. Table 4.8 also implies that our method is comparable with [15]. In fact, perceptually, we also find so. We note that in the last row of Table 4.8, the v_i's have smaller $\bar{\rho}$ values than those from the original clean signals.

For case (2), since we know the exact sources we have computed the following quantities:

$$\frac{\bar{\rho}(v_1,s_1^{ex})}{\bar{\rho}(v_1,s_2^{ex})}=217.425,\quad \frac{\bar{\rho}(v_2,s_1^{ex})}{\bar{\rho}(v_2,s_2^{ex})}=0.00511,$$
$$\frac{\bar{\rho}(s_1,s_1^{ex})}{\bar{\rho}(s_1,s_2^{ex})}=252.303,\quad \frac{\bar{\rho}(s_2,s_1^{ex})}{\bar{\rho}(s_2,s_2^{ex})}=0.00732, \tag{4.8}$$

where the s_1^{ex} and s_2^{ex} are the exact original source signals. The above ratio measures the relative closeness of v_i or s_i to s_1^{ex} and s_2^{ex}. These ratios indicate that v_i and s_i are quite close to s_i^{ex} and quite "orthogonal" to s_j^{ex} for $j\neq i$.

In Table 4.9, we computed the independence measure

$$I_p:=\left(\sum_{n=1}^{N}|\boldsymbol{u}^T\boldsymbol{C}_n\boldsymbol{w}|^p\right)^{1/p}, \tag{4.9}$$

where $\boldsymbol{u}=[\boldsymbol{a}^{22};\boldsymbol{a}^{12}]$ and $\boldsymbol{w}=[\boldsymbol{a}^{21};\boldsymbol{a}^{11}]$ (by (2.15)), and $p=1,2$. We also listed the values of the associated objective function F in (3.1). The last row of Table 4.9 showed the values of I_p and F evaluated at the exact $\boldsymbol{a}^{ij}$ of case (2). We see that the exact $\boldsymbol{a}^{ij}$ is not the global minimizer of F. The $\boldsymbol{a}^{ij}$ returned from LM iterations can give even smaller values of F. On the other hand, the exact a^{ij} for synthetic mixtures always gives the best separation by the ear. This shows that independence as an objective measure of separation is not the same as the perceptual "ear measure", it is rather a reasonable correlate.

Ideally, we hope that the exact $\{\boldsymbol{a}^{ij}\}$ is a global minimizer of F in (3.1) so that we can "locate" it by measuring the size of F. However, numerical experiments indicated that the landscape of F is complicated; F may have multiple local minimizers. In synthetic mixture computation, we observed that the desired demixing filter coefficients are not global minimizers of F in general, instead they are *near minimizers*. Partly this is because given any two clean sources of finite length, we do not expect their cross correlations with *different* time lags to be *all* zero as in (1.1). Hence the desired (perceptually optimal) demixing coefficients will not render F in (3.1) equal to zero, even though they can make F small. Algebraically, it is possible that there are other choices of demixing coefficients making F even smaller. If this happens, the LM solver searching for a minimizer may not find the desired demixing coefficients.

Related to the above observation, even though we have used a small data set to estimate the demixing filter coefficients (Table 4.10), we do not have a *quantitative* way to determine how the quality of the separation depends on the size of the data set. This is because a precise quantity to measure the quality of separation is not yet available. There are approximate objective measures. For example, the last two rows of Table 4.9 list the values of I_1, I_2 and F (see (4.9), (3.1)) for the synthetic

mixture case (case (2) with known solution) where the exact demixing coefficients produce larger I_1, I_2 and F, even though the exact demixing coefficients give the best separation. For the problem in Table 4.10, quality is saturated beyond $q = 50$, while for $q < 20$, quality is getting poor. Likewise, quality is reduced if sample size is less than 6000 or 400 milliseconds. Two different segments of the same data do not make much difference in estimated demixing filters in this case. In general, if the speakers are moving, different segments of data will give different estimations.

