Kernel and Spectral Methods for Solving the Permutation Problem in Frequency Domain BSS

Yueyue Na and Jian Yu
Lab of Machine Learning and Cognitive Computation, Department of Computer Science
Beijing Jiaotong University
Beijing, China
{08112069, jianyu}@bjtu.edu.cn

***Abstract*—In frequency domain blind source separation (FDBSS), separated frequency bin data in the same source must be grouped together before outputting the final result, which is the well-known permutation problem. Clustering techniques are broadly used in solving the permutation problem, however, some challenges still exist, for example, elongated datasets should be handled, and constraint from the background knowledge must be considered. Inspired by various successful applications of kernel and spectral clustering methods in machine learning and data mining community, we try to solve the permutation problem by these methods. In this paper, the weighted kernel k-means algorithm is modified according to the specific requirement of the permutation problem, and the spectral interpretation of the kernel approach is also investigated. In addition, we propose several kernel construction approaches to improving the permutation performance. Different experiments are carried out on a uniform platform, and show better performance of the proposed approach.**

Keywords-blind source separation; kernel; spectral clustering; permutation problem

I. INTRODUCTION

The basic problem in blind source separation (BSS) is to recover original source signals from their mixed observations under the condition that the mixing system is unknown [1]. BSS techniques have a lot of potential applications, such as speech enhancement, robust speech recognition, hearing aids, underwater signal processing, image denoising, etc. [1, 2, 3]

Independent component analysis (ICA) [1, 4, 5] is one of the most widely used BSS methods. In the basic ICA model, mutually independent sources are linearly and instantaneously mixed together, non-Gaussianity is used as the measure of independence, which is maximized to separate sources from observed signals. Although instantaneous ICA has good performance on experimental datasets and has many successful applications, the instantaneous mixing model is still too ideal to solve many other real-world problems. In [6], several ICA algorithms are compared on artificial, simulated, and real datasets, and all algorithms fail to solve the real-word audio mixing problem.

In real-word acoustic environment, as a result of signal propagates in specific velocity, source signals are mixed with each other, as well as their delays, attenuations, and reverberations, i.e. they are mixed in a convolutive manner. In such case, the mixing environment is modeled by the finite duration impulse response (FIR) filters from each source to each sensor. The convolutive mixing model is far more complicated than the instantaneous mixing model because there is an unknown demixing filter bank needing to be estimated, rather than a demixing matrix like in instantaneous ICA.

In convolutive mixture, FIR filter length can vary from taps to thousands of taps according to different applications. For speech and audio separation, very long FIR filters are needed since both sound propagation velocity and sampling rate are high. Although there are time domain approaches for the convolutive mixing problem, such long filters make the algorithm time consuming and difficult to converge. Thus, the frequency domain blind source separation (FDBSS) [7, 8] is often used in speech and audio separation. FDBSS algorithms usually have three steps: First, the short-time Fourier transform (STFT) converts input signals into time-frequency domain, where time domain convolutive mixture becomes frequency domain instantaneous mixture in each frequency bin. Then, complex-valued instantaneous ICA techniques are applied on individual frequency bin independently to get the separated bin-wise data. Finally, the inverse short-time Fourier transform (ISTFT) outputs the time domain separated signals. Solving the separation problem in frequency domain has many advantages: First, the whole problem is divided into several instantaneous mixing sub problems in each frequency bin, moreover, signal's non-Gaussianity in frequency domain is stronger than in time domain [9], which makes the problem easier; Second, well studied complex-valued ICA techniques [4, 5] can be used directly, and different ICA algorithms can be selected according to different data and applications; Third, since ICA is performed independently in each frequency bin, parallel techniques can be used to speed up the process.

However, FDBSS algorithms suffer from permutation and scaling ambiguities, because ICA in different frequency bins can output data in different orders and scales. The scaling ambiguity is relatively easy to solve [7], while the permutation problem is more difficult to handle, and will lead to more severe problem. If the permutation problem is not well solved, even if independent sources are separated correctly in every frequency bin, sources are still mixed across frequencies in time-frequency domain, this can cause the entire separation procedure fail.

This research was supported by "the Fundamental Research Funds for the Central Universities", "National Natural Science Foundation of China, no. 61033013".

Many algorithms have been proposed to solve the permutation problem. Since speech and music signal's spectrogram is sparse and highly correlated in neighboring frequency bins, the dependence of estimated sources across frequencies is commonly exploited to group spectrogram data from the same source together. The inter-frequency correlation of signal envelop is used in [7, 10] to align the permutation. In [11], the signal power ratio of neighboring frequency bins are grouped by a k-means-like algorithm according to their correlations. Since there is no centroid-based structure in the full frequency band, local optimization is performed after the clustering process. In [12], sparsity of the estimated signals is used as a kind of similarity measure, with a hierarchical agglomerative approach called dyadic sorting [13] to group consecutive subbands together. A region-growing approach is presented in [14], power ratio is also used in this algorithm, and neighboring frequency bins with high correlations are merged in advance to get robust output. In [15], Haar wavelet transform is used to extract features from spectrogram data, then SVD is used for dimensionality reduction purpose, at last hybrid k-means clustering is performed to solve the permutation problem.

Exploiting the directive pattern of sources is another commonly used strategy to solve the permutation problem. This class of methods is based on the far-field model and the near-field model [16], as well as the direct propagation path assumption of signals, and it is believed that contributions from the same source are likely to come from the same direction. By extracting the directive patterns from the estimated demixing or mixing matrices, direction-of-arrival (DOA) of sources or even source locations can be estimated, then the permutation problem can be aligned [16]. In order to calculate DOA, prior knowledge such as sensor location or sensor spacing, and signal propagation velocity must be known in advance, this makes the algorithm not fully blind. Early DOA based methods often suffer from the spatial aliasing problem, however, phase difference of the estimated mixing matrices is used as the clustering feature in [17, 18], influence of the spatial aliasing is avoided and no prior knowledge is required in this approach since DOA pattern is implicitly hidden in the phase information. In [10], a hybrid approach is proposed in order to gain robust and precise performance, both neighboring bin correlations and sources directive patterns are used in this approach.

