Journal of Systems Engineering and Electronics
Vol. 32, No. 5, October 2021, pp.1097 – 1110

Learning a discriminative high-fidelity dictionary for single channel source separation

TIAN Yuanrong[1,*] and WANG Xing[2]

1. School of Electronic Countermeasure, National University of Defense Technology, Hefei 230037, China;
2. Institute of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China

Abstract: Sparse-representation-based single-channel source separation, which aims to recover each source's signal using its corresponding sub-dictionary, has attracted many scholars' attention. The basic premise of this model is that each sub-dictionary possesses discriminative information about its corresponding source, and this information can be used to recover almost every sample from that source. However, in a more general sense, the samples from a source are composed not only of discriminative information but also common information shared with other sources. This paper proposes learning a discriminative high-fidelity dictionary to improve the separation performance. The innovations are threefold. Firstly, an extra sub-dictionary was combined into a conventional union dictionary to ensure that the source-specific sub-dictionaries can capture only the purely discriminative information for their corresponding sources because the common information is collected in the additional sub-dictionary. Secondly, a task-driven learning algorithm is designed to optimize the new union dictionary and a set of weights that indicate how much of the common information should be allocated to each source. Thirdly, a source separation scheme based on the learned dictionary is presented. Experimental results on a human speech dataset yield evidence that our algorithm can achieve better separation performance than either state-of-the-art or traditional algorithms.

Keywords: single channel source separation, sparse representation, dictionary learning, discrimination, high-fidelity.

DOI: 10.23919/JSEE.2021.000094

1. Introduction

Single-channel source separation (SCSS), as the term suggests, is the task of separating underlying samples from different sources from a single mixed signal [1,2]. This is an underdetermined problem because the number of unknown variables is far greater than the number of observed values. Over the last few decades, many methods have been proposed that exploit prior information about the underlying sources to determine the solution to the SCSS problem, such as computational auditory scene analysis (CASA) [3], the Gaussian mixture model (GMM) method [4] and the hidden Markov model (HMM) method [4,5]. CASA determines a solution mainly by considering the different start and end times of the different sources, whereas GMM and HMM train a generative model for each source to achieve separation. Although these methods have produced many remarkable results, SCSS remains a challenging issue.

In recent years, a great deal of attention has been devoted to sparse representation (SR), which assumes that most of the signals can be coded by very few atoms of a specific codebook (dictionary). The characteristic that only a few atoms are active in a coding procedure is called sparse prior, which is very useful for source separation. Generally, SR based methods for the SCSS problem basically involve two phases. First, the mixed signal features are sparsely represented in a union dictionary composed of several sub-dictionaries. Second, the underlying signal from each source is estimated by linearly combining atoms in the corresponding sub-dictionary with sparse coefficients. Numerous studies have suggested that the excellent performance of SR-based SCSS methods relies heavily on good dictionary properties, e.g., high discriminative capabilities [6−8], which are often obtained through machine learning. Such methods are also called dictionary learning (DL) [9]. One well-known DL method is the K-SVD algorithm [10], which updates atoms to better fit the training samples using a generalized singular value decomposition (SVD) method. Considering that humans sense a physical phenomenon as a whole based on sensing its individual parts, Lee et al. developed a new way to construct representative bases, which is called non-negative matrix factorization (NMF)

Manuscript received July 22, 2020.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (62001489) and the scientific research planning project of National University of Defense Technology (JS19-04).

[11]. Inspired by SR and Daniel's work, Hoyer was the first to incorporate a sparsity constraint into the NMF framework, calling the new method sparse NMF (SNMF) [12]. In addition to learning a single dictionary, coupled dictionary learning (CDL) has become increasingly popular in recent years [13,14].

Based on SR, NMF, SNMF, or CDL, many methods have been proposed for SCSS, among which the representative works are [15], [16] and [17]. All the work reported in [15−17] focused on training reconstruction sub-dictionaries separately and then combining the trained sub-dictionaries to address the SCSS task. However, when sparsely coding a mixed sample against a union dictionary, it is difficult to guarantee that one source-specific sub-dictionary is not active in the other sources' samples, which will damage the separation performance. This problem occurs because the separately trained sub-dictionaries possess not only the discriminative information for their corresponding sources but also the information shared with other sources. Thus, in the testing stage, a source's discriminative information will generally be collected in its corresponding sub-dictionary; however, the shared information is spread throughout the entire union dictionary, which leads to poor recovery performance. This problem has been a topic of study in recent years. Grais et al. [18,19] achieved good performance by adding a penalty term to the objective function to minimize the cross-coherence between source-specific sub-dictionaries. Xu et al. [20,21] proposed a discriminative dictionary learning (DDL) method to penalize the energy that contributes to a specific sub-dictionary but also originates from other sources. However, the work presented in [20,21] was actually rooted in pattern classification; consequently, they can separate coefficients coming from a specific source from the entire set of coefficients but cannot address the SCSS task, in which the input is a mixture. In [22,23], two discriminative sparse non-negative matrix factorization methods (DSNMF) were proposed, but interference between sub-dictionaries still existed in these algorithms, at least to some extent.

With its rapid development, the deep neural network (DNN) has also been successfully applied to SCSS tasks. The basic assumption of DNN based SCSS is that signals from different sources are not overlapped in most of the time-frequency units of their mixture spectrogram. Therefore, by carefully designing a mask on the mixture spectrogram for the target speaker, underlying signals of isolated sources can be separated from the single mixture record. Though a lot of recent works have suggested that DNN is an excellent trainer qualified to complete this job [24−27], it seems hardly to explicitly explain the model, such as features learned from each layer, and takes a lot of time to train a desirable network. Taking these two concerns into account, we exclude this kind of methods for comparison from our current experiments.

In this paper, we propose a new SR-based algorithm to improve SCSS performance. Our major contributions can be summarized as follows.

(i) A new structured dictionary is proposed for SCSS. Concretely, in addition to each source's corresponding sub-dictionary, we incorporate an additional sub-dictionary into the union dictionary. As discussed above, the main obstacle to improve the performance of SR-based SCSS is the interference between sub-dictionaries caused by information shared among different sources. By incorporating this additional sub-dictionary along with proper training, each source's discriminative information is better separated into its corresponding sub-dictionary because the bulk of the shared information is captured by the newly added sub-dictionary. In this sense, the new sub-dictionary is designed to collect the common information shared among different sources; therefore, we call it the common sub-dictionary. Accordingly, we refer to each source's corresponding sub-dictionary as a discriminative sub-dictionary. Because the interference between the discriminative sub-dictionaries is mitigated by the common sub-dictionary, the SCSS performance is improved.

(ii) A two-stage SCSS-task-driven learning algorithm is designed to optimize the dictionary. In the first stage, the dictionary is updated based on the sparse coefficients of mixed signals. This differs from the methods found in [20,21], in which the sparse coefficients used for training the dictionary are separately obtained from each isolated sample. The second stage attempts to optimize a set of weights that indicate how much of the common information should be allocated to each underlying source. These two stages execute iteratively until convergence is achieved. Furthermore, for each isolated source, the energy ratio of common component to discriminative component is calculated on the training dataset after the two-stage learning phase is accomplished. As will be analyzed in detail in Section 3 and Section 5, a dictionary trained using our learning algorithm achieves better performance because the updating of the dictionary based on the sparse coefficients of mixed samples implies that our learning algorithm is driven by the SCSS task. In contrast, the learning algorithms employed in [20,21] are based on the classification task.

(iii) A source separation scheme based on the learned dictionary is proposed. The scheme consists of three successive steps: first, the query mixture is coded against the learned dictionary to obtain sparse coefficients; second, ratio parameters indicating how much of the common in-

formation should be allocated to each underlying source are calculated; third, by employing the dictionary, sparse coefficients, and allocating ratios, the underlying source is reconstructed.

(iv) Extensive experiments on speech separation are presented, and the results are analyzed in detail.

The remainder of the paper is organized as follows. In Section 2, a basic formulation of the SR-based SCSS problem using a well-known dictionary construction method is described, and some notation is clarified. In Section 3, we construct a new dictionary structure and establish the corresponding learning algorithm. Section 4 outlines an SCSS method based on the learned dictionary introduced in Section 3. The results of simulation experiments are presented and analyzed in Section 5. Section 6 concludes the paper.

2. Problem formulation and notation

Our ears and eyes capture an enormous amount of overlapping information every second. This information is often first embodied in high dimensional signals and then passed to subsequent processors. Many studies have proven that the information in a high-dimensional signal, typically referenced to the time or space domain, actually resides in several low-dimensional subspaces that are easier to process [9,28]. Therefore, as long as two signals differ in any aspect, they can be distinguished by projecting them into the proper low-dimensional subspace. For example, independent component analysis (ICA) separates different sources by transforming them into a low-dimensional subspace that minimizes the mutual information between the observed samples. From this perspective, an important aspect of much previous work on signal analysis can be summarized as the selection or construction of an ideal subspace in which samples from different sources can be separated distinctly. Based on SR, the authors of [28] reported impressive results for face recognition by constructing a simple but interesting dictionary whose atoms were selected directly from the raw training samples. Our method is based on this scheme, but includes some modifications that make it suitable for the SCSS task.

