Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 72 (2008) 521–532
NEUROCOMPUTING
www.elsevier.com/locate/neucom

A dynamic algorithm for blind separation of convolutive sound mixtures

Jie Liu, Jack Xin*, Yingyong Qi

Department of Mathematics, UC Irvine, Irvine, CA 92697, USA

Received 26 March 2007; received in revised form 13 November 2007; accepted 3 December 2007
Communicated by S. Choi
Available online 31 December 2007

Abstract

We study an efficient dynamic blind source separation algorithm of convolutive sound mixtures based on updating statistical information in the frequency domain, and minimizing the support of time domain demixing filters by a weighted least square method. The permutation and scaling indeterminacies of separation, and concatenations of signals in adjacent time frames are resolved with optimization of $l^1 \times l^\infty$ norm on cross-correlation coefficients at multiple time lags. The algorithm is a direct method without iterations, and is adaptive to the environment. Computations on recorded benchmark mixtures of speech and music signals show excellent performance. The method in general separates a foreground source from a background of sounds as often encountered in realistic situations.

Keywords: Convolutive mixtures; Indeterminacies; Dynamic statistics update; Optimization; Blind separation

1. Introduction

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. However, realistic sound signals are often mixed through a media channel, so the received sound mixtures are linear convolutions of the unknown sources and the channel transmission functions. In simple terms, the observed signals are unknown weighted sums of the signals and its delays. Separating convolutive mixtures is a challenging problem especially in realistic settings.

In this paper, we study *a dynamic BSS method to separate a signal in the foreground from a background of sources*. The method uses both frequency and time domain information of sound signals in addition to the independence assumption on source signals. First, the convolutive mixture in the time domain is decomposed into instantaneous mixtures in the frequency domain by the fast Fourier transform (FFT). At each frequency, the joint approximate diagonalization of eigen-matrices (JADE) method is applied. The JADE method collects second and fourth order statistics from segments of sound signals to form a set of matrices for joint orthogonal diagonalization, which leads to an estimate of demixing matrix and independent sources. However, there remain extra degrees of freedom: permutation and scaling of estimated sources at each frequency. A proper choice of these parameters is critical for the separation quality. Moreover, the large number of samples of the statistical approach can cause delays in processing. These issues are to be addressed by utilizing dynamical information of signals in an optimization framework. We propose to dynamically update statistics with newly received signal frames, then use such statistics to determine permutation in the frequency domain by optimizing an $l^1 \times l^\infty$ norm of channel to channel cross-correlation coefficients with multiple time lags. Though cross-channel correlation functions and related similarity measure were proposed previously to fix permutation [13], they allow cancellations and may not measure similarity as accurately and reliably as the norm

*Corresponding author. Tel.: +1 9498245309; fax: +1 9498247993.
E-mail address: jxin@math.uci.edu (J. Xin).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.12.010

(metric) we introduced here. The freedom in scaling is fixed by minimizing the support of the estimated demixing matrix elements in the time domain. An efficient weighted least square method is formulated to achieve this purpose directly in contrast to iterative method in [17]. The resulting dynamic BSS algorithm is both direct and adapted to the acoustic environment. Encouraging results on satisfactory separation of recorded sound mixtures are reported.

The paper is organized as follows. In Section 2, a review is presented on frequency domain approach, cumulants and joint diagonalization problems and indeterminacies. Then the proposed dynamic method is presented, where objective functions of optimization, statistics update and efficient computations are addressed. Numerical results are shown and analyzed to demonstrate the capability of the algorithm to separate speech and music mixtures in both real room and synthetic environments. Conclusions are in Section 3.

2. Convolutive mixture and BSS

Let a real discrete time signal be $s(k) = [s_1(k), s_2(k), \ldots, s_n(k)]$, k a discrete time index, such that the components $s_i(k)$ $(i = 1, 2, \ldots, n)$, are zero-mean and mutually independent random processes. For simplicity, the processing will divide s into partially overlapping frames of length T each. The independent components are transmitted and mixed to give the observations $x_i(k)$:

$$x_i(k) = \sum_{j=1}^{n} \sum_{p=0}^{P-1} a_{ij}(p)\, s_j(k-p), \quad i = 1, 2, \ldots, n, \tag{2.1}$$

where $a_{ij}(p)$ denote mixing filter coefficients, the pth element of the P-point impulse response from source i to receiver j. The mixture in (2.1) is convolutive, and an additive Gaussian noise may be added. The sound signals we are interested in are speech and music, both are non-Gaussian [1]. We shall consider the case of equal number of receivers and sources, especially $n = 2$.

An efficient way to decompose the nonlocal equation (2.1) into local ones is by a T-point discrete Fourier transform (DFT), $X_j(\omega, t) = \sum_{\tau=0}^{T-1} x_j(t+\tau) \mathrm{e}^{-2\pi J \omega \tau}$, where $J = \sqrt{-1}$, ω is a frequency index, $\omega = 0, 1/T, \ldots, (T-1)/T$, t the frame index. Suppose $T > P$, and extend $a_{ij}(p)$ to all $p \in [0, T-1]$ by zero padding. Let $H_{ij}(\omega)$ denote the matrix function obtained by T-point DFT of $a_{ij}(p)$ in p, $\hat{s}_j(\omega, t)$ the T-point DFT of $s_j(k)$ in the tth frame. If $P \ll T$, then to a good approximation [17]:

$$X(\omega, t) \approx H(\omega) S(\omega, t), \tag{2.2}$$

where $X = [X_1, \ldots, X_n]^{\mathrm{Tr}}$, $S = [\hat{s}_1, \ldots, \hat{s}_n]^{\mathrm{Tr}}$, Tr is short for transpose. The components of S remains independent of each other, the problem is converted to a blind separation of instantaneous mixture in (2.2). Note that P is on the order of 40–50 typically, while T is 256 or 512, so the assumption $P \ll T$ is reasonable.

2.1. Instantaneous mixture and JADE

Let us briefly review an efficient and accurate method, the so-called JADE [2–4] for BSS of instantaneous mixture. There are many other approaches in the literature [6], e.g. info-max method [1] which is iterative and based on maximizing some information theoretical function. JADE is essentially a direct method for reducing covariance. We shall think of S as a random function of t, and suppress ω dependence. First assume that by proper scaling $E[|S_j(t)|^2] = 1$, $j = 1, \ldots, n$. It follows from independence of sources that ($'$ conjugate transpose):

$$E[S(t)S(t)'] = I_n, \quad R_X \equiv E[X(t)X(t)'] = HH', \tag{2.3}$$

the latter identity is a factorization of the Hermitian covariance matrix of the mixture. However, there is nonuniqueness in the ordering and phases of columns of H. Suppose that (1) the mixing matrix H is full rank; (2) the $S_j(t)$'s are independent at any t; (3) the process $S(t)$ is stationary. Let W be a matrix such that $I_n = WR_X W' = WHH'W'$, W is called a whitening matrix. Then WH is an orthogonal matrix, denoted by U. Multiplying W from the left onto (2.2), one finds that

$$Z(t) \equiv WX(t) = US(t). \tag{2.4}$$

The fourth-order statistics are needed to determine U. The fourth-order cumulant of four mean zero random variables is

$$\begin{aligned}\mathrm{Cum}[a, b, c, d] = {} & E(abcd) - E(ab)E(cd) \\ & - E(ac)E(bd) - E(ad)E(bc),\end{aligned} \tag{2.5}$$

which is zero if a, b, c, d split into two mutually independent groups. For source vector S, $\mathrm{Cum}[S_i, S_j, S_k, S_l] = \mathrm{kurt}_i\, \delta_{ijkl}$, $\mathrm{kurt}_i = \mathrm{Cum}[S_i, S_i, S_i, S_i]$ is the kurtosis. If $\mathrm{kurt}_i \neq 0$, the ith source is called kurtic. Kurtosis is zero for a mean zero Gaussian random variable. The last assumption of JADE is that (4) there is at most one nonkurtic source.