Recently, the Independent Vector Analysis (IVA) approach [19, 20] has been proposed to avoid the permutation problem. In IVA, the optimization process uses vectors, but not scalars like in ICA. The main idea of this approach is to estimate the demixing matrices in all frequency bins together so that the permutation ambiguity is avoided. However, as pointed in [15], the main difficulty of IVA still lies in the estimation of long demixing frequency responses precisely. In addition, the flexibility of FDBSS algorithm is lost since it is not convenient to change ICA algorithms according to applications.

Clustering techniques are widely used in solving the permutation problem in frequency domain blind source separation. However, it is not efficient to use traditional partition based clustering algorithms, like k-means, directly since there are no centroid-based structures in the input space. In the last decade, kernel and spectral clustering methods have been proposed to produce nonlinear partitions among clusters [21, 22]. Inspired by the successful applications of kernel and spectral clustering methods such as image segmentation [23], paleontological data mining [24], social network analysis, etc. we try to solve the permutation problem by kernel and spectral approaches. Our work in this paper mainly contains three parts: 1), we modify the weighted kernel k-means algorithm [21] to make it suitable for the permutation problem; 2), we propose additional kernel construction methods to improve the performance; 3), a platform is developed for BSS research and application purpose.

The rest of this paper is organized as follows: In section 2 we briefly introduce the concept of frequency domain BSS. In section 3 we present the weighted kernel k-means algorithm, its connection to spectral clustering is investigated in section 4. In order to improve the performance, in section 5 we give the kernel construction policy and corresponding improvements. In section 6 we conduct the experiments and show the comparison results. At last, we conclude this paper in section 7.

II. FDBSS AND PERMUTATION PROBLEM

In many real-world applications such as the "cocktail party problem", signals are often mixed in convolutive manner. Supposing there are N sources and M sensors ($M \geq N$), with the source vector $\boldsymbol{s}(t) = [s_0(t), \cdots, s_{N-1}(t)]^T$ and the observed sensor vector $\boldsymbol{x}(t) = [x_0(t), \cdots, x_{M-1}(t)]^T$, then the mixing process can be formulated in (1) [12, 14].

$$\boldsymbol{x}(t) = \boldsymbol{H}(t) * \boldsymbol{s}(t) = \sum_{l=0}^{L-1} \boldsymbol{H}(l)\boldsymbol{s}(t-l) \tag{1}$$

In (1), $\boldsymbol{H}(t)$ is a sequence of $M \times N$ matrices which are used to model the mixing environment with FIR filters of length L. The separating process can be regarded as an "inverse procedure" of the mixing process:

$$\boldsymbol{y}(t) = \boldsymbol{W}(t) * \boldsymbol{x}(t) = \sum_{l=0}^{L-1} \boldsymbol{W}(l)\boldsymbol{x}(t-l) \tag{2}$$

where $\boldsymbol{y}(t) = [y_0(t), \cdots, y_{N-1}(t)]^T$ is the estimated source signal vector, $\boldsymbol{W}(t)$ is a sequence of $N \times M$ matrices which represent the demixing filters [12, 14].

In frequency domain BSS, the L-point short-time Fourier transform (STFT) is applied so that the convolutive mixing and demixing process in time domain are converted to the instantaneous mixing and demixing in frequency domain:

$$\begin{aligned} \boldsymbol{X}(f,\tau) &= \boldsymbol{H}(f)\boldsymbol{S}(f,\tau) \\ \boldsymbol{Y}(f,\tau) &= \boldsymbol{W}(f)\boldsymbol{X}(f,\tau) \end{aligned} \tag{3}$$

where $f = 0, \cdots, L/2$ is the frequency bin index, only half of the frequency bins are needed because of the complex conjugate property of the FFT; τ is the STFT frame index, $\boldsymbol{X}(f,\tau)$, $\boldsymbol{Y}(f,\tau)$, $\boldsymbol{H}(f)$, $\boldsymbol{W}(f)$, $\boldsymbol{S}(f,\tau)$ are the frequency domain versions of $\boldsymbol{x}(t)$, $\boldsymbol{y}(t)$, $\boldsymbol{H}(t)$, $\boldsymbol{W}(t)$, $\boldsymbol{s}(t)$ [12, 14].

Since the observed signals in each frequency bin are instantaneously mixed, complex-valued ICA algorithms, like [4, 5], can be used to separate independent sources from their mixtures. However, the permutation and the scaling ambiguities from instantaneous ICA are introduced:

$$\boldsymbol{Y}(f,\tau) = \boldsymbol{W}(f)\boldsymbol{X}(f,\tau) = \boldsymbol{\Lambda}(f)\boldsymbol{\Pi}(f)\boldsymbol{S}(f,\tau) \tag{4}$$

In (4), $\boldsymbol{\Pi}(f)$ is a permutation matrix, $\boldsymbol{\Lambda}(f)$ is a diagonal scaling matrix [14]. The permutation and the scaling ambiguities occur at every frequency bin, if the permutation problem is not well handled, data from different frequency bins are still mixed in time-frequency domain, and this will cause the entire separation procedure fail. After the permutation problem is solved, the scaling ambiguity can be tackled by (5) [9, 14]:

$$\boldsymbol{W}_{ps}(f) = diag\{[\boldsymbol{\Pi}(f)\boldsymbol{W}(f)]^{+}\}\boldsymbol{\Pi}(f)\boldsymbol{W}(f) \tag{5}$$

where $\boldsymbol{W}_{ps}(f)$ is the aligned and rescaled demixing matrix for frequency bin *f*, $diag\{\cdot\}$ retains only the diagonal entries of a matrix, $[\cdot]^{+}$ computes the pseudo inversion of a matrix. Finally, $\boldsymbol{W}_{ps}$ is used to calculate the separated signals in each frequency bin, and the inverse short-time Fourier transform (ISTFT) is used to get the estimated time domain sources.