In SCSS, a mixed signal is observed, expressed as

$$\boldsymbol{z} = \sum_{s=1}^{N} \boldsymbol{x}_s \tag{1}$$

where $\boldsymbol{x}_s \in \mathbf{R}^{d\times 1}$ is a sample with unit L2 norm from the sth source, and N is the number of underlying sources. For ease of description, N=2 is considered for illustration here, namely,

$$\boldsymbol{z} = \boldsymbol{x}_1 + \boldsymbol{x}_2. \tag{2}$$

From the following description, one can observe that our proposed method is also suitable for the case in which N is more than 2.

The purpose of the SR-based SCSS task is to estimate every $\boldsymbol{x}_s$ from the observed $\boldsymbol{z}$ by using a dictionary. Given n_s training samples, $\{\boldsymbol{x}_s^{(1)}, \boldsymbol{x}_s^{(2)}, \cdots, \boldsymbol{x}_s^{(n_s)}\}$, from the sth source, the method presented in [28] directly used $\boldsymbol{X}_s = [\boldsymbol{x}_s^{(1)}, \boldsymbol{x}_s^{(2)}, \cdots, \boldsymbol{x}_s^{(n_s)}]$ as the sth source-specific sub-dictionary. By combining the samples from all sources, a union dictionary $\boldsymbol{D}$ is formed.

$$\boldsymbol{D} = [\boldsymbol{X}_1, \boldsymbol{X}_2] = [\boldsymbol{x}_1^{(1)}, \cdots, \boldsymbol{x}_1^{(n_1)}, \boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_2^{(n_2)}]. \tag{3}$$

Then, a tested signal $\boldsymbol{y}$, whether a mixed or isolated sample, can be constructed as a linear combination of the samples collected in $\boldsymbol{D}$ under a sparsity constraint:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{c} = c_{11}\boldsymbol{x}_1^{(1)} + \cdots + c_{1n_1}\boldsymbol{x}_1^{(n_1)} + c_{21}\boldsymbol{x}_2^{(1)} + \cdots + c_{2n_2}\boldsymbol{x}_2^{(n_2)} \tag{4}$$

where $\boldsymbol{c} = [c_{11}, \cdots, c_{1n_1}, c_{21}, \cdots, c_{2n_2}]^{\mathrm{T}}$ is a vector made up of the coding coefficients, in which most entries are zero (or approximately zero) and only a few are non-zero. Please note that the elements of $\boldsymbol{c}$ can be either positive or negative. Given $\boldsymbol{D}$, $\boldsymbol{c}$ contains nearly all the information about the input signal $\boldsymbol{y}$. Hence, various applications can be based on this model; for example, in [28], $\boldsymbol{y}$ was assigned to a class by identifying the sub-dictionary corresponding to the minimum reconstruction error. Unlike in [28], we treat the reconstruction $\hat{\boldsymbol{x}}_s$ as an estimate of $\boldsymbol{x}_s$ to accomplish the SCSS task.

$$\hat{\boldsymbol{x}}_s = \boldsymbol{X}_s\boldsymbol{c}_s = c_{s1}\boldsymbol{x}_s^{(1)} + c_{s2}\boldsymbol{x}_s^{(2)} + \cdots + c_{sn_s}\boldsymbol{x}_s^{(n_s)} \tag{5}$$

Although outstanding results have been reported in [20,21,28], the dictionary structure used in [20,21,28] has some drawbacks.

(i) To obtain a sparser representation in the dictionary, one would either need to collect more samples or select a fixed number of samples more carefully. However, increasing the dictionary size also increases the computational complexity for SR.

(ii) Every sub-dictionary (identical to $\boldsymbol{X}_s$ in this case) is constructed separately, and the relationships among the sub-dictionaries are not considered when they are combined. This approach can lead to severe interference in the sparse coding and may result in the need to choose between higher fidelity and stronger discrimination in the SCSS task.

3. Learning a discriminative high-fidelity dictionary

3.1 Adding high discrimination and fidelity capabilities to the dictionary

The shortcomings described above remind us that the uni-

on dictionary used for SR-based SCSS should have the following two main properties. The first is that the sparse coefficients $[c_{s1},c_{s2},\cdots,c_{ns}]^{\mathrm{T}}$ in the s th sub-dictionary should come only from source s and not from others, which we refer to as the "discriminative property". The second is that the signal recovered from a sub-dictionary, $\boldsymbol{X}_s\boldsymbol{c}_s$, should approximate the sample $\boldsymbol{x}_s$ from the corresponding source as closely as possible, which we call the "fidelity property". However, these properties generally conflict each other; thus, they cannot be improved simultaneously. The conflict occurs because higher fidelity requires not only the discriminative information specific to a source but also common information shared with others. In contrast, higher discrimination requires rejecting common information. The premise for this explanation is that $\boldsymbol{x}_s$ is composed of both discriminative information and common information. This premise can be taken to be true because it is extremely rare for samples from two different sources to be entirely different from each other.

Here, we attempt to simultaneously enhance the discriminative property and the fidelity property by adding an additional sub-dictionary to the union dictionary. For convenience, we refer to our work as discriminative high-fidelity dictionary learning (DHFDL).

$$\boldsymbol{D}=[\boldsymbol{D}_1,\boldsymbol{D}_2,\boldsymbol{D}_c] \tag{6}$$

$\boldsymbol{D}$ is generally normalized as its columns with unit L2 norms. The subscript of $\boldsymbol{D}_c$ indicates that this sub-dictionary, which we call the common sub-dictionary, is designed to represent the common information shared among different sources, while the subscript of $\boldsymbol{D}_s$ (where s can be 1 or 2) which is intended to capture the discriminative information of source s, is called a discriminative sub-dictionary. For notational simplicity, we assume that each sub-dictionary contains l atoms; therefore, $\boldsymbol{D}$ is a matrix with d rows and $L=3l$ columns. Consequently, a new DL approach is established as follows. Given a corpus of training samples from two sources, we artificially mix those samples in a pair-wise manner to serve as the input for source separation tasks. Assume that we have n_s samples from the sth source, we will construct a total of $M=n_1n_2$ source separation tasks. Then, the mixed signals along with the isolated signals are fed into an iterative algorithm to learn a suitable $\boldsymbol{D}$ and a pair of associated weights α_1 and α_2, where α_1 and α_2 are used to indicate how much of the common information should be allocated to each underlying source.

Here comes a question that the source to source energy ratio (SSR) of a mixed signal is always different between the training and the testing phase, thus the learned α_1 and α_2 cannot be directly used for SCSS tests under various SSR conditions. To cope with this problem, we make the following reasonable assumption: given a pair of sources, for each isolated source, the energy ratio of common component to discriminative component (ERoCD) remains to be constant no matter what mixing SSR is used. After α_1, α_2 and $\boldsymbol{D}$ are trained to be optimal, the ERoCD can be calculated by the statistical method on the training dataset. On the contrary, in the testing phase, ERoCD can be used to estimate α_1 and α_2 which help assigning the common information of mixture to each underlying source. In the rest of this paper, we denote ERoCD of source i as β_i.

Our learning function can be formulated as (7), where $\|\cdot\|_{\mathrm{F}}\geqslant 0$ and $\|\cdot\|_1$ are functions that calculate the values of the Frobenius and L1 norms, respectively; $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are two matrices that contain the training samples from sources 1 and 2, respectively; $\boldsymbol{D}(:,j)$, $\boldsymbol{D}(:,i)$ and $\boldsymbol{Z}(:,i)$ represent the jth or the ith column of the matrices $\boldsymbol{D}$, $\boldsymbol{C}$ and $\boldsymbol{Z}$, respectively; $\boldsymbol{C}=[\boldsymbol{C}_1^{\mathrm{T}},\boldsymbol{C}_2^{\mathrm{T}},\boldsymbol{C}_c^{\mathrm{T}}]^{\mathrm{T}}$ is the sparse coefficient matrix of $\boldsymbol{Z}$ in $\boldsymbol{D}$; $\boldsymbol{v}$ is the sparse coefficient vector of $\boldsymbol{Z}(:,i)$ to be optimized, $\boldsymbol{C}(:,i)$ denotes the optimal $\boldsymbol{v}$. $\gamma\geqslant 0$ is a tradeoff scalar, and η is a weight scalar that controls how sparsely $\boldsymbol{Z}(:,i)$ is coded; a larger η implies more zero entries in $\boldsymbol{C}(:,i)$, and γ and η mainly depend on the dataset. Numerous experiments analyzing these two parameters are presented in Section 5.

$$\begin{aligned}\min_{\boldsymbol{D},\alpha_1,\alpha_2} J(\boldsymbol{D},\alpha_1,\alpha_2)=&\frac{1}{2}\|(\boldsymbol{X}_1+\boldsymbol{X}_2)-\boldsymbol{D}\boldsymbol{C}\|_{\mathrm{F}}^2+\frac{\gamma}{2}\left(\|\boldsymbol{X}_1-\boldsymbol{D}_1\boldsymbol{C}_1-\alpha_1\boldsymbol{D}_c\boldsymbol{C}_c\|_{\mathrm{F}}^2+\|\boldsymbol{X}_2-\boldsymbol{D}_2\boldsymbol{C}_2-\alpha_2\boldsymbol{D}_c\boldsymbol{C}_c\|_{\mathrm{F}}^2\right)\\ &\text{s.t. }\|\boldsymbol{D}(:,j)\|_2=1,\quad j=1,2,\cdots,L,\\ &\alpha_1+\alpha_2=1,\quad \alpha_1>0;\ \alpha_2>0,\\ \boldsymbol{C}(:,i)=\arg&\left\{\min_{\boldsymbol{v}}\frac{1}{2}\|\boldsymbol{Z}(:,i)-\boldsymbol{D}\boldsymbol{v}\|_2^2+\eta\|\boldsymbol{v}\|_1\right\},\quad i=1,2,\cdots,M;\ \boldsymbol{Z}=\boldsymbol{X}_1+\boldsymbol{X}_2.\end{aligned} \tag{7}$$

The first term of the cost function in (7) measures how well the mixed samples are coded; we call this term the total error (TE). Minimizing the TE ensures that the total energy of the mixed input signal is retained, which is the basic requirement for a successful separation task. The second term of the cost function in (7) represents how

closely the estimated samples approximate the underlying true samples. We refer to this term as the isolated recovery error (IRE). Note that each source's recovery error in IRE is calculated separately. Consequently, a sufficiently small IRE means that $\boldsymbol{D}$, α_1 and α_2 can be used to effectively separate the mixed samples from the current sources. According to the above descriptions, minimizing the IRE implies a smaller TE. Although this is true, we retain the TE in the cost function to ensure sparse coding.