Define cumulant matrix set $Q_Z(M)$ from Z in (2.4) as the linear span of the Hermitian matrices $Q = (q_{ij})$ satisfying ($*$ complex conjugate):

$$q_{ij} = \sum_{k,l=1}^{n} \mathrm{Cum}(Z_i, Z_j^*, Z_k, Z_l^*) m_{lk}, \quad 1 \leqslant i, j \leqslant n, \tag{2.6}$$

where matrix $M = (m_{ij}) = e_l e_k'$, e_l being the unit vector with zero components except the lth component equal to one. Eqs. (2.4) and (2.6) imply that (u_p is the pth column of U):

$$Q = \sum_{p=1}^{n} (\mathrm{kurt}_p\, u_p' M u_p) u_p u_p', \quad \forall M, \tag{2.7}$$

or $Q = UDU'$, $D = \mathrm{diag}(\mathrm{kurt}_1\, u_1' M u_1, \ldots, \mathrm{kurt}_n\, u_n' M u_n)$. Hence, U is the joint diagonalizer of the matrix set $Q_Z(M)$. Once U is so determined, the mixing matrix $H = W^{-1}U$. It can be shown [3] using identity (2.7) that the joint diagonalizer of $Q_Z(M)$ is equal to U up to permutation and phase, or up to a matrix multiplier P where P has exactly one unit modulus entry in each row and column. Such a joint diagonalizer is called essentially equal to U.

The algorithm of finding the joint diagonalizer is a generalization of Jacobi method or Givens rotation method [9]. As the cumulant matrices are estimated in practice, exact joint diagonalizer may not exist, instead, an approximate joint diagonalizer, an orthogonal matrix V, is sought to maximize the quantity: $C(V,B)=\sum_{r=1}^{n^2}|\mathrm{diag}(V'B_rV)|^2$, where $B=\{B_1,B_2,\ldots,B_{n^2}\}$ is a set of basis (or eigen) matrices of $Q_Z(M)$, $|\mathrm{diag}(A)|^2$ is the sum of squares of diagonals of a matrix A. Maximizing $C(V,B)$ is same as minimizing off diagonal entries, which can be achieved in a finite number of steps of Givens rotations. The costs of joint diagonalization is roughly n^2 times that of diagonalizing a single Hermitian matrix.

Though stationarity is assumed for the theoretical analysis above, JADE turns out to be quite robust even when stationarity is not exactly satisfied for signals such as speech or music.

2.2. Dynamic method of separating convolutive mixture

For each frequency ω, Eq. (2.2) is a BSS problem of instantaneous mixtures. The speech or music signals in reality are stationary over short time scales and nonstationary over longer time scales, which depend on the production details. For speech signals, human voice is stationary for a few 10 ms, and becomes nonstationary for a time scale above 100 ms due to envelope modulations [7,13]. The short time stationarity permits FFT to generate meaningful spectra in Eq. (2.2) within each frame. For a sampling frequency of 16,000 Hz, each frame of 512 points lasts 32 ms. The mixing matrix H may depend on t over longer time scales, denoted by $H=H(\omega,t)$, unless the acoustic environment does not change as in most synthetic mixing. A demixing method with potential real time application should be able to capture the dynamic variation of mixing.

Our approach consists of six steps summarized in the following pseudo-code. The details are explained in subsequent subsections. The values of the parameters used in numerical experiments are listed in Table 1.

(I) *Take FFT to go to frequency domain*: Collect n_T frames of mixtures, each of length T; take FFT to get data stream $X(\omega,t)$, $t=1,2,\ldots,n_T$.
(II) *Perform separation at each frequency*: Call JADE to separate instantaneous mixtures $X(\omega,t)$ at each ω, and obtain mixing matrix $H_0(\omega)$.

Table 1
Parameters used in both dynamic and batch processing

Case	T	Overlap (%)	n_T (dyn.)	δn_T	Δn_T	K_0	K_1	β	q	n_T (bat.)
(1)	512	0	100	20	30	4	10	1.04	2	200
(2)	256	50	100	20	40	15	20	1.04	2	160
(3)	256	50	100	20	40	10	20	1.04	2	160
(4)	1024	0	120	40	80	14	20	1.04	2	292

(III) *Fix permutation in frequency domain*:
  (a) Find reference frequency ω_1 to minimize $C(\omega)$ in (2.10).
  (b) Fix permutation σ at other frequencies to maximize (2.11).
(IV) *Fix scaling in frequency domain*: Fix scaling factors to minimize the support of the demixing matrix in the time domain by directly solving (2.13).
(V) *Take IFFT to come back to time domain*: Take inverse FFT to find time domain demixing matrix $h^{(0)}(\tau)$. The separated signals $s^{(0)}(\tau)$ are obtained from mixture x convoluting with $h^{(0)}$.
(VI) *Fix permutation in time domain to append latest output*:
  (a) Collect $\delta n_T \ll n_T$ new frames, repeat (I)–(V) with frames δn_T+1 to δn_T+n_T to generate an updated time domain demixing matrix $h^{(1)}(\tau)$ and separated signals $s^{(1)}(\tau)$. The cost reduction techniques mentioned in Section 2.3 can be applied here to accelerate the computation.
  (b) Fix permutation of $s^{(1)}$ to be consistent with $s^{(0)}$ by maximizing (2.14) over their overlapped time interval; normalize maximum norm of $h^{(0)}$ and $h^{(1)}$ to one to maintain continuity in loudness; extend $s^{(0)}$ to δn_T+n_T frames.
  Repeat (VI) till end.

2.2.1. Take FFT and call JADE

Steps (I)–(II) is to find an initialization for $H(\omega,t)$. After receiving the initial n_T frames of mixtures, compute their FFT and obtain $X(\omega,t)$, $t=1,2,\ldots,n_T$, to collect n_T samples at each discrete frequency. For each ω, perform JADE, and estimate the mixing matrix denoted by $H_0(\omega)$. To ensure a good statistical estimate, n_T is on the order of 80–100, and may be properly reduced later.

Steps (I)–(II) give separated components of signals over all frequencies. However, such JADE output has indeterminacies in amplitude, order and phase. This benign problem for instantaneous mixtures becomes a major issue when one needs to assemble the separated individual components. For example, the permutation mismatches across frequencies can degrade the quality of separation seriously.

2.2.2. Fix permutation in frequency domain

Step (III) is to use nonstationarity of signals to sort out a consistent order of separated signals in the frequency domain. Such a method for batch processing was proposed in [13]. A separation method requiring the entire length of the signal is called batch processing. The sorting algorithm of [13] proceeds as follows:

(1) Estimate the envelope variation by a moving average over a number of frames (beyond stationarity time scale) for each separated frequency component. The envelope is denoted by $\mathrm{Env}(\omega,t,i)$, where i is the index of separated components.

(2) Compute a similarity measure equal to the sum of correlations of the envelopes of the separated components at each frequency. The similarity measure is $\mathrm{sim}(\omega)=\sum_{i\neq j}\rho(\mathrm{Env}(\omega,t,i),\mathrm{Env}(\omega,t,j))$, where $\rho(\cdot,\cdot)$ is the normalized correlation coefficients (see (2.9)) involving time average over the *entire signal length* to approximate the ensemble average so the t dependence drops out.
(3) Let ω_1 be the one with lowest similarity value where separation is the best. The ω_1 serves as a reference point for sorting.
(4) At other frequencies ω_k $(k=2,3,\ldots)$, find a permutation σ to maximize $\sum_{i=1}^{n}\rho(\mathrm{Env}(\omega_k,t,\sigma(i)),\sum_{j=1}^{k-1}\mathrm{Env}_s(\omega_j,t,i))$, among all permutations of $1,2,\ldots,n$. Here Env_s denotes the sorted envelopes in previous frequencies.
(5) Permute the order of separated components at the kth frequency bin according to σ in step (4), and define $\mathrm{Env}_s(\omega_k,t,i)$. Repeat (4) and (5) until $k=T$.