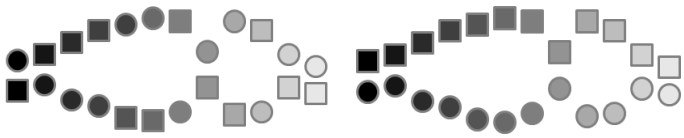
III. Solving the Permutation Problem

A. Problem Formulation

The permutation problem can be considered as a constrained clustering problem: let $V = \{\boldsymbol{v}_{n,f}\}$ be samples needing to be aligned, each $\boldsymbol{v}_{n,f}$ represents the feature vector corresponding to the output channel $n \in \{0, \cdots, N-1\}$ and the frequency bin $f \in \{0, \cdots, L/2\}$. The goal of the clustering is to group each $\boldsymbol{v}_{n,f}$ into *N* disjoint clusters $V_p \neq \emptyset$, $V_p \cap V_q = \emptyset$, $\bigcup_{p=0}^{N-1} V_p = V$, $p, q \in \{0, \cdots, N-1\}$ according to some similarity metric, so that samples in the same cluster are more similar to each other than to those in other clusters. After the clustering process, samples in the same cluster should correspond to frequency bins in the same channel. For the permutation problem, an additional constraint in (6) must be considered, it comes from the background knowledge that samples in the same frequency bin cannot belong to the same channel.

$$if\ \boldsymbol{v}_{n,f} \in V_p, n \neq n'\ then\ \boldsymbol{v}_{n',f} \notin V_p \tag{6}$$

When bin-wise features like signal envelop [10] and power ratio [11], or directive patterns like phase information [18] are used in the clustering, samples in the same cluster are probably distributed in a one dimensional nonlinear manifold [25] in high dimensional input data space. An illustrative view of data distribution is shown in Fig. 1.



(a) before permutation (b) after permutation

Figure 1. Data distribution illustration.

There are two channels in Fig. 1, different shapes represent samples in different channels, and different gray levels represent the variation from low frequency to high frequency. Since samples only have high similarities with their neighbors in the same channel, two samples may be quite different when they are far apart in the frequency band, even if they belong to the same channel. Usually, input signals are band-pass signals, there is little energy in very low and very high frequency bands, besides, the sparsity of speech and audio signal is not strong in low frequency part, these properties make samples in these frequency bands difficult to be separated. When the number of channels is large or environment reverberation is high, or source and sensor locations are nearly singular placed [26], ICA sometimes will output poor results in some frequency bins, and this will also degrade the permutation algorithm, which uses ICA's output as its input.

B. The Weighted Kernel k-means Algorithm

From the observations in the previous subsection, we can infer that elongated clusters need to be handled. The use of kernels allows mapping data from input space into an implicit high dimensional space called feature space, by using a proper nonlinear mapping, one can extract clusters that are nonlinearly separable in input space [21, 22]. So, it is a natural choice to solve the permutation problem by kernel methods.

The weighted kernel k-means algorithm is introduced by Dhillon et al. in [21], it is an extension of traditional k-means clustering by the use of kernel function. Let $\omega(\boldsymbol{v})$ denote the weight for sample $\boldsymbol{v}$, when the nonlinear mapping ϕ is used, the objective function of the algorithm can be formulated as:

$$\mathcal{D}\left(\{V_p\}_{p=0}^{N-1}\right) = \sum_{p=0}^{N-1} \sum_{\boldsymbol{v} \in V_p} \omega(\boldsymbol{v}) \left\| \phi(\boldsymbol{v}) - \boldsymbol{m}_p \right\|^2 \tag{7}$$

where $\boldsymbol{m}_p = \sum_{\boldsymbol{u} \in V_p} \omega(\boldsymbol{u})\phi(\boldsymbol{u}) / \sum_{\boldsymbol{u} \in V_p} \omega(\boldsymbol{u})$ is the best cluster center for $\phi(\boldsymbol{v})$ in the mapped feature space. The squared Euclidean distance between $\phi(\boldsymbol{v})$ and $\boldsymbol{m}_p$ can be expanded as:

$$\left\| \phi(\boldsymbol{v}) - \boldsymbol{m}_p \right\|^2 = t_1 - t_2 + t_3 \tag{8}$$

$$t_1 = \phi(\boldsymbol{v}) \cdot \phi(\boldsymbol{v}) \tag{9}$$

$$t_2 = \frac{2 \sum_{\boldsymbol{u} \in V_p} \omega(\boldsymbol{u}) \phi(\boldsymbol{v}) \cdot \phi(\boldsymbol{u})}{\sum_{\boldsymbol{u} \in V_p} \omega(\boldsymbol{u})} \tag{10}$$

$$t_3 = \frac{\sum_{\boldsymbol{u}_1, \boldsymbol{u}_2 \in V_p} \omega(\boldsymbol{u}_1)\omega(\boldsymbol{u}_2) \phi(\boldsymbol{u}_1) \cdot \phi(\boldsymbol{u}_2)}{(\sum_{\boldsymbol{u} \in V_p} \omega(\boldsymbol{u}))^2} \tag{11}$$

From (8) ~ (11) we can see that the squared Euclidean distance in (7) can be calculated in the form of inner product operations. Since all inner product values $\phi(\boldsymbol{u}) \cdot \phi(\boldsymbol{v})$ are provided by entries of the kernel matrix $\boldsymbol{K}$, the objective function can be computed without knowing the actual $\phi(\boldsymbol{v})$ and $\boldsymbol{m}_p$ in feature space.

The weighted kernel k-means algorithm is given in Algorithm 1. For the permutation problem, the number of clusters equals to the number of output channels, and the algorithm initialization is directly given by the output in ICA step. In step 4 of the algorithm, to hold the constraint in (6), our policy here is to try all possible permutations $\Pi: (0, \cdots, N-1) \rightarrow (\pi_0, \cdots \pi_{N-1})$ and select the one with minimum total cost. Although *N*! tries are needed, usually the number of sources is not large, so this method is efficient. The error rate of the weighted kernel k-means algorithm will reduce if good initialization is provided. In [21], the authors suggest a two-layer approach:

first generate the initialization by spectral clustering, and then refine the partition by kernel k-means. In our case, we find that initializing clusters according to ICA's output is efficient enough to solve the permutation problem. To further improve the performance, one can use other methods, policy in [27] for example, to have the output in ICA step nearly aligned, or use other permutation algorithm's output as initialization, then followed the weighted kernel k-means approach.