We further study the effects of the IRE and the common sub-dictionary on DL by means of the illustrations in Fig. 1. To allow the separation results to be displayed on a plane, d is reduced to two, and $\boldsymbol{Z}$, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are reduced to column vectors denoted by $\boldsymbol{z}$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively. The discriminative components of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are denoted by two orthogonal vectors $\boldsymbol{x}_1^{(d)}$ and $\boldsymbol{x}_2^{(d)}$, whereas the common components are denoted by two parallel vectors $\boldsymbol{x}_1^{(c)}$ and $\boldsymbol{x}_2^{(c)}$, which are both related to vector $\boldsymbol{x}^{(c)}$ as follows: $\boldsymbol{x}_1^{(c)}=\alpha_1\boldsymbol{x}^{(c)}$ and $\boldsymbol{x}_2^{(c)}=\alpha_2\boldsymbol{x}^{(c)}$.

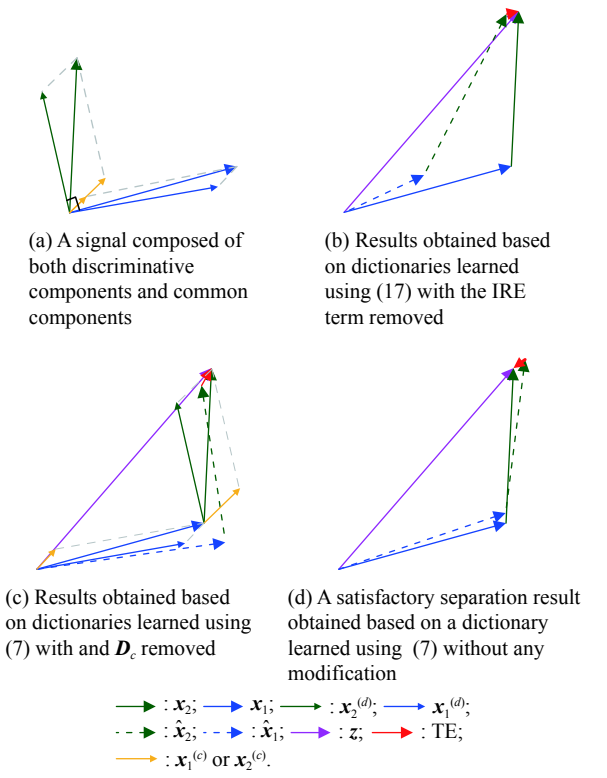

Fig. 1 Illustration of the effects of the IRE and the common sub-dictionary on DL

The decomposition of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is shown in Fig. 1(a). If we remove the IRE from the objective function in (7), the separation result shown in Fig. 1(b) may be obtained. However, although the TE in Fig. 1(b) is sufficiently small, the deviations between both $\boldsymbol{x}_1$ and $\hat{\boldsymbol{x}}_1$ and $\boldsymbol{x}_2$ and $\hat{\boldsymbol{x}}_2$ are too large to accept. Next, we retain the IRE in (7) but remove $\boldsymbol{D}_c$ from $\boldsymbol{D}$ to learn a discriminative dictionary. The separation of the mixed signal using this discriminative dictionary may yield the result shown in Fig. 1(c). In Fig. 1(c), $\boldsymbol{x}_1^{(d)}$ and $\boldsymbol{x}_2^{(d)}$ are approximately parallel to $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ because of the discriminative property of the dictionary, and the TE is also small. However, the large deviation between $\boldsymbol{x}_2$ and $\hat{\boldsymbol{x}}_2$ indicates that the learned dictionary is not suitable for the SCSS task. The separation result shown in Fig. 1(d) is based on the dictionary learned using (7) without any modification and demonstrates the success that can be achieved by satisfying the requirements on both the IRE and TE in this case.

Remark 1 In (7), one can observe that $\boldsymbol{C}$ is obtained from the mixed signal $\boldsymbol{Z}$; we call this coding strategy coding after mixing (CAM). Looking back at (4) and (5), one can also find that CAM is directly used as a key step in the final separation task. Here, we embed CAM in (7) to ensure that our learning algorithm is driven by the separation task. This is quite different from previously proposed separation methods [20,21], which separately code each underlying sample on $\boldsymbol{D}$ and then use the resulting sparse coefficients for DL, in a strategy that we refer to as coding before mixing (CBM). The main benefit of CBM is that one can identify the coefficients of a sample when they are spread throughout other sources' corresponding sub-dictionaries. Thus, penalizing these non-source-specific coefficients by updating their corresponding atoms can improve the discriminative property of the dictionary. Obviously, CBM is most suitable for single-input tasks (e.g., classification or de-noising tasks); however, it is less suitable for the SCSS task, which involves a multi-input task. Principle block diagrams for these two tasks are shown in Fig. 2.

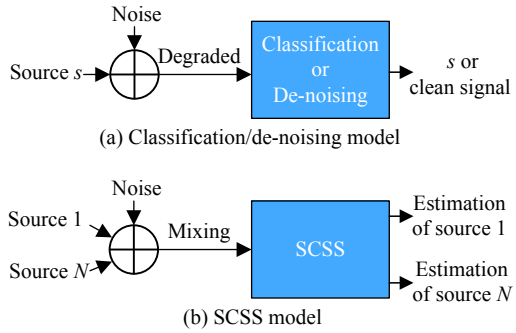

Fig. 2 Structures of the classification/de-noising model and the SCSS model

Furthermore, the CBM strategy is unsuitable for learning dictionaries for SCSS because in general, $\boldsymbol{C}$ is not equal to $\boldsymbol{C}_1^s+\boldsymbol{C}_2^s$, where $\boldsymbol{C}$ represents the sparse coefficients of $\boldsymbol{X}_1+\boldsymbol{X}_2$ and $\boldsymbol{C}_1^s$ and $\boldsymbol{C}_2^s$ are respectively obtained by separately sparsely coding $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ on $\boldsymbol{D}$.

3.2 Optimization

Equation (7) is a typical bi-level optimization problem. The minimization of the objective function is called the upper-level problem, and $\boldsymbol{C}(:,i)=\arg\{\min_{\boldsymbol{v}}\|\boldsymbol{Z}(:,i)-\boldsymbol{D}\boldsymbol{v}\|_2^2+\eta\|\boldsymbol{v}\|_1\}$ (at the bottom of (7)) is called the lower-level problem. This is a special kind of optimization in which optimizing the upper-level problem requires the sparse coefficients $\boldsymbol{C}$ to be known, whereas $\boldsymbol{D}$, which is used to solve the lower-level problem, is the variable optimized in the upper-level problem. Although bi-level problems are usually solved by using descent methods, (7) is difficult to solve because the L1 norm in the lower-level problem is not smooth. Fortunately, Yang et al. addressed a bi-level optimization problem similar to (7) in [13] and proposed an efficient procedure for updating the dictionary atoms. In this section, we follow the routine presented in [13] for updating the dictionary via the stochastic gradient descent algorithm. In addition, the interior-point method is employed to find the optimal α_1 and α_2 values. Thus, our strategy for solving (7) is to iteratively implement two stages for a specified number of times, namely, first updating $\boldsymbol{D}$ and then optimizing α_1 and α_2. After α_1, α_2 and $\boldsymbol{D}$ are updated to be optimal, the ERoCD of each source is calculated.