We shall modify the above sorting method in three aspects. The first is to use segments of signal instead of the entire signal to compute statistics (correlations) to minimize delay in processing. The second is to use correlation coefficients of separated signals at *un-equal times* or *multiple time lags* in step (2) to better characterize the degree of separation. Moreover, we notice that the similarity measure of [13] as seen above is a sum of correlation coefficients of potentially both signs, and so can be nearly zero due to cancellations even though each term in the sum is not small in absolute value. We introduce an $l^1\times l^\infty$ norm below to characterize more accurately channel similarity by taking sum of absolute values of correlation coefficients and maximum of time lags. The third is to simplify the maximization problem on σ to avoid comparing correlations with summed envelopes at all previous frequencies. We also do not use envelopes of signals inside correlation functions. The reason is that the smoothing nature of envelope operation reduces the amount of oscillations in the signals and may yield correlation values less accurate for capturing the degree of independence. Specifically, let $\hat{s}_i(\omega,t)=a_i(\omega,t)e^{j\phi_i(\omega,t)}$ be the ith separated signal at frequency ω, where $a_i(\omega,t)=|\hat{s}_i(\omega,t)|$, ϕ_i the phase functions, t the frame index. The correlation function of two time dependent signals over M frames is

$$\mathrm{cov}(a(\omega,t),b(\omega',t))=M^{-1}\sum_{t=1}^{M}a(\omega,t)b^*(\omega',t)-M^{-2}\sum_{t=1}^{M}a(\omega,t)\sum_{t=1}^{M}b^*(\omega',t),\quad(2.8)$$

and the (normalized) correlation coefficient is

$$\rho(a(\omega,t),b(\omega',t))=\frac{\mathrm{cov}(a(\omega,t),b(\omega',t))}{\sqrt{\mathrm{cov}(a(\omega,t),a(\omega,t))\mathrm{cov}(b(\omega',t),b(\omega',t))}}.\quad(2.9)$$

From speech production viewpoint, frequency components of a speech signal do not change drastically in time, instead are similarly affected by the motion of the speaker's vocal chords. The correlation coefficient is a natural tool for estimating coherence of frequency components of a speech signal. A similar argument may be applied to music signals as they are produced from cavities of instruments.

Now with $M=n_T$ in (2.8), define

$$C(\omega)=\sum_{i\neq j}\max_{k\in\{-K_0,\ldots,K_0\}}|\rho(|\hat{s}_i(\omega,t)|,|\hat{s}_j(\omega,t-k)|)|,\quad\text{for }\omega\in[\omega_L,\omega_U]\quad(2.10)$$

with some positive integer K_0. Find ω_1 between ω_L and ω_U to minimize $C(\omega)$. With ω_1 as reference, at any other ω, find the permutation σ to maximize:

$$\sigma=\mathrm{argmax}\sum_{i=1}^{n}\max_{k\in\{-K_0,\ldots,K_0\}}|\rho(|\hat{s}_i(\omega_1,t)|,|\hat{s}_{\sigma(i)}(\omega,t-k)|)|.\quad(2.11)$$

Notice that the objective functions in (2.10)–(2.11) are exactly the $l^1\times l^\infty$ norms over the indices $i(j)$ and k. Multiple time lag index k is to accommodate the translational invariance of sound quality to the ear. Maximizing over k helps to capture the correlation of the channels, and sum of i (j) reflects the total coherence of a vector signal.

2.2.3. Fix scaling in frequency domain and IFFT

Step (IV) fixes the scaling and phase indeterminacies in $\hat{s}(\omega,t)$. Each *row* of the demixing matrix $H_0^{-1}(\omega)$ may be multiplied by a complex number $\lambda_i(\omega)$ $(i=1,2\ldots,n)$ before inverse FFT (ifft) to reconstruct demixing matrix $h^{(0)}(\tau)$ in the time domain. The idea is to minimize the support of each row of the inverse FFT by a weighted least square method. In other words, we shall select λ_i's so that the entries of $\mathrm{ifft}(H_0^{-1})(\tau)\equiv h^{(0)}(\tau)$ are real and nearly zero if $\tau\geqslant Q$ for some $Q<T$, Q as small as possible, T being the length of FFT. Smaller Q improves the local approximation, or accuracy of Eq. (2.2). To be more specific, using $H_{0,i}^{-1}(\omega)$ to denote the ith *row vector* of $H_0^{-1}(\omega)$, we can explicitly write the equation to shorten the support of inverse FFT:

$$\mathrm{ifft}(\lambda_i(\omega)H_{0,i}^{-1}(\omega))(\tau)=0\quad(2.12)$$

in terms of the real and imaginary parts of $\lambda_i(\omega)$ for $\omega=0,1/T,\ldots,(T-1)/T$. Those real and imaginary parts are the variables and the equations are *linear*. Now, we let τ run from q to $T-1$. If we want small support, q should be small, then there are more equations than unknowns. So we multiply a weight to each equation and minimize in the least square sense. Eq. (2.12) for larger τ is multiplied by a larger weight in the hope that the value of the left-hand side

of (2.12) will be closer to zero during the least square process. If we choose the weighting function to be the exponential function β^τ for some $\beta>1$, then the above process can be mathematically written as

$$[\lambda_i(0),\ldots,\lambda_i((T-1)/T)] = \operatorname{argmin}\sum_{\tau=q}^{T-1}|\beta^\tau \operatorname{ifft}(\lambda_i(\omega)H_{0,i}^{-1}(\omega))(\tau)|^2, \quad (2.13)$$

where $H_{0,i}^{-1}(\omega)$ is the ith *row vector* of $H_0^{-1}(\omega)$.

A few comments are in order. First, since the mixing matrix $H_0(\omega)$ is the FFT of a real matrix, we impose that $H_0(\omega)=H_0(1-\omega)^*$. So, supposing T is even, we only need to apply JADE to obtain $H_0(\omega)$ for $\omega=0,1/T,\ldots,1/2$; $H_0(0)$ and $H_0(1/2)$ will automatically be real. When fixing the freedom of scaling in each ω, we choose $\lambda(0)$ and $\lambda(1/2)$ real, and $\lambda(\omega)=\lambda(1-\omega)^*$ for other ω. Second, to fix the overall scaling and render the solution nontrivial, we set $\lambda(0)=1$. Third, the weighted least square problem (2.13) can be solved by a direct method or matrix inversion [9, Chapter 6].

Note that when $n=2$, among the $2(T-q)$ equations from (2.12) with $\tau=q,\ldots,T-1$, there are $T-1$ variables including $\lambda_i(1/2)$, the real and imaginary parts of $\lambda_i(\omega)$ for $\omega=1/T,\ldots,1/2-1/T$. So, we can make roughly half of $h_i^{(0)}(\tau)\approx 0$, the best one can achieve in general.

Once $h_i^{(0)}(\tau)=\operatorname{ifft}(\lambda_i(\omega)H_{0,i}^{-1}(\omega))(\tau)$ is obtained, the separated signals, denoted by $s^{(0)}(\tau)$, are then produced by $h^{(0)}*x$ with x being the mixture and $*$ being convolution. Again, we use $\tau\in[0,Tn_T]$ as the time index, because we have used $t\in[0,n_T]$ as the frame index.