Input: $\boldsymbol{K}$: kernel matrix

Output: $V_0, \cdots, V_{N-1}$: aligned clusters

1. Initialize clusters $V_0^{(0)}, \cdots, V_{N-1}^{(0)}$ according to the initial permutation outputted by ICA.
2. Calculate the weight for each sample: $w_i = \sum_j k_{ij}$.
3. For all samples $\boldsymbol{v}_{n,f}$ in frequency bin f, calculate the cost of assigning each sample to each cluster according to (8) ~ (11).
4. Select the best assignment with minimum cost, meanwhile the constraint in (6) is kept.
5. Update clusters according to the results of step 4 in the entire frequency band. If new cluster assignment is the same as previous one, return the result, else go to step 3.

Algorithm 1. The weighted kernel k-means algorithm.

IV. Spectral Interpretations

Spectral method [28, 29] is another category of promising clustering methods in machine learning and data mining community. This class of algorithms is based on the spectral graph theory, and the clustering problem can be seen as a graph partition problem, which can be approximately solved by the eigenvalue decomposition of the graph Laplacian. In [21], it is proven that weighted kernel k-means and spectral clustering have the *same* objective function. Since the two kinds of methods are highly related, in this section we give spectral interpretations to the permutation problem.

A. Spectral Clustering Interpretation

The permutation problem can also be considered as a graph partition problem (see Fig. 2). The set of samples is represented as a connected weighted undirected grid $G = (V, E)$, where $V = \{\boldsymbol{v}_{n,f}\}$ are nodes of the grid, and edges are formed between every pair of nodes, the weight on each edge represents the similarity between the node pair. In the example of Fig. 2, the grid will be partitioned into two disjoint clusters V_0, V_1, $V_0 \cup V_1 = \mathrm{V}, V_0 \cap V_1 = \emptyset$ according to the dash line, the partition can be done by simply removing edges connecting the two parts. In order to get the optimal partition, samples in Fig. 2 need to be rearranged column by column so that the intra-cluster similarities are maximized meanwhile the inter-cluster similarities are minimized. It was proven in [23] that the goal can be achieved by minimizing the normalized cut criteria formulated in (12):

$$Ncut(V_0, V_1) = \frac{\sum_{\boldsymbol{u}\in V_0, \boldsymbol{v}\in V_1} sim(\boldsymbol{u},\boldsymbol{v})}{\sum_{\boldsymbol{u}\in V_0, \boldsymbol{v}\in V} sim(\boldsymbol{u},\boldsymbol{v})} + \frac{\sum_{\boldsymbol{u}\in V_0, \boldsymbol{v}\in V_1} sim(\boldsymbol{u},\boldsymbol{v})}{\sum_{\boldsymbol{u}\in V_1, \boldsymbol{v}\in V} sim(\boldsymbol{u},\boldsymbol{v})} \quad (12)$$

where $sim(\boldsymbol{u}, \boldsymbol{v}) \in [0, 1]$ represents the similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$. An approximate solution to this graph partition problem can be found efficiently by spectral clustering algorithms [23, 29].

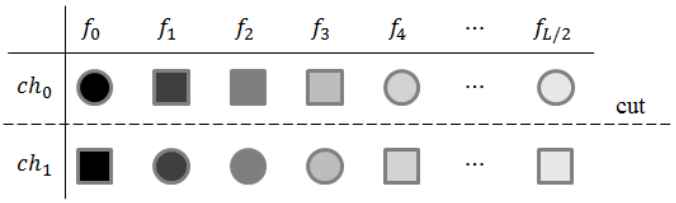


Figure 2. Spectral clustering interpretation.

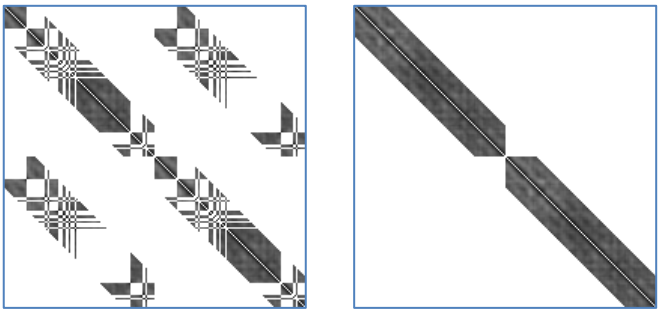
B. Spectral Ordering Interpretation

Spectral ordering is introduced by Ding et al. in [30], Mavroeidis et al. further use this method for paleontological data mining [24]. The purpose of ordering is to optimize a linear order for all samples and ensure that adjacent samples are similar, while dissimilar samples are far apart. Let $k_{i,j}$ denote the similarity between the ith sample and the jth sample, and $\Pi: (0, \cdots, N \times (L/2+1) - 1) \to (\pi_0, \cdots, \pi_{N\times(L/2+1)-1})$ represent a permutation, the objective function of spectral ordering can be formulated as:

$$J(\Pi) = \sum_{i,j} (i-j)^2 k_{\pi_i, \pi_j} \quad (13)$$

In [30], it is proven that optimize (13) can also be approximately done by the eigenvalue decomposition of graph Laplacian, which has the same mathematical expression as spectral clustering.

For the permutation problem, we can assume that a linear order exists on all samples in the same channel. When misalign occurs, the underlying linear order is disarranged (see rows of Fig. 2), so we can optimize (13) and use the ordering result to produce the correct permutation. An example of solving the permutation problem in two channels case by spectral ordering is given in Fig. 3. Visualizations of affinity matrices before and after spectral ordering are shown in this figure, darker pixels means higher similarities.



(a) before ordering (b) after ordering

Figure 3. Solving the permutation problem by spectral ordering.

Weighted kernel k-means and spectral clustering can be regarded as two different optimization strategies for the same objective function. However, for the permutation problem, the weighted kernel k-means approach is preferred in that the con-

straint in (6) is easy to be handled in the optimization procedure, in addition, eigenvalue decomposition for large matrices can be avoid.