3.2.1 Updating $\boldsymbol{D}$ with α_1 and α_2 fixed

In the stochastic gradient descent method, the dictionary is updated based on only one training sample during each loop. In a given loop, we let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote a pair of training samples (d-dimensional column vectors) from sources 1 and 2, respectively. Consistent with the notation used in Section 2, we let $\boldsymbol{z}=\boldsymbol{x}_1+\boldsymbol{x}_2$. The sparse coefficients of $\boldsymbol{z}$ are denoted by $\boldsymbol{c}=[\boldsymbol{c}_1^{\mathrm{T}},\boldsymbol{c}_2^{\mathrm{T}},\boldsymbol{c}_c^{\mathrm{T}}]^{\mathrm{T}}$, which is calculated by using (8).

$$\boldsymbol{c}=\arg\left\{\min_{\boldsymbol{v}}\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{v}\|_2^2+\eta\|\boldsymbol{v}\|_1\right\}. \tag{8}$$

Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ denote two index matrices as follows:

$$\begin{cases}\boldsymbol{P}_1=\begin{bmatrix}\boldsymbol{I}_{l\times l} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \alpha_1\boldsymbol{I}_{l\times l}\end{bmatrix}\\ \boldsymbol{P}_2=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_{l\times l} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \alpha_2\boldsymbol{I}_{l\times l}\end{bmatrix}\end{cases} \tag{9}$$

where $\boldsymbol{I}_{l\times l}$ is the identity matrix and $\boldsymbol{0}$ is an $l\times l$ zero matrix. Then, based on the notation defined above, under the assumption that α_1 and α_2 are optimal, (7) can be reformulated as the compact single-level optimization problem shown in (10):

$$\begin{aligned}&\min_{\boldsymbol{D}}J(\boldsymbol{D})=\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{c}\|_2^2+\\ &\frac{\gamma}{2}\left(\|\boldsymbol{x}_1-\boldsymbol{D}\boldsymbol{P}_1\boldsymbol{c}\|_2^2+\|\boldsymbol{x}_2-\boldsymbol{D}\boldsymbol{P}_2\boldsymbol{c}\|_2^2\right)\\ &\text{s.t. }\|\boldsymbol{D}(:,j)\|_2=1,\ j=1,2,\cdots,L.\end{aligned} \tag{10}$$

The major issue with descent methods is the availability of the gradient of J for a feasible $\boldsymbol{D}$. Applying the chain rule, we arrive at

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{D}}=(\boldsymbol{D}\boldsymbol{c}-\boldsymbol{z})^{\mathrm{T}}\left(\boldsymbol{c}+\boldsymbol{D}\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{D}}\right)+\\ &\gamma\left((\boldsymbol{D}\boldsymbol{P}_1\boldsymbol{c}-\boldsymbol{x}_1)^{\mathrm{T}}\left(\boldsymbol{P}_1\boldsymbol{c}+\boldsymbol{D}\boldsymbol{P}_1\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{D}}\right)+(\boldsymbol{D}\boldsymbol{P}_2\boldsymbol{c}-\boldsymbol{x}_2)^{\mathrm{T}}\left(\boldsymbol{P}_2\boldsymbol{c}+\boldsymbol{D}\boldsymbol{P}_2\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{D}}\right)\right).\end{aligned} \tag{11}$$

We let Γ denote the active set of $\boldsymbol{c}$ and Γ^c represent the complementary set of Γ. The gradient of $\boldsymbol{c}$ with respect to $\boldsymbol{D}$ is calculated as follows:

$$\begin{cases}\dfrac{\partial \boldsymbol{c}_\Gamma}{\partial \boldsymbol{D}_\Gamma}=(\boldsymbol{D}_\Gamma^{\mathrm{T}}\boldsymbol{D}_\Gamma)^{-1}\left(\dfrac{\partial \boldsymbol{D}_\Gamma^{\mathrm{T}}}{\partial \boldsymbol{D}_\Gamma}\boldsymbol{z}-\dfrac{\partial(\boldsymbol{D}_\Gamma^{\mathrm{T}}\boldsymbol{D}_\Gamma)}{\partial \boldsymbol{D}_\Gamma}\boldsymbol{c}_\Gamma\right)\\ \dfrac{\partial \boldsymbol{c}_\Gamma}{\partial \boldsymbol{D}_{\Gamma^c}}=\boldsymbol{0}\\ \dfrac{\partial \boldsymbol{c}_{\Gamma^c}}{\partial \boldsymbol{D}}=\boldsymbol{0}\end{cases} \tag{12}$$

where $\boldsymbol{D}_\Gamma$ and $\boldsymbol{D}_{\Gamma^c}$ are matrices that consist of the columns of $\boldsymbol{D}$ in Γ and Γ^c, respectively, and $\boldsymbol{c}_\Gamma$ and $\boldsymbol{c}_{\Gamma^c}$ are composed of the entries of $\boldsymbol{c}$ in Γ and Γ^c, respectively.

After obtaining $\partial J/\partial \boldsymbol{D}$, the dictionary can be updated as

$$\boldsymbol{D}^{(n+1)}=\boldsymbol{D}^{(n)}-\xi\frac{\partial J/\partial \boldsymbol{D}^{(n)}}{\|\partial J/\partial \boldsymbol{D}^{(n)}\|_2} \tag{13}$$

where $\xi=r_0/\sqrt{i/M+1}$. Here, i is the iteration number for the updating of $\boldsymbol{D}$, and r_0 is the initial learning rate. Considering the constraint expressed in (10), we normalize $\boldsymbol{D}$ at the end of each iteration.

3.2.2 Updating α_1 and α_2 with $\boldsymbol{D}$ fixed

If we suppose that $\boldsymbol{D}$ is fixed, then (7) can be reduced to a quadratic programming problem, as shown in (14).

$$\begin{aligned}&\min_{\boldsymbol{\alpha}}\ \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\alpha}+\boldsymbol{q}^{\mathrm{T}}\boldsymbol{\alpha}\\ &\text{s.t. }\alpha_1+\alpha_2=1,\ \alpha_1>0;\ \alpha_2>0\\ &\boldsymbol{H}=\begin{bmatrix}\|\boldsymbol{D}_c\boldsymbol{C}_c\|_{\mathrm{F}}^2 & 0\\ 0 & \|\boldsymbol{D}_c\boldsymbol{C}_c\|_{\mathrm{F}}^2\end{bmatrix}\\ &\boldsymbol{q}=\begin{bmatrix}\sum\sum(\boldsymbol{X}_1-\boldsymbol{D}_1\boldsymbol{C}_1)\odot(\boldsymbol{D}_c\boldsymbol{C}_c)\\ \sum\sum(\boldsymbol{X}_2-\boldsymbol{D}_2\boldsymbol{C}_2)\odot(\boldsymbol{D}_c\boldsymbol{C}_c)\end{bmatrix}\end{aligned} \tag{14}$$

where $\boldsymbol{\alpha}=[\alpha_1,\ \alpha_2]^{\mathrm{T}}$, and $\odot$ stands for the Hadamard product. The constant term and the scalar γ are ignored because they are meaningless for the optimization of $\boldsymbol{\alpha}$. In this study, the interior-point method is used to solve (14).

3.2.3 Calculating ERoCD of each source

After a desirable dictionary $\boldsymbol{D}$ and an optimal weight scalar $\boldsymbol{\alpha}$ are learned, we calculate β_i by

$$\beta_i = \frac{1}{M}\sum_{j=1}^{M}\frac{\left\|\alpha_i \boldsymbol{D}_c \boldsymbol{c}_c^{(j)}\right\|_2}{\left\|\boldsymbol{D}_i \boldsymbol{c}_i^{(j)}\right\|_2},\ i=1,2 \tag{15}$$

where $\boldsymbol{c}_i^{(j)}$ and $\boldsymbol{c}_c^{(j)}$ are cut from the jth training sample's sparse coefficients $\boldsymbol{c}^{(j)} = \arg\{\min_{\boldsymbol{v}} \frac{1}{2}\left\|\boldsymbol{z}_j - \boldsymbol{D}\boldsymbol{v}\right\|_2^2 + \eta\|\boldsymbol{v}\|_1\}$. Since the basic computing unit $\|\bullet\|_2$ of (15) is positive, $\beta_i \geqslant 0$ accordingly.

We summarize the proposed DHFDL algorithm in Algorithm 1. The convergence of DHFDL in practice is shown in Fig. 3.

Algorithm 1 DHFDL

Input Each source's training samples, i.e., $\boldsymbol{X}_1 \in \mathbf{R}^{d\times n_1}$ and $\boldsymbol{X}_2 \in \mathbf{R}^{d\times n_2}$. Initially, $\boldsymbol{D}^{(0)} \in \mathbf{R}^{d\times L}$ and $\boldsymbol{\alpha}^{(0)} \in \mathbf{R}^{2\times 1}$. T is the number of iterations. The model parameters are γ and η.

Output The optimal dictionary $\boldsymbol{D}$ and the ERoCD β_1 and β_2.

Initialization Initialize all the atoms of $\boldsymbol{D}^{(0)}$ as random vectors with unit L2 norms; $\boldsymbol{\alpha}^{(0)}=[0.5, 0.5]^{\mathrm{T}}$.

for t=0, 1, ⋯, T−1 **do**

  for i=1, 2, ⋯, M **do**

    Calculate $\partial J/\partial \boldsymbol{D}^{(t)}$ according to (11) and (12);

    Update $\boldsymbol{D}^{(t)} = \boldsymbol{D}^{(t)} - \dfrac{r_0}{\sqrt{\dfrac{i}{M}+1}}\dfrac{\partial J}{\partial \boldsymbol{D}^{(t)}}$;

    Normalize each column of $\boldsymbol{D}^{(t)}$

  end for

  $\boldsymbol{D}^{(t+1)}=\boldsymbol{D}^{(t)}$

  Set the solution to (14) as $\boldsymbol{\alpha}^{(t+1)}$

end for

Calculate β_1 and β_2 by (15). Please note that $\boldsymbol{D}$ and $\boldsymbol{\alpha}$ involved in (15) are $\boldsymbol{D}^{(T)}$ and $\boldsymbol{\alpha}^{(T)}$.