2.2.4. Update and append latest output

The last Step (VI) is to update $h^{(0)}(\tau)$ and expand $s^{(0)}(\tau)$ when $\delta n_T \ll n_T$ many new frames of mixtures arrive. Steps (I)–(V) are repeated using frames from δn_T+1 to δn_T+n_T, to generate a new time domain demixing matrix $h^{(1)}(\tau)$, $\tau\in[0,T-1]$, and separated signal $\tilde{s}^{(1)}(\tau)$, $\tau\in[T(n_T-\Delta n_T)+1,T(n_T+\delta n_T)]$ with T the size of one frame. Now, $\tilde{s}^{(1)}(\tau)$ and $s^{(0)}(\tau)$ share a common interval of size $T\Delta n_T$. On this common interval, $\tilde{s}^{(1)}(\tau)$ and $s^{(0)}(\tau)$ will be the same if we are doing a perfect job and if the ordering of $\tilde{s}^{(1)}$ is consistent with that of $s^{(0)}$. In order to determine the ordering of $\tilde{s}^{(1)}(\tau)$, we compute $\rho(\tilde{s}_i^{(0)}(\tau),s_j^{(1)}(\tau-k))$ on this common interval with different k and $i,j=1,\ldots,n$. Then we determine the permutation σ of the components of $\tilde{s}^{(1)}(t)$ by minimization:

$$\sigma=\operatorname{argmax}\sum_{i=1}^{n}\max_{k\in\{-K_1,\ldots,K_1\}}|\rho(s_i^{(0)}(\tau),\tilde{s}_{\sigma(i)}^{(1)}(\tau-k))| \quad (2.14)$$

with some constant K_1. After doing the necessary permutation of $\tilde{s}^{(1)}$, the separated signals are then extended to the extra frames δn_T+n_T by concatenating the newly separated δn_T many frames of $\tilde{s}^{(1)}$ with those of $\tilde{s}^{(0)}$. The continuity of concatenation is maintained by requiring that $\max_\tau|h_{ii}^{(k)}(\tau)|$'s $(i=1,2,\ldots,n)$ are invariant in k, where $k=1,2,\ldots,$ labels the updated filter matrix in time. The procedure repeats with the next arrival of mixture data, and is a direct method incorporating dynamic information.

Because sorting order depends only on the relative values of channel correlations, we observed in practice that the $\max_{k\in\{-K_\cdot,\ldots,K_\cdot\}}$ in Eqs. (2.10), (2.11), (2.14) may be replaced by $\sum_{k=-K_\cdot}^{K_\cdot}$, with a different choice of $K_\cdot$ value. The $\max_{k\in\{-K_\cdot,\ldots,K_\cdot\}}$ is a more accurate characterization however.

2.2.5. Remarks

We remark that Steps (II) and (III) are similar to the treatment in [13] which used a second-order decorrelation method and Givens rotations. Instead we use JADE because its quality of separation is in general better than second-order methods for instantaneous mixtures. A second-order direct method may save more computing time; however, here, we focus on separation of convolutive mixtures, and so optimize on quality of Step (II). JADE also has the advantage of being robust to independent Gaussian noises, and helps the algorithm to perform on noisy data. In Step (III), we corrected the similarity measure [13, Eq. (38)] into a norm (nonnegative σ in Eq. (2.11), and simplified the permutation sorting involved.

Step (IV) is entirely new, and replaces the iterative scaling fixing method of Parra and Spence [17] by a direct method of weighted least square. Step (VI) is not needed for batch methods, and is introduced here for dynamic adaptivity of our method.

2.3. Adaptive estimation and cost reductions

Cumulants and moments are symmetric functions in their arguments [15]. For example when $n=2$, there are 16 joint fourth-order cumulants from (2.5), however, only six of them need to be computed, the others follow from symmetry. Specifically, among the 16 cumulants:

$$Q(1)=\operatorname{Cum}(y_1,y_1^*,y_1^*,y_1),\quad Q(2)=\operatorname{Cum}(y_1,y_1^*,y_1^*,y_2),$$
$$Q(3)=\operatorname{Cum}(y_1,y_1^*,y_2^*,y_1),\quad Q(4)=\operatorname{Cum}(y_1,y_1^*,y_2^*,y_2),$$
$$Q(5)=\operatorname{Cum}(y_1,y_2^*,y_1^*,y_1),\quad Q(6)=\operatorname{Cum}(y_1,y_2^*,y_1^*,y_2),$$
$$Q(7)=\operatorname{Cum}(y_1,y_2^*,y_2^*,y_1),\quad Q(8)=\operatorname{Cum}(y_1,y_2^*,y_2^*,y_2),$$
$$Q(9)=\operatorname{Cum}(y_2,y_1^*,y_1^*,y_1),\quad Q(10)=\operatorname{Cum}(y_2,y_1^*,y_1^*,y_2),$$
$$Q(11)=\operatorname{Cum}(y_2,y_1^*,y_2^*,y_1),\quad Q(12)=\operatorname{Cum}(y_2,y_1^*,y_2^*,y_2),$$
$$Q(13)=\operatorname{Cum}(y_2,y_2^*,y_1^*,y_1),\quad Q(14)=\operatorname{Cum}(y_2,y_2^*,y_1^*,y_2),$$
$$Q(15)=\operatorname{Cum}(y_2,y_2^*,y_2^*,y_1),\quad Q(16)=\operatorname{Cum}(y_2,y_2^*,y_2^*,y_2),$$

we have the relations: $Q(2)=Q(3)^*=Q(5)^*=Q(9)$, $Q(4)=Q(6)=Q(11)=Q(13)$, $Q(7)=Q(10)^*$, $Q(8)=Q(15)=Q(12)^*=Q(14)^*$, where $*$ is complex conjugate. For N samples, we only need to compute the following six

$1 \times N$ vectors:

$$Y_1 = (y_1^1 y_1^1, \ldots, y_1^N y_1^N), \quad Y_2 = (y_1^1 y_2^1, \ldots, y_1^N y_2^N),$$

$$Y_3 = (y_2^1 y_2^1, \ldots, y_2^N y_2^N), \quad Y_4 = (y_1^1 y_1^{1*}, \ldots, y_1^N y_1^{N*}),$$

$$Y_5 = (y_1^1 y_2^{1*}, \ldots, y_1^N y_2^{N*}), \quad Y_6 = (y_2^1 y_2^{1*}, \ldots, y_2^N y_2^{N*}),$$

then all the fourth-order and second-order statistical quantities can be reconstructed. For example,

$$Q(1) = \frac{1}{N} Y_4 \cdot Y_4^{\mathrm{Tr}} - \frac{1}{N^2}(2\mathrm{sum}(Y_4)\mathrm{sum}(Y_4) + \mathrm{sum}(Y_1)(\mathrm{sum}(Y_1)^*)), \tag{2.15}$$

where $\mathrm{sum}(Y_i)$ is the summation of the N components of Y_i.

As formula (2.5) suggests, cumulants are updated through moments when δn_T early samples are replaced by the same number of new samples. As δn_T is much less than the total number of terms n_T in the empirical estimator of expectation, the adjustment costs $2\delta n_T$ flops for each second moments and $6\delta n_T$ flops for each joint fourth-order moment. The contributions of the early samples are subtracted from the second and fourth moments, then the contributions of the new samples are added. The cumulant update approach is similar to cumulant tracking method of moving targets ([12] and references therein).

Due to dynamical cumulants update, the prewhitening step at each frequency is performed after cumulants are computed from $X(\omega)$. This is different from JADE [3] where the prewhitening occurs before computing the commulants. This way, it is more convenient to make use of the previous cumulant information and updated $X(\omega)$. Afterward, we use the multilinearity of the cummulants to transform them back to the commulants of the prewhitened $X(\omega)$, before joint diagonalization.

It is desirable to decrease n_T to lower the number of samples for cumulants estimation. However, this tends to increase the variance in the estimated cumulants, and render estimation less stable in time. Numerical experiments indicated that with n_T as low as 40, the separation using overlapping frames is still reliable with reasonable quality.

It is known [7] that the identity of a speaker is carried by pitch (perception of the fundamental frequency in speech production) which varies in the low frequency range of a few hundred Hertz. We found that instead of searching among all frequencies for the reference frequency ω_1 in Step (III)(a), it is often sufficient to search in the low frequency range. The smaller searching range alleviates the workload in sorting and permutation correcting. This is similar to a feature oriented method, see [16,18,5] among others.