V. KERNEL CONSTRUCTION

A. Neighborhood Based Approach

Kernel construction is very important for the weighted kernel k-means algorithm in Algorithm 1, clustering result is directly affected by the kernel quality. Gaussian kernel in (14) is a commonly used kernel function in many applications, the parameter σ controls how fast the values decay to zero.

$$K(\boldsymbol{u},\boldsymbol{v}) = exp(-\|\boldsymbol{u}-\boldsymbol{v}\|^2/2\sigma^2) \tag{14}$$

Although the Gaussian kernel can also be used in the permutation problem, the parameter σ is difficult to tune. Considering the special properties of the permutation problem: samples are distributed in a 2D grid, and only have large similarities in an interval, we use (15) to construct the kernel matrix instead of the Gaussian kernel:

$$K^{(1)}(\boldsymbol{v}_{n,f},\boldsymbol{v}_{n',f'}) = \begin{cases} sim(\boldsymbol{v}_{n,f},\boldsymbol{v}_{n',f'}) & |f-f'| \le \delta \\ 0 & f \ne f' \\ & otherwise \end{cases} \tag{15}$$

In (15), the parameter δ is introduced to control the frequency neighborhood, and 2D channel-frequency bin index can be converted to 1D matrix index as: $i = n \times (L/2+1) + f$. Different similarity measures can be used in the kernel construction, for example: power ratio correlation introduced in [11], or cosine of phase difference in [18], etc. When power ratio correlation is used as the similarity metric, since correlation coefficient ranges from -1 to 1, we simply set the similarity to zero when two feature vectors are negatively correlated.

Kernel matrix and affinity matrix, which is used in spectral clustering, are very similar since entries of both matrices represent similarities of samples, however, the two matrices have different physical meaning: kernel matrix represents the inner product of samples in feature space, while affinity matrix stands for a weighted undirected graph. According to their physical interpretations, diagonal entries of a kernel matrix should be set to nonzero values in order to ensure the positive semi-definiteness, while diagonal entries of an affinity matrix should be set to 0 because no self-connections are allowed in a simple graph.

B. Single Linkage Improvement

When kernel matrix is constructed by (15), a sample is connected to all other samples in its neighboring frequency bins. However, from (6) we can infer that a sample can only connect to one sample in one frequency bin. According to this constraint, we only keep the connection from one sample to its nearest neighbor in one frequency bin, and this yields the single linkage approach. In (16), $k_{ij}^{(2)}$ and $k_{ij}^{(1)}$ are entries of the kernel matrix $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(1)}$, $j'\ mod\ (L/2+1)$ computes the frequency bin index for the j'th element in kernel matrix. After the single linkage process in (16), we still need to set $k_{ji}^{(2)} = k_{ij}^{(2)}$ to keep the kernel matrix symmetric. Fig. 4 is an illustration of the single linkage approach. In this figure, solid lines and dash lines are connections in (15), however, only the solid connections are kept in (16).

$$k_{ij}^{(2)} = \begin{cases} k_{ij}^{(1)} & k_{ij}^{(1)} = \max_{j'\ mod\ (L/2+1)=f} k_{ij'}^{(1)} \\ 0 & otherwise \end{cases} \tag{16}$$

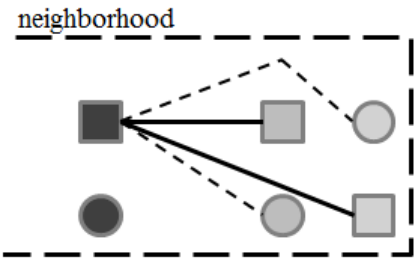
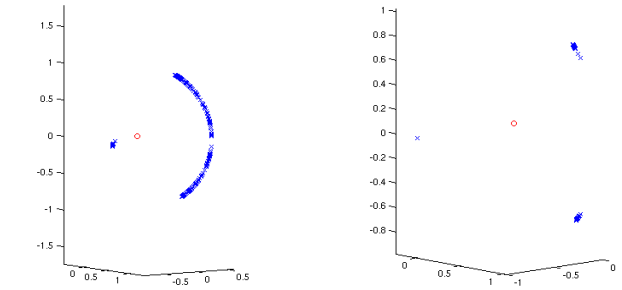


Figure 4. Single linkage connection.

Effect of the single linkage approach can be considered as decreasing the inter-cluster similarities, an example of three channels case is shown in Fig. 5. In this figure, the mapped samples in spectral embedding space by NJW algorithm [29] are shown, where the red circle is the origin. When the affinity matrix which is used in NJW is constructed by the single linkage scheme, it becomes easier to cut the connected graph represented by the affinity matrix into disconnected parts. Samples in the same cluster are nearly collapsed to one point in spectral embedding space in Fig. 5, which makes the cluster structures easy to detect.



(a) neighborhood based approach (b) single linkage approach

Figure 5. Effect of single linkage.

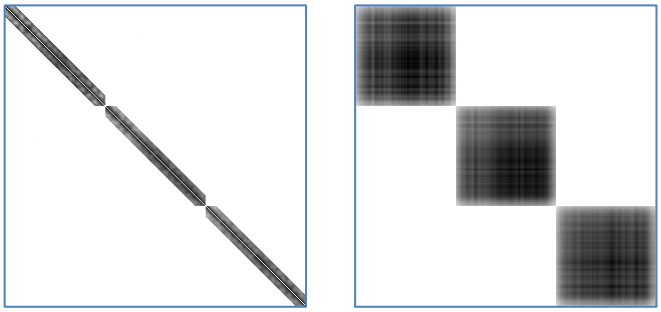
C. Connectivity Matrix Improvement

Given an affinity matrix $\boldsymbol{K}^{(2)}$, the decomposition in (17) is used in [30], where $\boldsymbol{D}$ is a diagonal matrix with $d_{ii} = \sum_j k_{ij}^{(2)}$, λ_n and $\boldsymbol{z}_n$ are top-N eigenvalues and corresponding eigenvectors of $\boldsymbol{L}$. Based on this decomposition, the connectivity matrix $\boldsymbol{K}^{(3)}$ is constructed by (18), followed the noise reduction procedure in (19). It is shown in [30] that connectivity matrix has a so-called self-aggregation property that connectivities between different clusters are suppressed while connectivities within clusters are enhanced. Fig. 6 shows the difference between affinity matrix and connectivity matrix, permutation problem is solved in this example for visualization purpose. We can see that after the approximation in (17) ~ (19), elongated cluster structures are converted to compact centroid-based cluster structures, which can be easily handled by k-means.