We remark that minimizing the l_2 constrained objective function (meaning the l_1 in the last two terms of (3.1) or (3.2) is replaced by l_2) tends to give oscillatory $\boldsymbol{a}^{ij}$'s which are much less sparse than the $\|\cdot\|_{l_1}$ constrained $\boldsymbol{a}^{ij}$'s. This can be seen by comparing Table 4.7 with Table 4.6. In case (2), the $\boldsymbol{a}^{ij}$'s computed by l_2 constrained objective function can be very different from their exact values, and they may yield much smaller values of F, I_1, and I_2 than those at the exact $\boldsymbol{a}^{ij}$'s. In some sense, l_2 constrained filter coefficients bear quite some resemblance to the frequency domain solutions [23, 20, 17] in that the support of $\boldsymbol{a}^{ij}$'s is much longer. In contrast, the l_1 constrained solutions tend to be sparse and have less support, which helps the demixing of long convolutions in low dimensions.

Finally, we comment on how one may apply the current algorithm to data recorded in a noisy environment. If noise comes from a point source, then adding an additional receiver, we see that the derivation in Sec. 2 is still valid. If noise does not come from a point source or an additional receiver is not available, the source signals are then noisy, as one may encounter in recording if a machine or environmental noise is not properly shielded. The working assumption of independence of the source signals is no longer valid because noise polluted source signals are correlated. A preprocessing step (such as whitening [14]) is often proposed to help condition the mixtures. The preprocessing ameliorates this problem to some extent but cannot entirely cure the problem. The data of case (1)-1 and case (1)-2 are high quality recordings with minimal noise disturbance, while case(1)-3 is recorded with air-conditioning noise [15]. The resulting separation of case(1)-3 is not as clean as in case(1)-1 or case(1)-2. On the positive side, applying BSS method to noisy mixtures improves the hearing conditions of the source signals, especially if the source signals are more dominant than noise.

5. Conclusions

We developed a time domain approach to BSS based on an l_1 constrained minimization of cross correlations with time lags. The method produces estimates of low dimensional demixing/mixing filter coefficients that are effective for separating both room recorded and synthetic mixtures of music and speech. Future work will further investigate preprocessing and fast algorithms.

Acknowledgements. The authors wish to thank Professor S. Osher and Professor G. Papanicolaou for their helpful comments and encouragements. Part of the work was presented at the Conference on Mathematics and Vision held at the Institute for Mathematical Behavioral Sciences (IMBS) at UC Irvine in Nov. 2007. We thank IMBS director and Professor Donald Saari for his interest and for sponsoring the conference. We also thank the anonymous referees for their constructive comments.

The work was partially supported by NSF grant DMS-0712881, NIH grant 2R44DC006734; the CORCLR (Academic Senate Council on Research, Computing and Library Resources) faculty research grant MI-2006-07-6, and a Pilot award of the Center for Hearing Research at UC Irvine.

REFERENCES

[1] A. Bell and T. Sejnowski, *An information-maximization approach to blind separation and blind deconvolution*, Neural Computation, 7, 1129–1159, 1995.

[2] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, Berlin, 2001.

[3] A. Bregman, *Auditory Scene Analysis*, MIT Press, Cambridge, MA, 1990.

[4] G. Brown and D.L. Wang, *Separation of speech by computational auditory scene analysis*, J. Benesty, S. Makino and J. Chen (eds.), Speech Enhancement, Springer, New York, 371–402, 2005.

[5] E. Candès, J. Romberg and T. Tao, *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52(2), 489–509, 2006.

[6] E. Candès, J. Romberg and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math., 59(8), 1207–1223, 2006.

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

[8] J.F. Cardoso and A. Souloumiac, *Blind beamforming for non-gaussian signals*, IEEE Proceedings-F, 140(6), 362–370, 1993.

[9] T. Chan and C. Wong, *Total variation blind deconvolution*, IEEE Transactions Image Processing, 7, 370–375, 1998.