2.4. Experimental results

The proposed algorithm with adaptivity and cost reduction considerations was implemented in Matlab. The original code of JADE by J.-F. Cardoso is obtained from a open source (http://web.media.mit.edu/~paris/) maintained by P. Smaragdis. Separation results with both dynamic and batch processing of four different types of mixtures are reported here:

(1) real room recorded speech–music data;
(2) synthetic mixture of speech and music;
(3) synthetic mixture of speech and speech noise;
(4) real room recorded speech–speech data in noisy environment.

They will be called cases (1)–(4) in the following discussion. Cases (1) and (4) have been studied by Lee [11] and our results will be compared with his.

The values of the parameters in our computation are listed in Table 1. In the table, "n_T (dyn.)" is the initial value of n_T in dynamic processing and "n_T (bat.)" is the n_T in batch processing. Other than n_T, dynamic and batch process share the same parameters. The frame size is T, "overlap" is the overlapping percentage between two successive frames, δn_T and Δn_T are as in Step (VI), K_0 and K_1 are from (2.11) and (2.14), β is in (2.13), and q is the lower limit of τ in (2.12).

Note that the values of ω_L and ω_U from (2.10) are not listed in the table. In our computation, we use the following two choices:

(A) $\omega_L = 0$, $\omega_U = \frac{1}{2}$.
(B) $\omega_L = \omega_U = 4/T$, namely fixing reference frequency $\omega_1 = 4/T$.

For the four cases reported in this paper, *both choices work and generate very similar results*. As a consequence, we will only plot the results of the first choice. The first choice is more general while the second is motivated by the pitch range of speech signal and is computationally more favorable. However, we do not know precisely the robustness of the latter.

2.4.1. Computational results

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

$$\bar{\rho}(a,b) = \max_{k \in \{-K_2, \ldots, K_2\}} |\rho(a(\tau), b(\tau + k))| \tag{2.16}$$

with ρ defined in (2.9). The $\bar{\rho}$ is computed for the mixtures, the sources and the separated signals for both batch and dynamic processing. Exceptions are the lack of sources in cases (1) and (4), where we will compute the $\bar{\rho}$ for Lee's separation results [11]. We choose $K_2 = 20$ in all the computations. The results are listed in Table 2 which shows that the $\bar{\rho}$ values of the mixtures are much larger than those

of the dynamically separated signals, which are on the same order as the $\bar{\rho}$ values of the batch separated signals. The $\bar{\rho}$ value of Lee's separation results are 0.0354 and 0.0079 for cases (1) and (4), which are of the same order as ours. In the synthetic cases (2) and (3), the $\bar{\rho}$ values of the batch separated signals are on the same order of the $\bar{\rho}$ values of the source signals or 10^{-2}. In cases (2) and (3), we use the ratio $\bar{\rho}(y,s_1)/\bar{\rho}(y,s_2)$ to measure the relative closeness of a signal y to source signals s_1 and s_2. Table 3 lists these ratios for y being the separated signals by dynamic and batch methods with A and B denoting the two ways of setting the reference frequency ω_1. The outcomes are similar no matter $y=\tilde{s}_1$ or $y=\tilde{s}_2$ (first or second separated signal) in either dynamic or batch cases and either way of selecting the reference frequency ω_1.

Table 2
Values of the correlation coefficient $\bar{\rho}(y,z)$, (y,z) being either the two mixtures or the two sources or the two separated signals

$\bar{\rho}(\cdot,\cdot)$	Mixture	Dyn. separation	Bat. separation	Sources/Lee's
(1)-A	0.8230	0.0269	0.0160	0.0354 (L)
(1)-B	0.8230	0.0225	0.0159	0.0354 (L)
(2)-A	0.6240	0.0503	0.0673	0.0201 (S)
(2)-B	0.6240	0.0182	0.0600	0.0201 (S)
(3)-A	0.4613	0.0351	0.0378	0.0243 (S)
(3)-B	0.4613	0.0267	0.0677	0.0243 (S)
(4)-A	0.5466	0.0067	0.0214	0.0079 (L)
(4)-B	0.5466	0.0078	0.0137	0.0079 (L)

The A and B in the first column denote the two different ways of selecting the reference frequency ω_1. Lee's separation results (L) substitute for sources (S) in case of room recordings.

Table 3
Ratios of $\bar{\rho}(y,s_1)$ and $\bar{\rho}(y,s_2)$, y being a separated signal on the first column by dynamic or batch method, s_1 and s_2 are source signals

$\bar{\rho}(y,s_1)/\bar{\rho}(y,s_2)$	Case (2)	Case (3)
$y=$ dyn. $\tilde{s}_1$(A)	4.5899	4.5096
$y=$ dyn. $\tilde{s}_2$(A)	0.1086	0.2852
$y=$ dyn. $\tilde{s}_1$(B)	5.3083	5.8411
$y=$ dyn. $\tilde{s}_2$(B)	0.0494	0.2799
$y=$ bat. $\tilde{s}_1$(A)	15.0912	1.4632
$y=$ bat. $\tilde{s}_2$(A)	0.0760	0.1665
$y=$ bat. $\tilde{s}_1$(B)	6.2227	25.8122
$y=$ bat. $\tilde{s}_2$(B)	0.0636	0.1719

The ratio measures the relative closeness of y to s_1 and s_2. If the ratio is larger (smaller) than one, y is closer to s_1 and (s_2). The A and B in the first column denote the two different ways of selecting ω_1.

In case (1), the recorded data [11] consists of two mixtures of a piece of music (source 1) and a digit (1–10) counting sentence (source 2) recorded in a normal office size room. The sampling frequency is 16 kHz, and about 100k data points are shown in Fig. 1. The signals last a little over 6 s. The result of dynamic BSS algorithm is shown in Fig. 2. As a comparison, we show in Fig. 3 result of batch processing of Steps (I)–(V) of the algorithm with $n_T=200$. The batch processing gives a clear separation upon listening to the separated signals. The dynamic processing is comparable. The filter coefficients in the time domain $h_{ij}(\tau)$ at the last update of dynamic processing are shown in Fig. 4. Due to weighted least square optimization in Step (IV), they are localized and oscillatory with support length Q close to half of the FFT size T.

For cases (2) and (3), we show the *envelopes of the absolute values* of the mixtures or the separated signals. The signal envelope was computed using the standard procedure of amplitude demodulation, i.e. low-pass filtering the rectified signal. The filter was an FIR filter with 400 taps and the cutoff frequency was 100 Hz. Signal envelopes help to visualize and compare source and processed signals. We have normalized all the envelopes so that the maximum height is 1. The values of a_{ij} in (2.1), which are used to synthetically generate the mixtures, are shown in Fig. 5

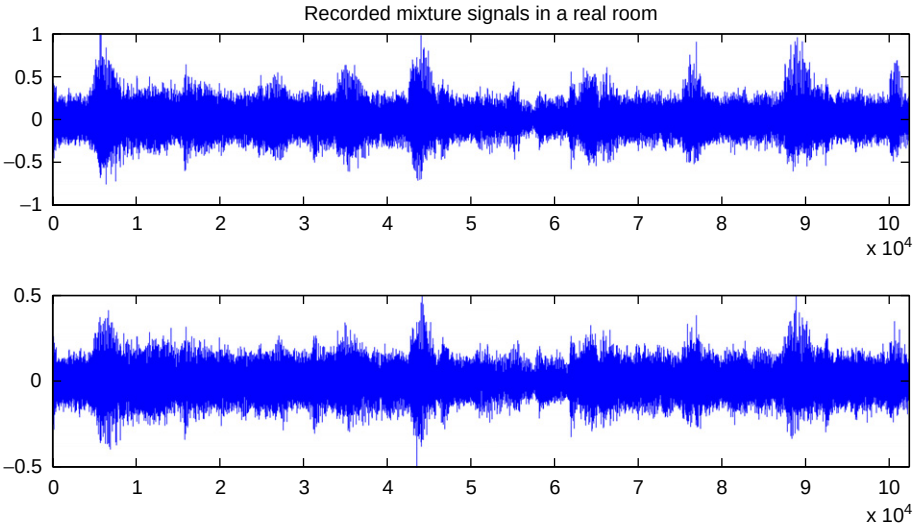


Fig. 1. Case (1), two recorded signals in a real room where a speaker was counting 10 digits with music playing in the background.

