TIME-DOMAIN BLIND AUDIO SOURCE SEPARATION USING ADVANCED COMPONENT CLUSTERING AND RECONSTRUCTION

Zbyněk Koldovský

Technical University of Liberec,
Studentská 2, 461 17 Liberec,
Czech Republic
zbynek.koldovsky@tul.cz

Petr Tichavský

Institute of Information Theory and
Automation, P. O. Box 18, 182 08 Prague 8,
Czech Republic
tichavsk@utia.cas.cz

ABSTRACT

We present a novel time-domain method for blind separation of convolutive mixture of audio sources (the cocktail party problem). The method allows efficient separation with good signal-to-interference ratio (SIR) and signal-to-distortion ratio (SDR) using short data segments only. In practice, we are able to separate 2-4 speakers from audio recording of the length less than 6000 samples, which is less than 1 s in the 8 kHz sampling. The average time needed to process the data with filter of the length 20 was 2.2 seconds in Matlab v. 7.2 on an ordinary PC with 3GHz processor.

1. INTRODUCTION

The cocktail-party problem is an acoustic blind deconvolution task where d independent audio sources should be retrieved from their unknown convolutive mixtures recorded by m microphones. Owing to propagation of sound in a natural acoustic environment, the mixing process is described by

$$x_i(n) = \sum_{j=1}^{d} \sum_{\tau=0}^{M_{ij}} h_{ij}(\tau) s_j(n-\tau), \quad i = 1, \ldots, m, \tag{1}$$

where $x_1(n), \ldots, x_m(n)$ are the observed signals on microphones, $s_1(n), \ldots, s_d(n)$ are the unknown original (audio) signals, and h_{ij}'s are source-sensor impulse responses each of length M_{ij}.

To retrieve the original signals without any prior knowledge is not possible, in general. However, a blind separation might be possible using an assumption of mutual independence of the original sources, which usually fits well real-world conditions. Thus, the separation can be done by methods of the Independent Component Analysis (ICA).

Audio sources have generally unknown temporal structure, which introduces ambiguity into solutions of their blind separation. The main goal is therefore to estimate responses of the sources at microphones rather than estimating the sources themselves. For the kth source and the ith microphone, the response is

$$s_i^k(n) = \sum_{\tau=0}^{M_{ij}} h_{ij}(\tau) s_k(n-\tau), \tag{2}$$

and is estimated as an output of a MISO filter of length L

$$\widehat{s}_i^k(n) = \sum_{j=1}^{m} \sum_{\tau=0}^{L-1} w_{ij}^k(\tau) x_j(n-\tau). \tag{3}$$

In this paper, we present an algorithm that solves the cocktail-party problem in the above described sense.

2. SEPARATION PROCEDURE

Our method comes from a prototype of the time-domain approach proposed in [1], and works as follows (cf. Fig. 1)

1. An ICA transform (done by an ICA method) is applied to a $m \cdot L$ dimensional signal subspace spanned by elements of

$$\mathbf{x}(n) = [x_1(n), x_1(n-1), \ldots, x_1(n-L+1), \\ x_2(n), \ldots \quad \ldots, x_m(n-L+1)]^T. \tag{4}$$

   Resulting *(independent) components* can be understand as outputs of $m \cdot L$ MISO FIR filters of length L, selected so that the outputs are mutually independent as much as possible.

2. The components are grouped using a clustering algorithm so that components sharing a common cluster are (ideally) filtered versions each of other and belong to the same original audio source.

3. A reconstruction procedure is applied to clusters of components to get the individual responses (2).

[0] This work was supported by Ministry of Education, Youth and Sports of the Czech Republic through the project 1M0572 and through the Grant 102/07/P384 of the Grant Agency of the Czech Republic.

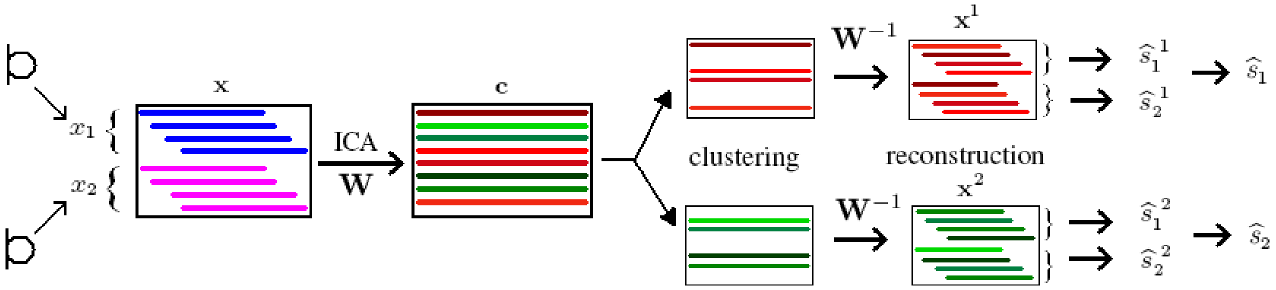

Fig. 1. Illustration of how the presented separation procedure processes batch of signals from two microphones and results in two separated sources. This is the case where the binary weighting is used in the reconstruction.