$$\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{K}^{(2)}\boldsymbol{D}^{-1/2}$$
$$\boldsymbol{K}^{(2)} = \boldsymbol{D}^{1/2}\boldsymbol{L}\boldsymbol{D}^{1/2} \approx \boldsymbol{D}^{1/2}(\sum_{n=0}^{N-1}\boldsymbol{z}_n\lambda_n\boldsymbol{z}_n^T)\boldsymbol{D}^{1/2} \quad (17)$$

$$\boldsymbol{K}^{(3)} = \boldsymbol{D}^{1/2}(\sum_{n=0}^{N-1}\boldsymbol{z}_n\boldsymbol{z}_n^T)\boldsymbol{D}^{1/2} \quad (18)$$

$$k_{ij}^{(3)} = \begin{cases} k_{ij}^{(3)} & k_{ij}^{(3)}/((k_{ii}^{(3)})^{1/2}(k_{jj}^{(3)})^{1/2}) \geq 0.8 \\ 0 & otherwise \end{cases} \quad (19)$$



(a) affinity matrix (b) connectivity matrix

Figure 6. Connectivity matrix visualization.

VI. EXPERIMENTS

A. Platform Design and Implementation

All our experiments are carried out on a uniform platform developed in Java (source code is available for public). In this system, long and multiple signal records are supported, and intermediate results can be saved for further use, moreover, new ICA and permutation algorithms, or other new features, can easily be integrated in. The FastICA [4] and the RobustICA [5] algorithms are implemented in the system. We believe that this platform can be used for research purpose, or used as a prototype for real-world BSS applications. Fig. 7 is a flowchart of the platform.

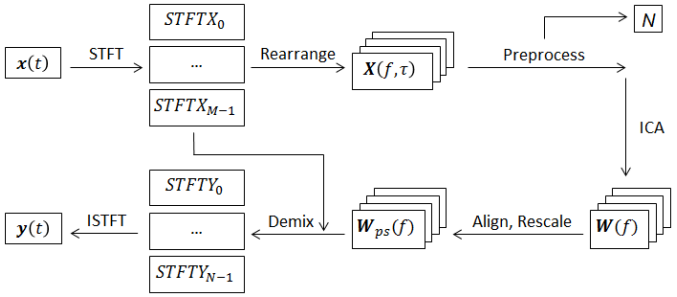


Figure 7. The frequency domain BSS platform.

Seven permutation algorithms are compared in the following experiments, including: sequential align by signal envelop correlation (envelop) used in [10], dyadic sorting by power ratio correlation (dyadic) [11, 13], the sparsity based algorithm (sparsity) [12], the region-growing algorithm (region) [14], the NJW approach (njw) [29], the spectral ordering approach (so) [30], and the weighted kernel k-means algorithm (wkkmeans) [21]. The complex-valued FastICA algorithm [4] is used for instantaneous separation, and all permutation algorithms are compared upon the same ICA separation results in a single experiment. Separation performance is evaluated in terms of signal-to-interference ratio (SIR) improvement [7, 14], please notice that the SIR values in our experiments are different from the results reported in [10, 14] because we calculate SIR in frequency domain according to the method in [31].

B. Separation of Two Sources

The first experiment evaluates the separation performance of two sources, dataset from [32] is used, signal sampling rate is 8000 Hz. FIR filters of different length for the mixing system are randomly generated by concatenating different all-pass filters. In this experiment, STFT block size is set to 1024, with 7/8 overlap, FFT size is set to 2048.

The performance is shown in Fig. 8, this figure tells us that the spectral clustering approach (njw, so) and the kernel approach (wkkmeans) have very similar performance in two channels case, both approaches have comparable performance with state-of-the-art permutation algorithms.

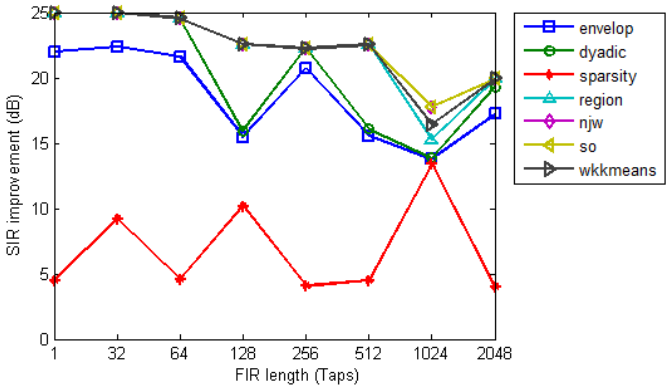


Figure 8. Performance of two sources separation.

C. Separation in High Reverberation Environment

In the second experiment, separation performance of multiple sources in simulated high reverberation environment is evaluated. Different FIR filters of 2048 taps are randomly generated for signal mixture, an example filter is shown in Fig. 9.

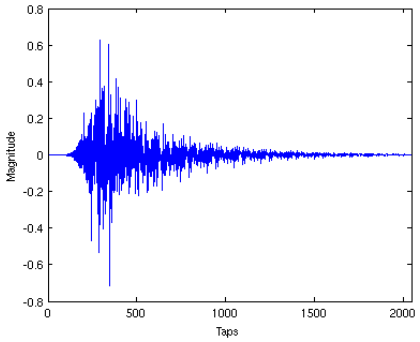


Figure 9. FIR filter example.

Two datasets are used in this experiment. The first one is from [32], with 8000 Hz sampling rate, the second one is recorded by us, with 20 seconds wave files, 22050 Hz sampling rate. For the first dataset, STFT block size is set to 1024, with 7/8 overlap, FFT block size is set to 2048; for the second dataset, STFT block size is set to 2500, with 3/4 overlap, FFT block size is set to 4096.