Return $\boldsymbol{D}^{(T)}$, β_1 and β_2

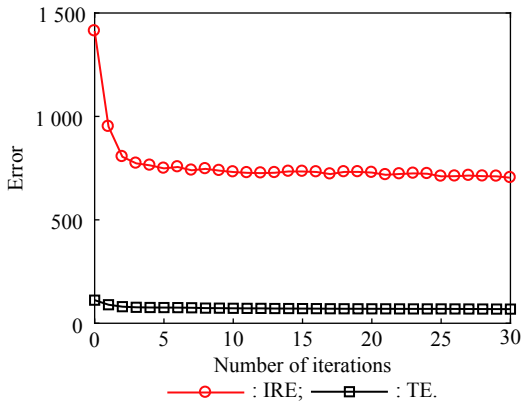

Fig. 3 Curves of the error values associated with (7) in the processing of a small set of the experimental database

From Fig. 3, one can see that the IRE decreases rapidly as the number of iterations increases, and it ultimately tends toward stability. When the number of iterations is small, the IRE drops rapidly. However, the TE remains unchanged because the dictionary is overcomplete and the mixed signal can be fitted well. Also note that the TE is smaller than the IRE in each iteration; this mainly occurs because the sparse coding is performed on the entire union dictionary, which means that it closely tracks the TE term. Because the optimization problem in (7) is highly nonlinear, we can expect the stochastic gradient procedure to find only a local minimum. However, we find that our algorithm works well in practice.

4. SCSS scheme based on learned dictionary

After the training of $\boldsymbol{D}$ and the calculating of β_1 and β_2 are complete, the SCSS problem can be solved by performing the following three steps in sequence.

First, we code a query mixture sample $\boldsymbol{z}=\boldsymbol{x}_1+\boldsymbol{x}_2$ against the dictionary $\boldsymbol{D}$ and obtain the coding coefficients $\boldsymbol{c}$ by solving

$$\boldsymbol{c} = \arg\left\{\min_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{v}\|_2^2 + \eta\|\boldsymbol{v}\|_1\right\} \tag{16}$$

where $\boldsymbol{v}$ is the sparse coefficient vector of $\boldsymbol{z}$ to be optimized, $\boldsymbol{c}=[\boldsymbol{c}_1^{\mathrm{T}}, \boldsymbol{c}_2^{\mathrm{T}}, \boldsymbol{c}_c^{\mathrm{T}}]^{\mathrm{T}}$ denotes the optimal value of $\boldsymbol{v}$, where $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_c$ are the coefficient vectors over the sub-dictionaries $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ and $\boldsymbol{D}_c$, respectively.

Second, estimate α_1 and α_2 as

$$\begin{cases} \alpha_1=\beta_1\|\boldsymbol{D}_1\boldsymbol{c}_1\|_2/(\beta_1\|\boldsymbol{D}_1\boldsymbol{c}_1\|_2+\beta_2\|\boldsymbol{D}_2\boldsymbol{c}_2\|_2) \\ \alpha_2=\beta_2\|\boldsymbol{D}_2\boldsymbol{c}_2\|_2/(\beta_1\|\boldsymbol{D}_1\boldsymbol{c}_1\|_2+\beta_2\|\boldsymbol{D}_2\boldsymbol{c}_2\|_2) \end{cases}. \tag{17}$$

Finally, we can calculate the underlying samples associated with the different sources as follows:

$$\begin{cases} \hat{\boldsymbol{x}}_1 = \boldsymbol{D}_1\boldsymbol{c}_1 + \alpha_1\boldsymbol{D}_c\boldsymbol{c}_c \\ \hat{\boldsymbol{x}}_2 = \boldsymbol{D}_2\boldsymbol{c}_2 + \alpha_2\boldsymbol{D}_c\boldsymbol{c}_c \end{cases}. \tag{18}$$

5. Experimental results and discussion

In this section, the effects of several important parameters on DL are first simulated and then analyzed. Then, the SCSS and classification performances based on the learned dictionary are compared with those of several existing methods. Because the new dictionary structure and the learning algorithm are motivated by the SCSS task, the effects of the algorithm parameters are analyzed based on the SCSS performance.

5.1 Experimental setup

5.1.1 Evaluation dataset and extracted features

All the experiments in this paper were simulated by using the PASCAL computational hearing in multi-source environments (CHiME) speech separation and recogni-

tion challenge dataset [29]. The CHiME evaluation dataset is an extension of the GRID corpus (each of 34 speakers spoke 1 000 utterances) and consists of three parts: training, development and test sets. The training set is composed of 500 clean utterances spoken by each of the 34 speakers, and the development and test sets are composed of 600 utterances at each of 6 signal to noise ratio (SNR) levels, namely, −6 dB, −3 dB, 0 dB, 3 dB, 6 dB, and 9 dB. For each noise level, the content is different. Limited by the performance of our computer, all our experiments run for 10 trails, and the results are the averages. For each trail, we randomly selected 6 out of the 34 speakers (3 men and 3 women) and randomly divided the 500 utterances into two parts, 350 for training and 150 for testing. Also, 10 short sentences were grouped to form a long sentence for each speaker to further reduce the learning task. Thus, in total, we construct 1 225 long mixed training sentences and 225 long clean mixed testing sentences for each pair of speakers. The corresponding utterances of the 6 selected speakers in the CHiME test dataset were then used as the noisy scenario test data to evaluate our method.

Similar to [15,22], the Mel spectra were extracted as features in our experiments. Specifically, each sentence was first enhanced by using a finite impulse response (FIR) filter and then transformed by using a short-time Fourier transform (STFT). Finally, the STFT power spectra were projected to the Mel scale. The FIR coefficient was 0.97, the STFT window was 32 ms (512 sample points at a 16 kHz sampling rate) sliding at 16 ms, and the number of Mel-scale pitches was 80.

5.1.2 Performance metrics

Two metrics are employed to evaluate the separation performance. The first is the signal to recovery error ratio (SER):

$$\mathrm{SER}=\frac{1}{2}\sum_{s=1}^{2}10\lg\left(\|\boldsymbol{f}_s\|_{\mathrm{F}}^2\Big/\left\|\boldsymbol{f}_s-\hat{\boldsymbol{f}}_s\right\|_{\mathrm{F}}^2\right)\tag{19}$$

where $\boldsymbol{f}_s$ is the Mel spectra of $\boldsymbol{X}_s$, and $\hat{\boldsymbol{f}}_s$ is the reconstruction of $\boldsymbol{f}_s$. Obviously, SER keeps close track of the IRE term of our objective function. One may also note that the SER metric is similar to $\mathrm{SNR}^{\mathrm{mel}}$ defined in [22] but with some slight differences, such as the scalar 1/2. The second metric is the signal to interference ratio (SIR) [30].

We selected these two metrics in the Mel spectral domain for two reasons. The first reason is that in practice, the phase information of the underlying isolated speech signals generally cannot be obtained from a single observed mixed speech signal. Therefore, the separated results in the Mel spectral domain cannot be inverted to the spectrum domain or the time domain because of a lack of phase information. The second reason is that the Mel spectrum is a powerful feature of the human voice; numerous works have indicated that speech signal processing tasks [1,4,15,22,26,27,31] can be well realized in the Mel spectral domain. Another note about these two criteria is that the SIR is calculated by using a window because it involves the projection of the reconstructions onto a subspace expanded by the Mel spectra of the two speech signals in an analysis window, whereas $\boldsymbol{f}_s$ used in (19) to calculate the SER is a matrix formed by arranging all of the Mel spectra of the test speech signals. Therefore, in later sections, one can observe that the SER is lower than the SIR, but this has no effect on the comparison of different methods.

5.1.3 Comparison of methods

To evaluate the SCSS performance based on the DHFDL algorithm, for comparison, the K-SVD-, DDL- and DSNMF-LS [23]-based SCSS methods were also simulated. Because the original K-SVD method trains the sub-dictionary for each source separately, for comparison purposes, we present the results obtained by combining these trained sub-dictionaries and coding the mixed speech samples over the resulting union dictionary. Note that the K-SVD based SCSS method described here is similar to that in [20,21] except for some trivialities. The only difference between DHFDL and DDL is that DHFDL trains a union dictionary containing an additional sub-dictionary, namely, the common sub-dictionary. DSNMF-LS is similar with DDL except for its updating rule and non-negative constraint.

5.2 Effects of parameters on the dictionary

Many parameters influence the performance of our method. In this section, we discuss three of them: the size of each sub-dictionary, l; the sparsity parameter, η; and the weight coefficient, γ. Each sub-dictionary contains l atoms; hence, $L=3l$ for DHFDL and $L=2l$ for DDL, DSNMF-LS and K-SVD. r_0 is set to 0.1. The effects of the three parameters are analyzed based on the SER and SIR results achieved in the separation of clean mixed samples.

5.2.1 Obtaining values for l and η

Fixing γ to 0.85, we trained models with l values of 60, 70, 80, 100, and 120 and with η values of 0.06, 0.07, 0.08, 0.1, and 0.15. Other parameters involved in DSNMF-LS are in line with [23]. The performances of the different models in terms of the SER and SIR are listed in Table 1. The best values are shown in bold.