4. Finally, a delay-and-sum beamformer can be applied to the responses to yield the final original source estimate.

The following three sections are devoted to details of Steps 1-3 of our procedure; see also illustration in Figure 1.

3. STEP 1: ICA DECOMPOSITION

In general, components of (4) obtained via ICA are expected to have their mutual spatio-temporal interference cancelled as much as possible. In an ideal case, they are differently filtered copies of independent original sources $s_1(n), \ldots, s_d(n)$. Three basic classes of ICA algorithms could be considered to realize the ICA decomposition.

The first class, represented by JADE, FastICA, Infomax and recently EFICA, is based on non-Gaussianity of the original sources; cf. [2] and the references therein. The second class, represented by SOBI and WASOBI, relies on distinct coloration of the sources (spectral diversity) [4], and the third class, represented by BGL (Block Gaussian Likelihood) [3], is based on their non-stationarity. While the first class uses higher-order statistics (nonlinear transformations) of the data, the other two classes are based on second-order statistics. The BGL consists in dividing the received signals in certain number of non-overlapping segments, computing signal covariance matrices on each segment, and an approximate joint diagonalization (AJD) of these matrices.

In [1], the EFICA algorithm was shown to be very good ICA transform of (4). This paper concerns utilization of a fast BGL algorithm implementation[1] that uses a novel AJD algorithm proposed in [4]. This implementation allows the ICA of data of length 6000 samples with $m = 2$, $L = 20$, and by considering 20 blocks in about one second[2], which corresponds to AJD of 20 matrices of the dimension 40×40. EFICA is slower and can do similar separation in 6 s.

4. STEP 2: CLUSTERING OF INDEPENDENT COMPONENTS

Let $\mathbf{x}$ denote the batch-matrix of $\mathbf{x}(n)$ for $n = 1, \ldots, N$. Once ICA is applied to $\mathbf{x}$ yielding the de-mixing transform $\mathbf{W}$, there are $m{\cdot}L$ arbitrarily ordered components in $\mathbf{c} = \mathbf{W}\mathbf{x}$. Without any loss in generality we normalize rows of $\mathbf{W}$ so that all the components have unit average variance.

The key assumption of our method is that each component is a filtered version of one of the d original sources, thus, the components can be grouped into d clusters so that each of them corresponds to one original source. Therefore, we propose a measure of similarity between the components which reflects how a filtered version of one component can be close to the other component and vice versa.

Let $c_j(n)$ denote the jth component, $j = 1, \ldots, m \cdot L$. The similarity measure between the ith and jth component is defined as

$$D_{ij} = \hat{\mathrm{E}}[\mathbf{P}_i c_j(n)]^2 + \hat{\mathrm{E}}[\mathbf{P}_j c_i(n)]^2, \tag{5}$$

where $\mathbf{P}_i$ denotes a projector on a subspace spanned by

$$c_i(n - L + 1), \ldots, c_i(n + L - 1), \tag{6}$$

and $\hat{\mathrm{E}}$ denotes the sample mean operator. The projection operator is given as

$$\mathbf{P}_i = \mathbf{I} - \mathbf{C}_i(\mathbf{C}_i^T\mathbf{C}_i)^{-1}\mathbf{C}_i^T \tag{7}$$

where $\mathbf{C}_i$ is composed of delayed versions of c_i as in (6). The matrix $\mathbf{C}_i^T\mathbf{C}_i$ is roughly equal to a multiple of autocovariance matrix of c_i of size $(2L - 1) \times (2L - 1)$. It can be computed using the FFT and inverted by the Levinson algorithm, both in a fast manner. In this way, for instance, computation of all the distances D_{ij} when $N = 6000$, $m = 2$ and $L = 20$ can be done in about 0.5 s.

The matrix of D_{ij}'s is then used as an input for a standard agglomerative hierarchical clustering algorithm with average linking strategy [1]. To be explicit, the clustering begins with $m \cdot L$ singletons. In each step, the number of clusters is reduced by one by fusing two clusters with the smallest mutual

[1] The implementation of the BGL algorithm is available online at [9].