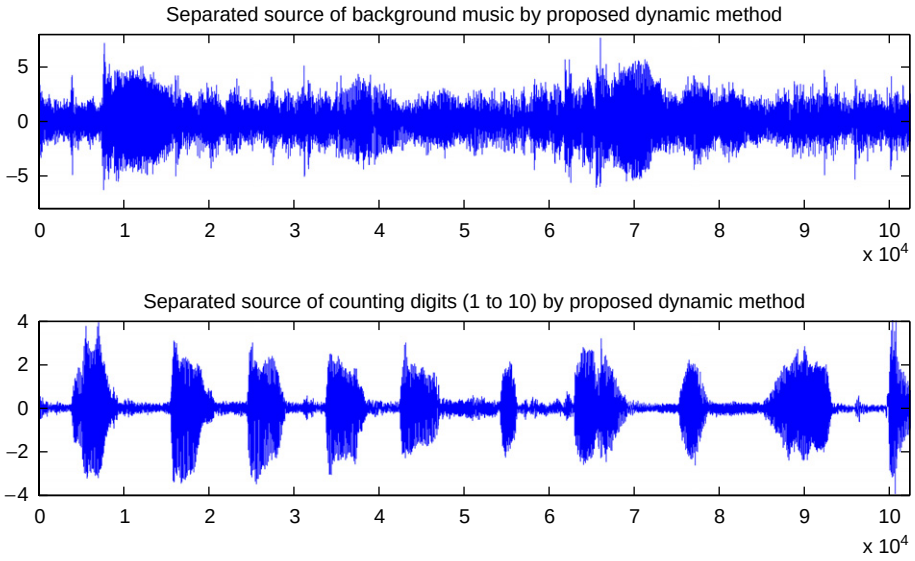


Fig. 2. Case (1) with choice A, separated digit counting sentence (bottom) and background music (top) by the proposed dynamic method. Choice B gives similar results.

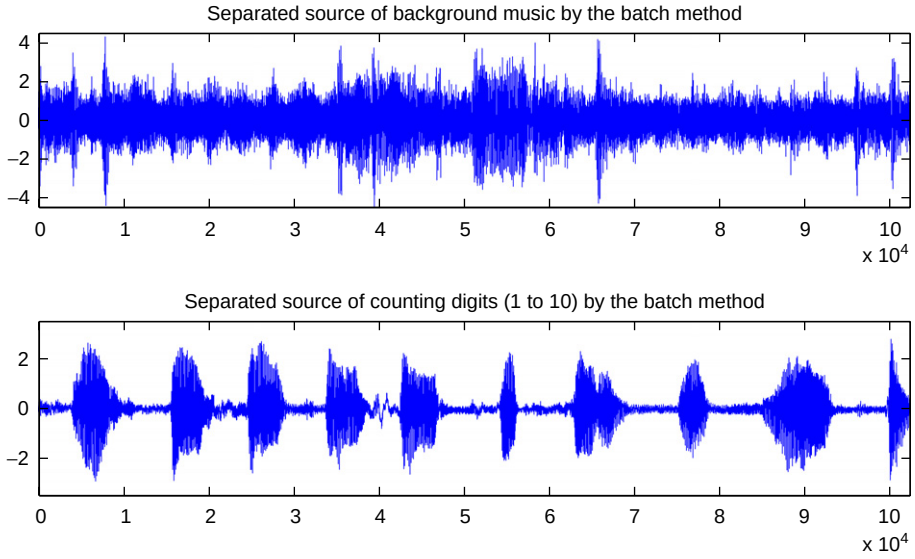


Fig. 3. Case (1) with choice A, separated digit counting sentence (bottom) and background music (top) by batch processing using the proposed Steps (I)–(V). Batch processed signals sounds a little smoother. Choice B gives similar results.

(see [19, (8)]). Figs. 6 and 7 show the mixtures and separated signals of case (2). Figs. 8 and 9 show the mixtures and separated signals of case (3). In view of these plots, Tables 2 and 3, separation is quite satisfactory, which is also confirmed by hearing the separated signals.

In case (4), we computed Lee's recorded speech–speech mixtures in a conference room with air-conditioning noise [11]. The last row of Table 2 showed the $\bar{\rho}$ values of our dynamic (batch) separation and those of Lee's info-max type method [11]. A typical demixing filter matrix in the time domain is shown in Fig. 10. Our filter size is 1024, while Lee's is 2048. Upon hearing the separated signals, our results and Lee's are quite similar.

2.4.2. Further remarks and comparison with info-max

We summarize in Table 4 the time consumption in the batch algorithm. The difference between the total time and the time for JADE + switch + scaling is the time for FFT, IFFT, convolution and others. Dynamic processing takes slightly longer due to additional updates. These runs were done in Matlab on a Compaq Presario V6133CL notebook with 1.6 GHz CPU and 2G bytes of memory. The processing time can be reduced (much) further if one uses C/C++ or Fortran instead of Matlab.

In Step (II), JADE can be replaced by other instantaneous separation method such as AMUSE [6] or info-max [1]. The advantage of JADE comparing with the others is

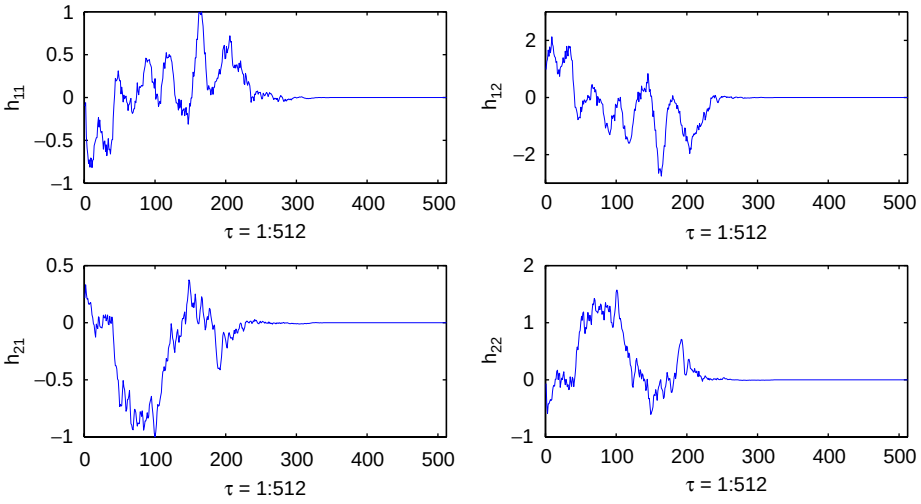


Fig. 4. Case (1) with choice A, the localized and oscillatory filter coefficients in the time domain at the last frame of dynamic processing. Choice B gives similar results.