Table 1 Comparison of K-SVD-, DDL-, DSNMF-LS and DHFDL-based SCSS for various values of l and η

Method	l	SER/dB					SIR/dB				
		η=0.06	η=0.07	η=0.08	η=0.1	η=0.15	η=0.06	η=0.07	η=0.08	η=0.1	η=0.15
K-SVD	60	0.75	1.39	1.81	1.82	1.98	6.56	7.83	8.67	8.93	**9.35**
	70	0.48	1.25	1.72	1.91	2.02	5.77	6.98	8.32	8.79	9.06
	80	0.92	1.31	1.66	1.91	**2.07**	5.40	7.01	7.61	8.34	9.02
	100	1.03	1.33	1.47	1.67	1.84	5.55	6.49	7.21	7.50	7.89
	120	1.09	1.33	1.53	1.67	1.79	5.50	6.51	6.91	7.16	7.57
DDL	60	2.15	2.09	2.03	2.00	1.95	9.21	8.95	9.07	9.25	9.34
	70	2.12	2.17	2.13	2.09	2.07	9.40	9.32	9.18	9.34	9.50
	80	2.12	2.18	2.17	2.11	2.12	9.68	9.23	9.30	9.07	9.38
	100	2.08	2.12	2.11	2.16	2.16	**9.96**	9.42	9.23	9.29	9.25
	120	2.00	2.11	2.17	2.22	**2.24**	9.87	9.71	9.72	9.70	9.69
DSNMF-LS	60	2.00	2.11	2.19	2.21	2.19	8.98	9.12	9.38	9.46	9.42
	70	2.08	2.22	2.26	2.26	2.20	9.00	9.22	9.46	9.61	9.57
	80	2.10	2.30	**2.35**	2.30	2.21	9.40	9.76	9.92	**10.09**	9.65
	100	2.00	2.10	2.19	2.16	2.17	9.48	9.62	9.82	9.79	10.00
	120	1.96	1.98	2.12	2.14	2.15	9.03	9.54	9.76	9.84	9.99
DHFDL	60	2.56	2.90	2.87	2.83	2.83	9.45	9.26	8.98	8.75	8.79
	70	2.64	2.80	2.80	2.90	**2.91**	10.48	9.93	8.99	9.29	9.16
	80	2.46	2.68	2.77	2.84	2.90	**10.49**	10.02	9.72	9.34	9.36
	100	2.16	2.58	2.63	2.74	2.81	7.78	10.32	9.95	9.79	9.93
	120	2.13	2.39	2.59	2.64	2.72	7.19	9.63	10.22	10.38	10.21

Table 1 shows that DHFDL outperforms the three compared SCSS methods in terms of both the SER and the SIR at almost all the settings. To display the trends in the SER and the SIR as l and η increase, the averages of all the rows and columns, respectively, for the different methods are plotted in Fig. 4.

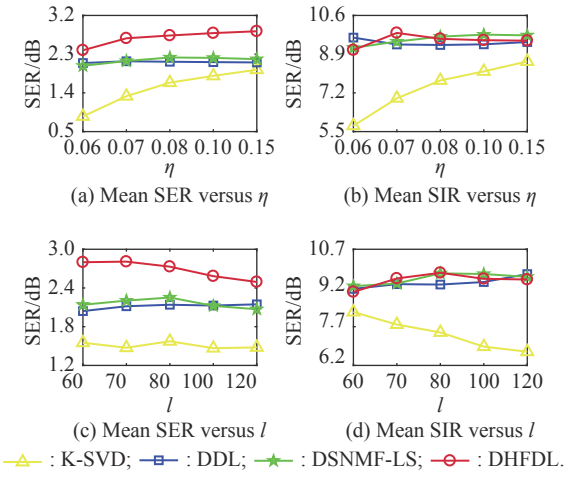

Fig. 4 Comparisons of K-SVD, DDL, DSNMF-LS, and DHFDL-based SCSS with varying η and l values

From the horizontal comparisons in Table 1, one can observe that both the SER and the SIR of K-SVD decrease as η decreases. This is mainly because using more coefficients to code the mixed signals may result in stronger interference because possible correlations between the sub-dictionaries are not considered. From Fig. 4 (a) and Fig. 4(b), one can see that the SER and the SIR obviously increase for DDL, DSNMF-LS and DHFDL slower than those for K-SVD due to the discriminative capabilities of the former's sub-dictionaries.

In detail, Table 1 shows that the SIR of DHFDL generally decreases with increasing η, except in the rows corresponding to l=100, 120. This decreasing SIR trend occurs because a higher η means that less discriminative information is used to reconstruct the input signals. In contrast, the increased values when l=100, 120 mainly occur because the sub-dictionaries are overcomplete when l=100, 120; therefore, a smaller η may result in stronger interference in these cases. The SIR of DDL shows essentially the same trend as that of DHFDL, although with some disturbance caused by the common information present in its sub-dictionaries.

The SER of DHFDL increases as η increases for all l except 60, and the SER of DDL shows a downward trend at l=60, 70 but an upward trend at l=80, 100 and 120. For both SER and SIR of DSNMF-LS, when $l \leqslant 80$ the best values tend to shift to smaller η with l increasing. While $l>80$, the best values appear in the largest η. This performance indicates that DSNMF-LS can achieve better performance over under-complete dictionary ($l \leqslant 80$), but not over the over-complete dictionary ($l>80$).

The vertical comparisons in Table 1 show that when η=0.1, 0.15, the SERs of K-SVD, DSNMF-LS and DHFDL initially increase and then decrease with increasing l, whereas the SER of DDL monotonically increases with increasing l. The main reason the trend increases in the DDL is that its sub-dictionaries contain not only discriminative information but also a relatively large proportion of common information. Therefore, a larger l can cause $\|\boldsymbol{f}_s-\hat{\boldsymbol{f}}\|_{\mathrm{F}}^2$ to decrease, which increases the SER. For DHFDL, a possible reason for the decreasing phase is that a larger l increases the interference caused by the common sub-dictionary. This is even more the case when η=0.06, 0.08, 0.1, where the SER of DHFDL exhibits a monotonically decreasing trend as l increases. For K-SVD and DSNMF-LS, the increasing phase of the SER can be explained by the fact that with a larger l, the dictionary contains richer information when η is relatively large. However, when η is small, such as 0.07 or 0.08, the interference caused by increasing l outweighs the advantage gained from information enrichment.

The advantage of DHFDL becomes clear when its SIR is compared with those of K-SVD, DSNMF-LS and DDL in each column. In detail, when η= 0.08, 0.1, 0.15, the SIR of K-SVD decreases as l increases, whereas the SIRs of DDL, DSNMF-LS and DHFDL mainly show upward trends as l increases. This difference occurs mainly because the sub-dictionaries in DDL and DHFDL have a discriminative capability. Moreover, as shown in Fig. 4(d), the increase rate of SIR decreases the different algorithms as l grows, and the SIR increases for DHFDL is distinctly faster than that for DSNMF-LS and DDL. As previously discussed, the use of too many coefficients to code a mixed signal may result in stronger interference; consequently, the SIR of DHFDL drops to 7.194 8 when η=0.06, 0.07 and l=100, 120.

In summary, our proposed DHFDL algorithm offers significantly improved speech separation performance in terms of both the SER and the SIR. When l is fixed at a specific value, a moderate decrease in η can reduce the reconstruction error and improve the SIR and the SER. However, an η that is too small may result in severe interference because too many coefficients are used to code the input speech signal. A similar trend can be seen when l increases with fixed η; the explanations of the previous trends also apply in this case. For all four methods, one can observe that l=80 and η=0.15 may be chosen as a good trade-off for fair comparison in rest experiments.

5.2.2 Effects of γ

We assigned values of 0.5, 0.75, 1, 2, 4, 6, and 8 to the parameter γ, which balances the contributions of the TE term and the IRE term in the training objective function, to investigate how it influences our method and DDL-based SCSS. The results are shown in Fig. 5. Note that K-SVD and DSNMF-LS were excluded from this experiment because their objective function for learning contains only one term.

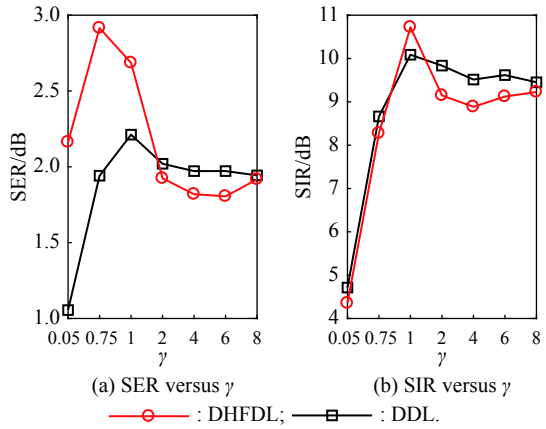

Fig. 5 Comparison of the SCSS results obtained via DHFDL and DDL for different values of γ

As shown in Fig. 5, the SER and SIR of the SCSS methods based on DHFDL and DDL both show an initial rapid climb to a maximum and then drop somewhat. This trend indicates that increasing γ to approximately 0.5 helps the dictionary learn more information about the speech signals, whereas an excessively high value of γ may exacerbate the interference between the sub-dictionaries and reduce the SER and the SIR. From Fig. 5(a), one can also conclude that DHFDL outperforms DDL in terms of the SER when γ<2. This is because in DHFDL, the common sub-dictionary improves the discriminative capability of the source-specific sub-dictionaries. In the case of γ=0.75, the SER of DHFDL is close to 3, whereas DDL achieves an SER of less than 2. As γ increases, the SER drops faster and to a lower value for DHFDL than for DDL because a larger γ amplifies the interference caused by the common sub-dictionary. Fig. 5(b) shows the same qualitative trend as Fig. 5(a) except during the rising phase. The SIR of DHFDL is lower than the SIR of DDL, indicating that the common sub-dictionary may play a more important role when γ is neither too large nor too small. Overall, the best SCSS performances in terms of both the SIR and the SER are achieved by DHFDL, providing evidence that DHFDL is superior to DDL, at least to some extent.