[2] All our computations were done in Matlab (version 7.2) on a PC with 3GHz processor and 2GB RAM.

distance, which is computed as an average distance between individual components in the clusters. Hence, the number of clusters is sequentially reduced until it equals the assumed (or estimated) number of sources d.

5. STEP 3: RECONSTRUCTION

The reconstruction aims at transforming components of each cluster into responses (2). Unlike in [1], we admit here that any component (in any cluster) can contribute to reconstruction of any source,

$$\mathbf{x}^i = \mathbf{W}^{-1}\mathrm{diag}[\lambda_1^i, \ldots, \lambda_{mL}^i]\,\mathbf{c} \tag{8}$$

where λ_ℓ^i is a weight for reconstruction of the ith source using the ℓth component, $i = 1, \ldots, d$ and $\ell = 1, \ldots mL$. Finally, since the structure of $\mathbf{x}^i$ should be similar to that of $\mathbf{x}$ (see Figure 1), it suggests to retrieve the desired responses from $\mathbf{x}^i$ as

$$\widehat{s}_k^i(n) = \sum_{p=1}^{L} \mathbf{x}^i_{(k-1)L+p}(n + p - 1). \tag{9}$$

In [1], the weighting of components was very simple: λ_ℓ^i was set either to one or to zero according to whether the ℓth component belongs to the ith cluster or not. We call it a hard (*binary*) weighting.

The binary weighting has one drawback. It might happen that one or more components obtained in Step 1 is not clearly assigned to any cluster obtained in Step 2. Very often this happens in the case of low-frequency components, because it is difficult to separate them when the microphones are closely spaced. Nevertheless, these components are assigned to some of the clusters despite containing a significant portion of energy of other interfering sources.

In present work, we suggest a more advanced reconstruction by use of so-called *fuzzy* weighting. We allow all components to contribute to any reconstructed $\mathbf{x}^i$ by defining λ_ℓ^i so that its value reflects affiliation of the ℓth component to the ith cluster. Our ad-hoc definition of the weight λ_ℓ^i is

$$\lambda_\ell^i = \left(\frac{\sum_{j \in K_i, j \neq \ell} D_{\ell j}}{\sum_{j \notin K_i, j \neq \ell} D_{\ell j}} \right)^{\alpha}, \tag{10}$$

where K_i contains indices of components in the ith cluster, and α is an adjustable positive parameter that controls "hardness" of the weighting.

Note that the reconstruction is already computationally cheap compared to Steps 1-2.

6. EXPERIMENTAL RESULTS

We have repeated the experiment from [1] with the inclusion of our improved method presented here and the state-of-the-art algorithm of Sawada et al. [5]. Two audio sources (each pair described in the first column of Table 1) were played over two loudspeakers and recorded by two microphones in an environment shown in Figure 2. The recorded signals of length 18000 samples were processed by the competing methods yielding separated signals, whose quality was evaluated via two criteria: Signal-to-Interference Ratio (SIR) and Signal-to-Distortion ratio (SDR). Computations of the criteria was done by means of BSS_EVAL Toolbox from [6] using known original sources. Table 1 summarizes the results.

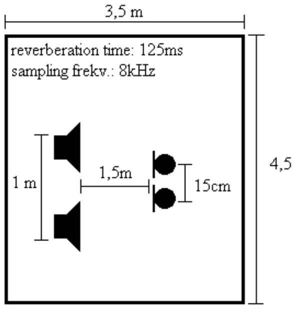

Fig. 2. Experimental conditions of our experiment.

As can be seen, performance of the presented method is comparable with that of other techniques in spite of using short separating filter and a segment of only 6000 samples of the whole data for the separation. The results were achieved when choosing the BGL algorithm for the ICA separator and taking $\alpha = 2$ in (10). Our further research will aim at objective comparison of other possible options of our method.

It should be noted that objective evaluation of separation quality is a difficult task in the cocktail party problem, and the evaluating criteria used here provide rather perfunctory comparison. For instance, the result achieved by the Sawada's algorithm with $L = 400$ is perceptually better, which is likely thanks to the longer separating filter that allows higher frequency resolution and does not depress badly separated frequencies as much as a short filter.

7. DISCUSSION

In this section, we devote several noteworthy comments to our time-domain approach presented in the paper.

✓ The utilization of ICA via seeking all independent components in $\mathbf{x}$ is very effective. This way, no constraint is imposed on the separating filters like in [8], thus, all MISO filters of length L having independent outputs are taken into account.