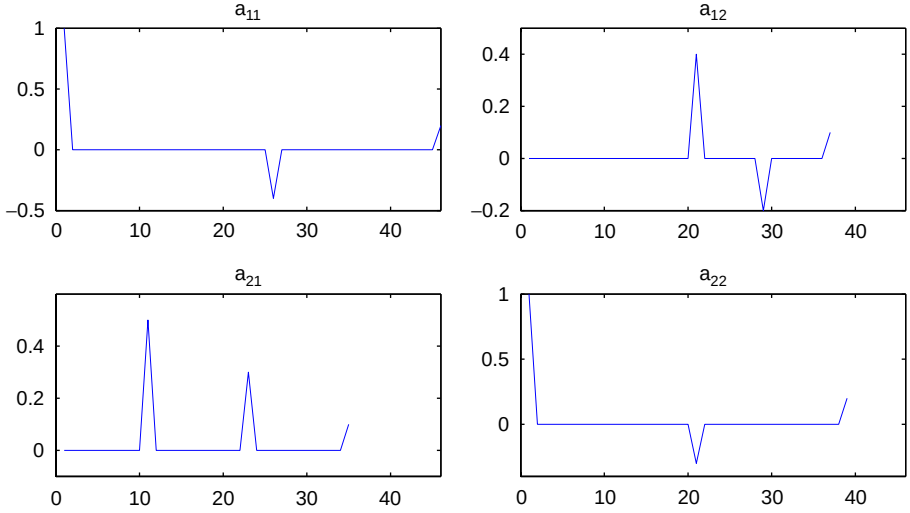


Fig. 5. The weights a_{ij} used in generating synthetic mixtures of cases (2) and (3), as proposed in [19].

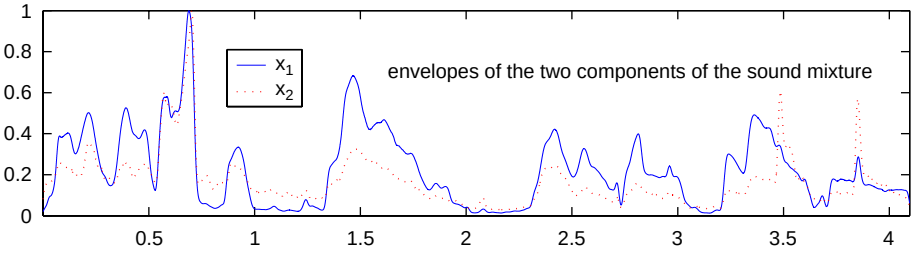


Fig. 6. Case (2), the synthetic mixtures are generated by a female voice and a piece of instrumental music.

the quality and robustness. Unlike iterative methods such as info-max, our method is a direct method so that: (1) no prior knowledge of the probability distribution of the sources is needed; (2) no initial guess of demixing filter is required; (3) the algorithm is dynamically stable. Info-max method is a time domain method, computationally simple and low cost, though certain choices may have to be made by users. As reported in [8], it may need a judicious choice of initial data and a proper step size to avoid divergence in iteration. In [8], an a posteriori scalar gradient constraint is proposed to alleviate such problems. The price to pay in our method for the benefits (1)–(3) is that steps (III) and

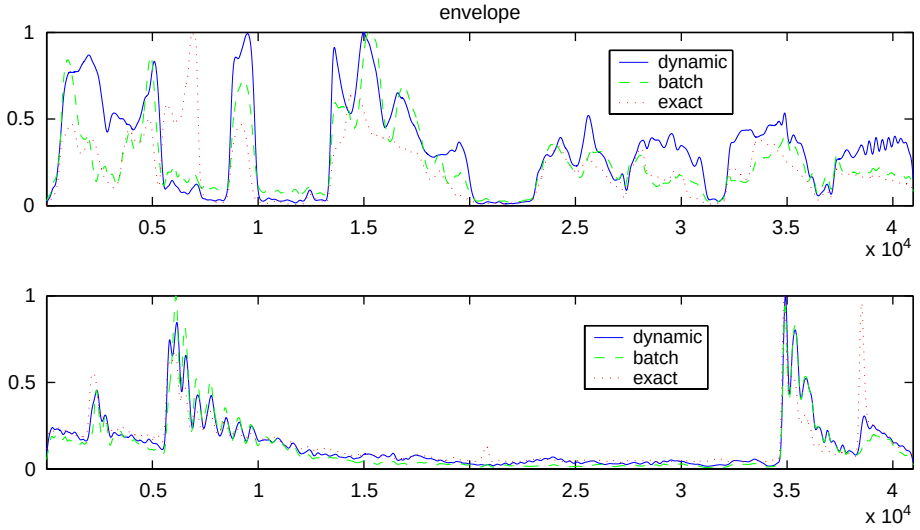


Fig. 7. Case (2) with choice A, the envelopes of the separated signals from mixtures whose envelopes are in Fig. 6. The small amplitude portion of the music is well recovered. Choice B gives similar results.

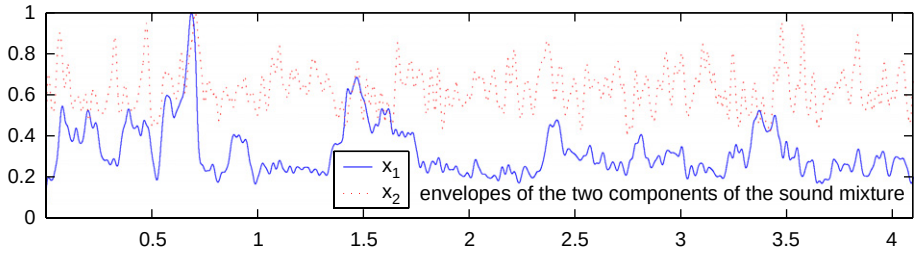


Fig. 8. Case (3), the synthetic mixtures of a female voice and a speech noise with signal-to-noise ratio equal to −3.8206 dB. The x_1 plot shows a speech in a strong noise, the valley structures in the speech signal are filled by noise.

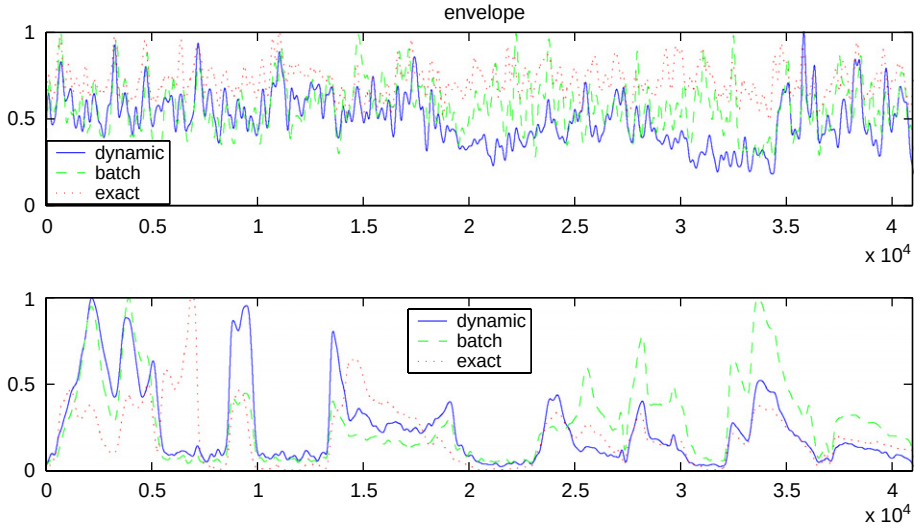


Fig. 9. Case (3) with choice A, the envelopes of the separated signals, noise (top) and speech (bottom). The envelopes of the two mixtures are in Fig. 8. The strongly noisy x_1 in Fig. 8 has been cleaned, the valleys in the envelope re-appeared. Choice B gives an even better result.

(IV) require more data processing and computation time. The data processing may be improved by various strategies as noted in the discussion of Section 2.3. Much future work is needed here.