5.3 SCSS results

Based on the results obtained in Section 5.2, in the following experiments, we set the parameters as follows:

l=80, η=0.15, r_0=0.1 and γ=0.85.

5.3.1 Illustrative example

We first provided an example of separating a mixed speech signal. Two short sentences were randomly selected: ‘sgai8a.wav’, spoken by a male speaker with the label ‘id2’, and ‘bgid7s.wav’, spoken by a female speaker with the label ‘id31’. In addition to reconstructing the Mel spectra of the two underlying sentences, we also inverted the Mel spectra into the time domain using the code package provided by Dan Eills on his homepage. Because we assumed the phases of the underlying sentences after the STFT to be unknown, random phases were employed instead. The results are shown in Fig. 6.

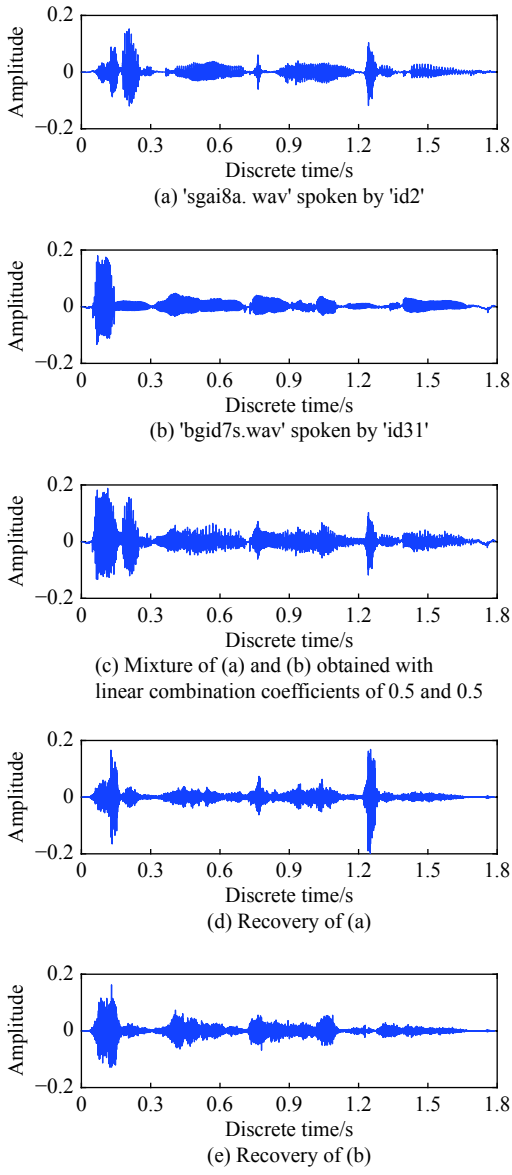

Fig. 6 Separation of two underlying speech signals from a single mixed signal using the proposed SCSS method based on DHFDL

From Fig. 6, one can observe that our proposed method of single-channel source separation based on DHFDL can reconstruct the underlying speech signals well, although with some errors. These errors mainly reside in locations where the amplitudes are small and vary rapidly. This behavior can be explained as follows: sparse coding mainly captures the general features of the input signal using only a few of the coefficients; consequently, the envelopes of Fig. 6(a) and Fig. 6(d) are similar to each other, as are those of Fig. 6(b) and Fig. 6(e). Meanwhile, the window size of the filters used to extract the Mel spectra from the power spectra increases as the frequency increases, meaning that the Mel spectra place more emphasis on low frequencies. In addition, the random phases used to invert the power spectra into the time domain are another important factor that cannot be ignored.

5.3.2 Results for different genders

We further evaluated our method by separating mixed signals composed of speech signals generated by speakers of the same gender and different genders, as presented in this section. The results are shown in Fig. 7.

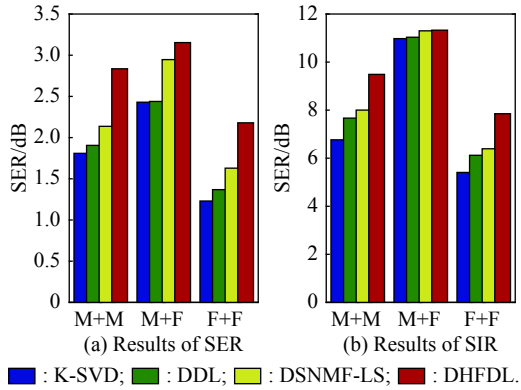

Fig. 7 Performance comparison between SCSS methods based on K-SVD, DDL, DSNMF-LS and DHFDL for mixed speech signals corresponding to different gender combinations

In Fig. 7, M+M, F+M and F+F denote mixed signals generated by mixing speech signals from two men, a man and a woman, and two women, respectively. Fig. 7 reveals that our DHFDL-based method outperforms the SCSS methods based on DDL, DSNMF-LS and K-SVD in terms of both the SER and the SIR in all cases. Notably, when separating the mixed speech signals of two men, the SER and SIR performances of our method exceed those of the DSNMF-LS-based method by nearly 1 dB and 1.8 dB, respectively, and the SER and SIR performances of DSNMF-LS exceed those of DDL by 0.2 dB and

0.5 dB, respectively. K-SVD achieves the worst performance among these four methods. Compared with DSNMF-LS, DHFDL mainly benefits from the common sub-dictionary, whereas the superior performance of DDL and DSNMF-LS with respect to K-SVD can be mainly attributed to the joint optimization of the sub-dictionaries. As discussed in pervious sections, we state that joint optimizing the sub-dictionaries can lift the discriminative property of the sub-dictionaries.

As shown in Fig. 7, all the methods achieve their best performances in terms of both the SER and the SIR in the M+F case, and a sharper contrast is seen among the methods in the M+M and F+F cases. This finding can be explained as follows: speech signals from speakers of the same gender always contain more common information, which DHFDL can handle well because of the common sub-dictionary, whereas the other methods cannot. Although the relative performance improvement of our method compared with the other methods is reduced in the M+F case, the results indicate that the common sub-dictionary can still suppress the interference between the source-specific sub-dictionaries and enhance the separation performance.

5.3.3 Results for different SSRs

In this section, we test our method in different SSRs, and compare the results with K-SVD-, DDL-, and DSNMF-LS-based SCSS methods. For each pair of speakers, we artificially mixed the utterances spoken by them with the energy ratio varing from −2 dB to 2 dB with a step size of 1 dB. The results are listed in Table 2.

Table 2 SER and SIR results for SCSS based on different DL methods at different SSRs

dB

SSR	SER				SIR			
	K-SVD	DDL	DSNMF-LS	DHFDL	K-SVD	DDL	DSNMF-LS	DHFDL
−2	0.94	1.68	1.81	2.04	7.62	8.27	8.43	8.75
−1	1.42	2.05	2.09	2.37	8.34	8.95	9.12	9.13
0	2.07	2.12	2.21	2.90	9.02	9.38	9.65	9.36
1	1.43	2.05	2.18	2.36	8.34	8.97	9.12	9.14
2	0.96	1.67	1.69	1.99	7.67	8.20	8.43	8.69
Average	1.36	1.91	2.00	2.33	8.20	8.75	8.95	9.01

From Table 2, one can observe both SER and SIR for all the methods are getting worse with the absolute value of SSR increasing. This is an expected result since a large absolute value of SSR means the mixture comprises a weak signal and a strong signal. Furthermore, because the dictionary is trained by SSR=0 dB and we do not know the current test mixture's input SSR, the strong signal can interfere with the recovery of the weak signal, on the contrary, a worse recovery of the weak signal can also damage the separating of the strong signal. Both aspects cause the lower separation performance under large SSR cases.

It is also obvious to see that the SCSS performance decreases in order of DHFDL, DSNMF-LS, DDL and K-SVD. We can roughly conclude from this trend that discriminative property can lift the quality of SCSS; adding common sub-dictionary to discriminative sub-dictionaries can further improve the performance of SCSS; non-negative constraint imposed on dictionary benefits SCSS. When compared with DSNMF-LS, the small drops of DHFDL from 0 dB to 1 dB and from 1 dB to 2 dB indicate the validation and superiority of our proposed method.

5.3.4 Noisy scenario

This section presents the results of testing our SCSS method and the other methods considered for comparison on noisy data. For any 6 speakers randomly selected from the entire speaker set of 34 speakers for a trail, we acquired 15 short noise-contaminated sentences for each speaker at each SNR level. At each level (−6 dB, −3 dB, 0 dB, 3 dB, 6 dB, and 9 dB), we mixed pairs of utterances from different speakers with SSR varing from −2 dB to 2 dB, with 1 dB as a step. The average results of different SSRs were calculated as clean signals (although the signals fed into the system were mixed with noise), and the recovery was assessed by using (19) and SIR. The results are listed in Table 3.