The separation performance is given in Table 1, we can see that the average performance on the second dataset is higher than the first one, the reason behind this phenomena is that there are plenty of data available in the second dataset, which makes the independent assumption strong enough [33]. The weighted kernel k-means approach has slightly performance improvement on both datasets compared with state-of-the-art permutation algorithms, however, since the constraint in (6) cannot be fulfilled in the spectral clustering methods, the NJW and the spectral ordering approaches usually fail in more than 2 channels cases.

TABLE I. PERFORMANCE IN HIGH REVERBERATION ENVIRONMENT

	SIR (Dataset 1 [32])			SIR (Dataset 2)		
	N=2	*N=3*	*N=4*	*N=2*	*N=3*	*N=4*
envelop	17.22	3.70	7.94	18.50	14.34	15.83
dyadic	19.30	3.26	11.93	29.40	16.17	18.34
region	19.92	14.73	15.80	29.40	21.30	23.43
njw	19.92	--	--	29.40	--	--
so	19.92	--	--	29.40	--	--
wkkmeans	19.92	15.52	18.75	29.40	21.35	26.56

D. *Separation of Nine Sources*

In order to test the scalability and stability of our platform and the proposed permutation algorithm, we try to perform the separation of nine sources which are downloaded from the cocktail party problem demo web page [34], sampling rate is 8000 Hz. For simplicity, random mixing filters of length 1 are used, i.e. instantaneous mixing is performed, however, the permutation problem remains in frequency domain BSS approach. In this experiment, STFT block size is set to 512, with 7/8 overlap, FFT block size is set to 1024.

TABLE II. SEPARATION PERFORMANCE OF NINE SOURCES MIXTURE

	envelop	region	wkkmeans	region+ wkkmeans
Input SIR	-4.87, -8.05, -13.42, -9.66, -8.78, -18.17, -16.95, -9.40, -5.43			
Output SIR	11.46	13.59	12.87	13.59
	11.08	15.19	10.81	15.83
	6.41	17.16	5.03	17.41
	8.39	19.43	13.90	20.85
	8.59	15.17	15.81	16.18
	4.45	15.99	-3.35	16.16
	2.58	5.59	-2.87	5.59
	11.25	16.68	9.62	15.47
	8.68	11.34	11.21	11.34
SIR	18.62	24.98	18.64	25.24

Separation performance is given in Table 2. In this experiment we can see that the region growing approach [14] is very robust, even ICA was failed to converge in a few frequency bins in the separation procedure. Use the weighted kernel k-means algorithm directly is not as good as the region growing approach. However, when we initialize the weighted kernel k-means by the result of region growing, the separation performance is slightly improved. Unfortunately, output SIR of the 8th source is decreased when the region growing is followed by the weighted kernel k-means. This means that the weighted kernel k-means algorithm still cannot keep the performance monotonously increasing when good initialization is used.

E. *Align by Directive Pattern*

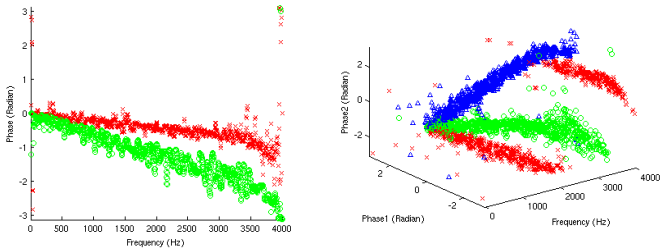
Other features can also be used in the weighted kernel k-means framework. In the last experiment, we solve the permutation problem by the source directive pattern in (20), which is used in [18]. In this experiment, only feature and similarity calculation policy are changed, while the algorithm remains as in Algorithm 1.

$$\theta_{m,n}(f) = \angle(h_{m,n}(f)/h_{m^*,n}(f)), m \neq m^* \qquad (20)$$

In (20), $\angle(\cdot) \in (-\pi, \pi]$ is the phase of a complex number, $h_{m,n}(f)$ represents entries of the estimated mixing matrix for frequency bin f, and m^* is the reference sensor index. A M-1 dimensional feature vector $\boldsymbol{\theta}_{n,f}$ for output channel n, frequency bin f can be constructed according to (20). Similarity between two feature vectors can be calculated by (21), where $[a]_+ = 0$ when $a < 0$, and $[a]_+ = a$ when $a \geq 0$. The spatial aliasing problem will not affect the similarity calculation thanks to that the cosine operation still can give correct result when spatial aliasing occurs.

$$sim(\boldsymbol{\theta}_{n,f}, \boldsymbol{\theta}_{n',f'}) = min_m[\cos(\theta_{m,n}(f) - \theta_{m,n'}(f'))]_+ \qquad (21)$$

Dataset in [32] is also used in this experiment, however, the mixed signals provided by the dataset is used directly since the directive pattern is clear in these mixtures. The aligned directive patterns by weighted kernel k-means are shown in Fig. 10, and the separation performance is given in Table 3.



(a) 2 sources (N=2) (b) 3 sources (N=3)

Figure 10. Aligned directive patterns.

TABLE III. PERFORMANCE OF DIRECTIVE FEATURE

	2 sources (N=2)	3 sources (N=3)
power ratio	15.77	11.69
phase	13.99	8.17

Since the mixing filters are unknown for the SIR evaluation method in [31], SIR in this experiment is calculated by the bss_eval toolbox [35]. Separation performance by power ratio plus weighted kernel k-means is also given for comparison. Table 3 tells us that the permutation performance by phase difference is not as good as the performance by power ratio, this can be explained by Fig. 10 that there are strong phase overlapping in low frequency part.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a new approach for the permutation problem in frequency domain blind source separation. In

order to group data in the same source together, elongated cluster structures should be considered, and the weighted kernel k-means algorithm is used to do the work. We modify this algorithm to make it suitable for the permutation problem, meanwhile, spectral interpretations of the kernel method is also investigated. We also present several kernel construction methods to improve the performance. Experiments are conducted on a uniform Java platform, and experimental results show that the proposed approach improves the separation performance.

In future work, we would like to incorporate the manifold clustering method in our platform to discover the one dimensional manifold structures for the permutation problem. Since the constraint in (6) can also be considered as a kind of "cannot-link" supervision information, we also want to use semi-supervised learning methods to further improve the performance.