Steps (III) (permutation) and (IV) (scaling) are necessary for separation to work, in particular Step (IV). Our numerical experiments suggest that Step (IV) is critical. Dropping Step (IV) can lead to outputs that sound no

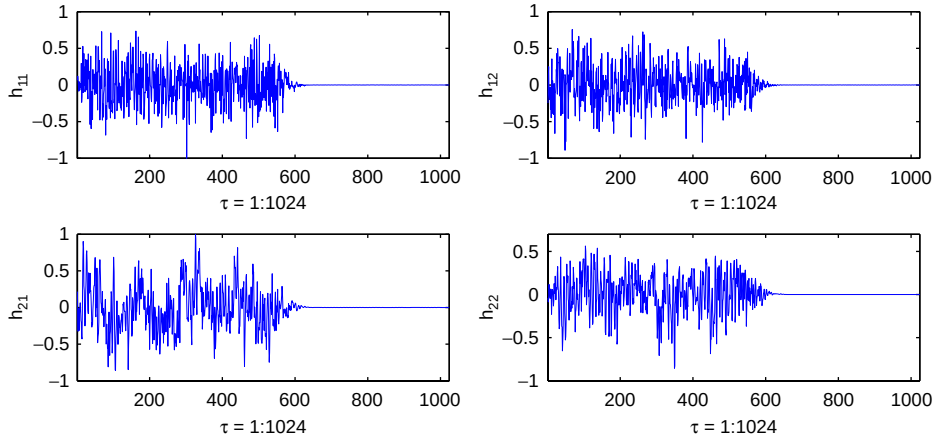


Fig. 10. A typical demixing filter weight matrix in the time domain for Case (4).

Table 4
Typical time consumption (in seconds) in the batch algorithm

	JADE	Switch	Scaling	Total	Signal duration
(1)-A, batch	2.250	0.265	1.188	4.969	6.875
(4)-A, batch	4.547	2.015	8.829	22.110	18.750

The processing times are comparable with the signal durations.

better than the original mixtures. Without Step (III), sometimes we can still get separated signals, for example, in case (1)-dyn. Even though such results are not as good as those with Step (III), they are better than those without Step (IV). In general, dropping Step (III) will also affect separation quality.

Step (VI) is not computationally demanding and may be ignored, e.g. in case (1)-dyn with the current parameters. In general, it is helpful for eliminating possible mis-matches when joining signal segments.

The output from our algorithm is rather stable for parameters in the ranges specified in Table 1. For example, in case (1), the parameters in Table 1 can be changed to other values in the same column without affecting much the separation, except the value of K_0. The parameter K_0 is more sensitive. For example, in case (1)-batch, with other parameters fixed, taking $K_0 = 15$ yields similar results, while $K_0 = 10$ deteriorates results. It appears that choosing K_0 near the lower (upper) end of its range is a better choice. However, there is no guarantee that a single preset value of K_0 is optimal for different mixture signals. This may be an indication of the limit of accuracy of statistical approach.

3. Conclusions

A dynamic BSS algorithm is proposed to track the time dependence of signal statistics and to be adaptive to the potentially time varying environment. Besides efficiently updating cumulants, the method made precise the procedure of sorting permutation indeterminacy in the frequency domain by optimizing a metric (the $l^1 \times l^\infty$ norm) on multiple time lagged channel correlation coefficients. A direct and efficient weighted least square approach is introduced to compactify the support of demixing filter to improve the accuracy of frequency domain localization of convolutive mixtures. Experimental results show robust and satisfactory separation of real recorded data and synthetic mixtures. An interesting line of future work will be concerned with various strategies to speed up computation.

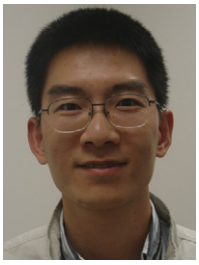
Acknowledgments

The authors thank the three anonymous referees for their valuable comments which improved the presentation of this paper. The work was partially supported by NSF Grants ITR-0219004, DMS-0549215, 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, T. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.
[2] J.-F. Cardoso, Blind signal separation: statistical principles, Proc. IEEE 9 (10) (1998) 2009–2025.
[3] J.-F. Cardoso, A. Souloumiac, Blind beamforming for non-Gaussian signals, IEEE Proc.-F 140 (6) (1993) 362–370.
[4] J.-F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization, SIAM J. Matrix Anal. 17 (1996) 161–164.
[5] T. Chan, C. Wong, Total variation blind deconvolution, IEEE Trans. Image Process. 7 (1998) 370–375.
[6] S. Choi, A. Cichocki, H. Park, S. Lee, Blind source separation and independent component analysis: a review, Neural Inf. Process.—Lett. Rev. 6 (1) (2005) 1–57.
[7] L. Deng, D. O'Shaughnessy, Speech Processing—A Dynamic and Optimization-Oriented Approach, Marcel Dekker Inc., New York, 2003, 626pp.
[8] S.C. Douglas, M. Gupta, Scaled natural gradient algorithms for instantaneous and convolutive blind source separation, in: IEEE ICASSP, 2007, pp. II637–II640.
[9] G. Golub, C. Van Loan, Matrix Computations, John Hopkins University Press, Baltimore, MD, 1983.

[11] T.-W. Lee, Blind source separation: audio examples ⟨http://inc2.ucsd.edu/~tewon⟩.
[12] T. Liu, J. Mendel, Cumulant-based subspace tracking, Signal Process. 76 (1999) 237–252.
[13] N. Murata, S. Ikeda, A. Ziehe, An approach to blind separation based on temporal structure of speech signals, Neurocomputing 41 (2001) 1–24.
[15] C. Nikias, A. Petropulu, Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework, in: A. Oppenheim (Ed.), Prentice-Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[16] S. Osher, L. Rudin, Feature-oriented image enhancement using shock filters, SIAM J. Numer. Anal. 27 (4) (1990) 919–940.
[17] L. Parra, C. Spence, Convolutive blind separation of non-stationary sources, IEEE Trans. Speech Audio Process. 8 (5) (2000) 320–327.
[18] Y. Qi, J. Xin, A perception and PDE based nonlinear transformation for processing spoken words, Physica D 149 (2001) 143–160.
[19] K. Torkkola, Blind separation of convolved sources based on information maximization, Neural Networks Signal Process. VI (1996) 423–432.



Jie Liu received B.S. degree in Applied Mathematics from Tsinghua University in 2000 and Ph.D. degree in Applied Mathematics and Scientific Computation from the University of Maryland at College Park in 2006. He is currently a visiting assistant professor at the University of California at Irvine. His research interests include computational fluid dynamics and blind source separation.

Jack Xin received his B.S. in Computational Mathematics at Peking University in 1985, M.S. and Ph.D. in Applied Mathematics at New York University in 1988 and 1990. He was a post-doctoral fellow at Berkeley and Princeton in 1991 and 1992. He was an assistant and associate professor of mathematics at the University of Arizona from 1991 to 1999. He was a professor of mathematics from 1999 to 2005 at the University of Texas at Austin. He has been a professor of mathematics in the Department of Mathematics, Center for Hearing Research, Institute for Mathematical Behavioral Sciences, and Center for Mathematical and Computational Biology at UC Irvine since 2005. He is a fellow of the John S. Guggenheim Foundation. His research interests include applied analysis, numerical methods and their applications in nonlinear and multiscale dynamics in fluids, mathematical modelling in speech and hearing sciences, and sound signal processing.

Yingyong Qi has been a specialist at the Mathematics Department of UC Irvine since 2005, and a Principal Engineer of Qualcomm at San Diego since 1999. From 1989 to 1999, he was an associate professor, jointly appointed by the Department of Speech and Hearing Sciences and Department of Electrical Engineering, at the University of Arizona. He received his bachelor degree in Physics at the University of Science and Technology of China in 1983 and his master degree in Acoustics at the Institute of Acoustics, Chinese Academy of Sciences in 1985. He received his first Ph.D. in Speech Science at the Ohio State University in 1989, and his second Ph.D. in Electrical Engineering at the University of Arizona in 1993. His major research interests include multimedia technologies on embedded systems, multimedia signal processing, and acoustics of speech production and perception. In addition to many grant awards from various US research agencies, he received the Klatt Memorial Award in Speech Science from the Acoustical Society of America in 1991, the AASFAA Outstanding Faculty Award for excellence in research, teaching and community service from the University of Arizona in 1998, and several QUALSTAR Awards for exceptional contributions to Qualcomm. He was a member of Acoustical Society of America and Institute of Electrical and Electronics Engineers.