Table 3 SER and SIR results for SCSS based on different DL methods at different SNR levels

dB

SNR	SER				SIR			
	K-SVD	DDL	DSNMF-LS	DHFDL	K-SVD	DDL	DSNMF-LS	DHFDL
−6	0.10	0.09	0.13	0.49	0.78	0.95	1.20	2.97
−3	0.27	0.15	0.20	0.64	1.50	1.54	1.78	3.90
0	0.37	0.40	0.48	0.79	2.59	2.38	2.35	5.20
3	0.57	0.87	0.95	1.00	3.12	3.10	3.28	6.02
6	1.23	1.69	1.70	1.86	3.70	4.12	4.94	7.13
9	1.98	2.22	2.48	2.69	4.84	5.44	6.88	7.34
Average	0.75	0.90	0.99	1.24	2.76	2.92	3.41	5.43

In Table 3, each row shows an increasing trend as the SNR increases. This trend is expected, because a higher SNR means less interference from noise. A comparison of different rows reveals that the differences in the metrics between K-SVD and DDL gradually diminish as the SNR decreases. In contrast, the gaps in the metrics between DHFDL and the other methods widen. At SNR=−6 dB, SIR and SER differences between DDL and K-SVD almost disappear, whereas the corresponding gaps between DHFDL and DSNMF-LS are 1.77 dB and 0.36 dB, respectively. This phenomenon can be explained as follows: in the presence of strong noise, the amount of common information increases, which exacerbates the interference between sub-dictionaries. By virtue of its common sub-dictionary, DHFDL has some resistance to this noisy scenario.

6. Conclusions and future work

In this paper, a learning algorithm called DHFDL, which is based on a union dictionary with a novel structure, is proposed to improve the performance of SCSS. Unlike in conventional methods, we consider not only the discriminative information of each isolated source but also the common information shared among different sources, and we jointly optimize the entire union dictionary (which includes both the discriminative sub-dictionaries and a common sub-dictionary). The learned dictionary collects discriminative information in the source-specific sub-dictionaries and collects common information in the common sub-dictionary. This structure is enormously beneficial for separating mixed signals. To solve the objective function for DHFDL, which is a bi-level optimization problem, we propose an algorithm that consists of a dictionary updating step and a weight optimization step; these two steps are performed iteratively until convergence is reached. Numerical experiments confirm the advantages of the proposed method compared with other SCSS algorithms.

A signal's phase is an important information which affects the SCSS well. Though we have demonstrated the superiority of our method through extending experiments in the Mel domain, we cannot revert a signal in the Mel domain to the time domain due to the lack of phase information. However, converting signals in the MFC domain to the time domain is meaningful for speech enhancement and very interesting, and therefore we will conclude the complex mixture signal separation task based on DL in our future research scope.

References

[1] ROWEIS S T. One microphone source separation. Advances in Neural Information Processing Systems, 2000, 13: 793–799.

[2] VINCENT E, VIRTANEN T, GANNOT S. Audio source separation and speech enhancement. New York: John Wiley & Sons Press, 2018.

[3] XU C, RAO W, XIAO X, et al. Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM. Proc. of the International Conference on Acoustics, Speech and Signal Processing, 2018: 6–10.

[4] WANG Q, WOO W L, DLAY S S, et al. Informed single channel speech separation with time-frequency exemplar GMM-HMM model. Proc. of the International Conference on Digital Signal Processing, 2015: 1130–1134.

[5] YEMINY Y R, KELLER Y, GANNOT S. Single microphone speech separation by diffusion-based HMM estimation. EURASIP Journal on Audio, Speech, and Music Processing, 2016. DOI: 10.1186/s13636-016-0094-9.

[6] ZIBULEVSKY M, PEARLMUTTER B A. Blind source separation by sparse decomposition in a signal dictionary. Neural Computing, 2001, 13(4): 863–882.

[7] GOWREESUNKER B V, TEWFIK A H. Blind source separation using monochannel overcomplete dictionaries. Proc. of the International Conference on Acoustics, Speech and Signal Processing, 2008: 33−36.

[8] QIAN Y M, WENG C, CHANG X K, et al. Past review, current progress, and challenges ahead on the cocktail party problem. Frontiers of Information Technology & Electronic Engineering, 2018, 19: 40–63.

[9] BAO C L, JI H, QUAN Y H, et al. Dictionary learning for sparse coding: algorithms and convergence analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2015,

38(7): 1356–1369.
[10] AHARON M, ELAD M, BRUCKSTEIN A. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. on Signal Processing, 2006, 54(11): 4311–4322.
[11] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization. Nature , 1999, 401(6755): 788–791.
[12] HOYER P O. Non-negative sparse coding. Proc. of the 12th Workshop on Neural Networks for Signal Processing, 2002: 557−565.
[13] YANG J C, WANG Z W, LIN Z, et al. Coupled dictionary training for image super-resolution. IEEE Trans. on Image Processing, 2012, 21(8): 3467–3478.
[14] WEI X, SHEN H, LI Y X, et al. Reconstructible nonlinear dimensionality reduction via joint dictionary learning. IEEE Trans. on Neural Networks and Learning Systems, 2018, 30(1): 175–189.
[15] SCHMIDT M N, OLSSON R K. Single-channel speech separation using sparse non-negative matrix factorization. Proc. of the 9th International Conference on Spoken Language Processing, 2006: 2614–2617.
[16] KING B J, ATLAS L. Single-channel source separation using complex matrix factorization. IEEE Trans. on Audio Speech and Language Processing, 2011, 19(8): 2591–2597.
[17] GRAIS E M, ERDOGAN H. Single channel speech music separation using nonnegative matrix factorization with sliding window and spectral masks. Proc. of the 12th Annual Conference on International Speech Communication Association, 2011: 1773–1776.
[18] GRAIS E M, ERDOGAN H. Discriminative nonnegative dictionary learning using cross-coherence penalties for single channel source separation. Proc. of the 14th Annual Conference on International Speech Communication Association, 2013: 808–812.
[19] GANG A, BIYANI P. On discriminative framework for single channel audio source separation. Proc. of the 17th Annual Conference on International Speech Communication Association, 2016: 565–569.
[20] XU Y F, BAO G Z, XU X, et al. Single channel speech separation using sequential discriminative dictionary learning. Signal Processing, 2015, 106: 134–140.
[21] SUN L H, ZHAO C, SU M, et al. Single-channel blind source separation based on joint dictionary with common sub-dictionary. International Journal of Speech Technology, 2018, 21(1): 19–27.
[22] WANG Z, SHA F. Discriminative non-negative matrix factorization for single-channel speech separation. Proc. of the International Conference on Acoustics, Speech and Signal Processing, 2014: 3749–3753.
[23] WENINGER F, ROUX J L, HERSHEY J R, et al. Discriminative NMF and its application to single-channel source separation. Proc. of the 15th Annual Conference on International Speech Communication Association, 2014: 865–869.
[24] WANG Y, WANG D L. Towards scaling up classification-based speech separation. IEEE Trans. on Audio Speech and Language Processing, 2013, 21(7): 1381–1390.
[25] WENINGER F, HERSHEY J R, ROUX J L, et al. Discriminatively trained recurrent neural networks for single-channel speech separation. Proc. of the Global Conference on Signal and Information Processing, 2014: 577–581.
[26] GRAIS E M, PLUMBLEY M D. Single channel audio source separation using convolutional denoising autoencoders. Proc. of the Global Conference on Signal and Information Processing, 2017: 1265–1269.
[27] HERSHEY J R, CHEN Z, ROUX J L, et al. Deep clustering: discriminative embeddings for segmentation and separation. Proc. of the International Conference on Acoustics, Speech and Signal Processing, 2016: 31–35.
[28] WRIGHT J, YANG A Y, GANESH A, et al. Robust face recognition via sparse representation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009, 31(2): 210–227.
[29] CHRISTENSEN H, BARKER J, MA N, et al. The CHiME corpus: a resource and a challenge for computational hearing in multisource environments. Proc. of the 11th Annual Conference on International Speech Communication Association, 2010: 1918–1921.
[30] VINCENT E, GRIBONVAL R, FEVOTTE C. Performance measurement in blind audio source separation. IEEE Trans. on Audio Speech and Language Processing, 2006, 14(4): 1462–1469.
[31] MAJEED S A, HUSAIN H, SAMAD S A, et al. Mel frequency cepstral coefficients (MFCC) feature extraction enhancement in the application of speech recognition: a comparison study. Journal of Theoretical and Applied Information Technology, 2015, 79(1): 38–56.

Biographies

TIAN Yuanrong was born in 1989. He received his B.S. degree in electronics and information engineering from China University of Geosciences in 2011, and M.S. and Ph. D. degrees in communication and information system from Air Force Engineering University in 2014 and 2019 respectively. Currently, he is a lecturer in National University of Defense Technology. His primary research is on pattern analysis of geometric or statistical models in high-dimensional data space and applications in signal interception and analysis, image targets detection and mixing audio separation.
E-mail: tianyuanrong20@nudt.edu.cn

WANG Xing was born in 1965. He received his B.S. and M.S. degrees in communication and electronical system from the former Air Force Engineering College, China, in 1987 and 1990. His Ph. D. degree is obtained in signal and information processing from Northwestern Polytechnical University in 2001. From 1996 to 1999, he was a visiting scholar at Zhukovsky Air Force Engineering College, Moscow, Russia. He served as the director of the Airborne Electronic Countermeasures Laboratory, Air Force Enginering University (AFEU), from 2001 to 2012. He is currently a professor in the Radar and Electronic Countermeasure Department, AFEU. His research interests mainly include radar signal interception and jamming, statistics model analysis, artificial intelligence based on machine learning, circuit design and integration.
E-mail: xwang_mail@yeah.net