✓ The proposed method is able to operate efficiently on short data segments, whereas frequency-domain methods usually require longer data to generate sufficiently long short-time Fourier transforms of signals. This is a

algorithm	presented method		prototype from [1]		Parra, Spence [7]		Sawada et al. [5]		Sawada et al. [5]	
filter length L	20		20		128		20		400	
average comp. time (secs)	2.2		18.2		9.1		2.3		3.3	
	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR
man's voice #1	17.49	11.56	10.44	6.1	6.16	4.64	4.01	1.42	10.68	5.7
man's voice #2	15.65	11.81	5.63	2.48	5.44	1.38	11.08	7.32	13.16	6.75
man's voice	20.62	13.43	2.16	5.98	9.79	2.97	13.6	9.2	8.57	3.87
woman's voice	7.43	4.53	4.11	1.67	6.97	4.12	-2.02	-3.67	10.56	4.9
man's voice	18.79	10.27	14.19	6.71	8.45	4.76	15.08	6.68	18.8	5.75
Gaussian noise	17.68	13.61	9.43	5.87	11.34	8.65	13.24	9.96	17.22	11.69
man's voice	18.09	10.7	17.89	6.81	7.82	2.69	16.39	6.9	18.83	5.8
typewriter	23.5	17.31	12.22	8.88	11.97	9.50	14.83	11.47	19.21	13.71

Table 1. Results of separation of two sources from two microphones.

progressive feature towards development of fast adaptive real-time running algorithms.

✓ Performance of the method is invariant to initialization.

✓ The number of clusters could be arbitrary, thus, theoretically it could be higher than the number of microphones m.

✗ The dimension of $\mathbf{x}$ is $m \cdot L$, which makes the ICA computationally expensive when L is large.

Regarding the last negative feature of our method: There are two issues standing against each other in audio blind separation. While the length of separating filters is usually required to be large ($L > 100$) due to acoustic conditions, it is the question how long data do we need to effectively estimate all the filter coefficients? To reduce the number of parameters, one may consider long filters having most of their coefficients equal to zero.

8. CONCLUSIONS

The proposed algorithm was designed for batch processing. However, the computation times necessary for the separation are only about 2-3 times longer than the recorded signals. Since the separating mechanism can be kept frozen for certain time, we believe that the algorithm can be modified for on-line signal processing, which will be suitable for relatively fast changing environment, e.g. with moving acoustic sources. P-code of our method is available online at [10].

9. ACKNOWLEDGEMENT

Our thanks are due to Hiroshi Sawada for providing us results achieved by his algorithm in our experiment.

10. REFERENCES

[1] Z. Koldovský, P. Tichavský, “Time-domain blind audio source separation using advanced ICA methods, ” *Interspeech 2007: The 8th Annual Conference of the International Speech Communication Association*, Antwerp, Belgium, pp. 846-849, August 27-31, 2007.

[2] Z. Koldovský, P. Tichavský, and E. Oja, “Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound, ” *IEEE Tr. Neural Networks*, vol. 17, no. 5, pp. 1265- 1277, September 2006.

[3] D-T. Pham and J-F. Cardoso, “Blind separation of instantaneous mixtures of non stationary sources, ” *IEEE Trans. Signal Processing*, pp. 1837-1848, vol. 49, no. 9, 2001.

[4] P. Tichavský, A. Yeredor and J. Nielsen, “A fast approximate joint diagonalization algorithm using a criterion with a block diagonal weight matrix”, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Las Vegas, USA, March 2008.

[5] H. Sawada, S. Araki, and S. Makino, “MLSP 2007 data analysis competition: Frequency-domain blind source separation for convolutive mixtures of speech/audio signals, ” in *Proc. MLSP2007*, pp. 45 - 50, Aug. 2007.

[6] Vincent, E., Gribonval, R., and Févotte, C.: “Performance Measurement in Blind Audio Source Separation”, *IEEE Trans. on Speech and Audio Processing*, Vol 14, No 4, pp. 1462 - 1469, July 2006.

[7] Parra, L., and Spence, C.: “Convolutive Blind Separation of Non-Stationary Sources”, *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 3, pp. 320-327, May 2000.

[8] S. Douglas, M. Gupta, H. Sawada, and S. Makino, ”Spatio-temporal FastICA algorithms for the blind separation of convolutive mixtures, ” *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 5, pp. 1511-1520, July 2007.

[9] P. Tichavský, Fast Matlab^{TM} implementation of the BGL algorithm [online], `http://si.utia.cas.cz/downloadPT.htm`

[10] Z. Koldovský, Matlab^{TM} p-code of the proposed algorithm for blind audio source separation [online], `http://itakura.kes.tul.cz/zbynek/downloads.htm`