[10] S. Choi, A. Cichocki, H. Park and S. Lee, *Blind source separation and independent component analysis: a review*, Neural Information Processing -Letters and Reviews, 6(1), 1–57, 2005.

[11] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, Wiley, 2002.

[12] D. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization*, Proc. Nat. Acad. Sci., USA, 100(5), 2197–2202, 2003.

[13] D. Donoho, *For most large undertermined systems of equations, the minimal l_1-norm solution is also the sparsest solution*, Comm. Pure Appl. Math., 59(6), 797–829, 2006.

[14] S.C. Douglas and M. Gupta, *Scaled Natural Gradient Algorithms for Instantaneous and Convolutive Blind Source Separation*, IEEE ICASSP, II637–II640, 2007.

[15] T.W. Lee, *Blind Source Separation: Audio Examples.* `http://inc2.ucsd.edu/~tewon`, `http://www.cnl.salk.edu/~tewon/Blind/blind_audio.html`

[16] S. Levy and P. Fullagar, *Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution*, Geophysics, 46, 1235–1243, 1981.

[17] J. Liu, J. Xin and Y. Qi, *A dynamic algorithm for blind separation of convolutive sound mixtures*, Neurocomputing, 72, 521–532, 2008.

[18] M.I.A. Lourakis, *A brief description of the Levenberg-Marquardt algorithm implemented by* `levmar`. `http://www.ics.forth.gr/ lourakis/levmar/levmar.pdf`

[19] K. Madsen, H.B. Nielsen and O. Tingleff, *Methods for non-linear least squares problems*, 2nd edition. `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3215/pdf/imm3215.pdf`

[20] N. Murata, S. Ikeda and A. Ziehe, *An approach to blind separation based on temporal structure of speech signals*, Neurocomputing, 41, 1–24, 2001.

[21] T. Nishikawa, H. Saruwatari and K. Shikano, *Blind source separation based on multi-stage ICA combining frequency-domain ICA and time-domain ICA*, Proc. IEEE ICASSP, 1, 917–920, 2002.

[22] S. Osher, A. Solé and L. Vese, *Image decomposition and restoration using total variation minimization and the H^{-1} norm*, Multiscale Modeling Simulations, 1, 349–370, 2003.

[23] L. Parra and C. Spence, *Convolutive Blind Separation of Non-Stationary Sources*, IEEE Transactions on Speech and Audio Processing, 8(5), 320–327, 2000.

[24] Y. Qi and J. Xin, *A perception and PDE based nonlinear transformation for processing spoken words*, Physica D, 149,143–160, 2001.

[25] N. Roman, S. Srinivasan and D.L. Wang, *Binaural segregation in multisource reverberent enviroments*, J. Acoustic Soc. America, 120, 4040–4051, 2006.

[26] L. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60, 259–268, 1992.

[27] F. Santosa and W. Symes, *Linear inversion of band-limited reflection seismograms*, SIAM J. Sci. Stat. Comput., 7, 1307–1330, 1986.

[28] M. Sondhi, D. Morgan and J. Hall, *Stereophonic acoustic echo cancellation–an overview of the fundamental problem*, IEEE Signal Processing Letters, 2(8), 148–151, 1995.

[29] K. Torkkola, *Blind separation of convolved sources based on information maximization*, Neural Networks Signal Processing, VI, 423–432, 1996.

[30] E. Weinstein, M. Feder and A.V. Oppenheim, *Multi-channel signal separation by decorrelation*, IEEE Trans. on Speech and Audio Processing, 1(4), 405–413, 1993.

[31] K.-C. Yen and Y. Zhao, *Adaptive co-channel speech separation and recognition*, IEEE Trans., Speech and Audio Processing, 7(2), 138–151, 1999.

[32] W. Yin, S. Osher, D. Goldfarb and J. Darbon, *Bregman iterative algorithms for l_1-minimization with applications to compressed sensing*, SIAM J. Image Sci., 1(143), 143–168, 2008.