REFERENCES

[1] A. Hyvarinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411-430, 2000.

[2] Z. Xue, J. Li, S. Li, B. Wan, "Using ICA to Remove Eye Blink and Power Line Artifacts in EEG," *International Conference on Innovative Computing, Information and Control*, vol. 3, pp. 107-110, 2006.

[3] A. Mansour, N. Benchekroun, C. Gervaise, "Blind Separation of Underwater Acoustic Signals," *ICA 2006*, pp. 181-188, 2006.

[4] E. Bingham and A. Hyvarinen, "A Fast Fixed-point Algorithm for Independent Component Analysis of Complex Valued Signals," *International Journal of Neural Systems*, vol. 10, no. 1, pp. 1-8, 2000.

[5] V. Zarzoso and P. Comon, "Robust Independent Component Analysis by Iterative Maximization of the Kurtosis Contrast With Algebraic Optimal Step Size," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 248-261, 2010.

[6] Y. Li, D. Powers, J. Peach, "Comparison of Blind Source Separation Algorithms," *WSES 2001 Neural Networks and Applications (NNA-01), World Scientific Engineering Society*, pp. 18-21, 2001.

[7] S. Makino, H. Sawada, R. Mukai, S. Araki, "Blind Source Separation of Convolutive Mixtures of Speech in Frequency Domain," *IEICE Trans. Fundamentals*, vol. E88-A, no. 7, pp. 1640-1655, 2005.

[8] M. S. Pedersen, J. Larsen, U. Kjems, L. C. Parra, "A Survey of Convolutive Blind Source Separation Methods," *Springer Handbook on Speech Processing and Speech Communication*, 2007.

[9] N. Mitianoudis and M. E. Davies, "Audio Source Separation of Convolutive Mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 489-497, 2003.

[10] H. Sawada, R. Mukai, S. Araki, S. Makino, "A Robust and Precise Method for Solving the Permutation Problem of Frequency-Domain Blind Source Separation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 530-538, 2004.

[11] H. Sawada, S. Araki, S. Makino, "Measuring Dependence of Bin-wise Separated Signals for Permutation Alginment in Frequency-domain BSS," *IEEE International Symposium on Circuits and Systems*, pp. 3247-3250, 2007.

[12] R. Mazur and A. Mertins, "A Sparsity Based Criterion for Solving the Permutation Ambiguity in Convolutive Blind Source Separation," *International Conference on Acoustic, Speech, and Signal Processing*, pp. 1996-1999, 2011.

[13] K. Rahbar and J. P. Reilly, "A Frequency Domain Method for Blind Source Separation of Convolutive Audio Mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 832-844, 2005.

[14] L. Wang, H. Ding, F. Yin, "A Region-Growing Permutation Alignment Approach in Frequency-Domain Blind Source Separation of Speech Mixtures," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 549-557, 2011.

[15] Z. Chen and L. Chan, "New Approaches for Solving Permutation Indeterminacy and Scaling Ambiguity in Frequency Domain Separation of Convolved Mixtures," *International Joint Conference on Neural Networks*, pp. 911-918, 2011.

[16] R. Mukai, H. Sawada, S. Araki, S. Makino, "Frequency-Domain Blind Source Separation of Many Speech Signals Using Near-Field and Far-Field Models," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1-13, 2006.

[17] H. Sawada, S. Araki, R. Mukai, S. Makino, "Grouping Separated Frequency Components by Estimating Propagation Model Parameters in Frequency-Domain Blind Source Separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1592-1604, 2007.

[18] T. Ngo and S. Nam, "An Expectation-Maximization Method for the Permutation Problem in Frequency-Domain Blind Source Separation," *International Conference on Acoustic, Speech, and Signal Processing*, pp. 17-20, 2010.

[19] I. Lee, T. Kim, T. Lee, "Independent Vector Analysis for Convolutive Blind Speech Separation," *Blind Speech Separation*, pp. 169-192, 2007.

[20] T. Kim, "Real-Time Independent Vector Analysis for Convolutive Blind Source Separation," *IEEE Transactions on Circuits and Systems*, vol. 57, no. 7, pp. 1431-1438, 2010.

[21] I. S. Dhillon, Y. Guan, B. Kulis, "Kernel k-means, Spectral Clustering and Normalized Cuts," *KDD*, 2004.

[22] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, "A Survey of Kernel and Spectral Methods for Clustering," *Pattern Recognition*, vol. 41, pp. 176-190, 2008.

[23] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[24] D. Mavroeidis and E. Bingham, "Enhancing the Stability of Spectral Ordering with Sparsification and Partial Supervision: Application to Paleontological Data," *International Conference on Data Mining*, pp. 462-471, 2008.

[25] Y. Wang, Y. Jiang, Y. Wu, Z. Zhou, "Spectral Clustering on Multiple Manifolds," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1149-1161, 2011.

[26] J. R. Hopgood, P. J. W. Rayner, P. W. T. Yuen, "The Effect of Sensor Placement in Blind Source Separation," *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pp. 95-98, 2001.

[27] F. Nesta, P. Svaizer, M. Omologo, "Convolutive BSS of Short Mixtures by ICA Recursively Regularized Across Frequencies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 624-639, 2011.

[28] U. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395-416, 2007.

[29] A. Y. Ng, M. I. Jordan, Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems*, pp. 849-856, 2001.

[30] C. Ding, X. He, "Linearized Cluster Assignment via Spectral Ordering," *International Conference on Machine Learning*, 2004.

[31] M. Z. Ikram and D. R. Morgan, "Permutation Inconsistency in Blind Speech Separation: Investigation and Solutions," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 1-13, 2005.

[32] http://www.kecl.ntt.co.jp/icl/signal/sawada/demo/bss2to4/index.html

[33] S. Araki, R. Mukai, S. Makino, T. Nishikawa, H. Saruwatari, "The Fundamental Limitation of Frequency Domain Blind Source Separation for Convolutive Mixtures of Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 2, pp. 109-116, 2003.

[34] http://research.ics.tkk.fi/ica/cocktail/cocktail_en.cgi

[35] E. Vincent, R. Gribonval, C. Fevotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462-1469, 2006.