Chapter 2
Blind Source Separation Based on Dictionary Learning: A Singularity-Aware Approach

Xiaochen Zhao, Guangyu Zhou, Wei Dai and Wenwu Wang

Abstract This chapter surveys recent works in applying sparse signal processing techniques, in particular, dictionary learning algorithms to solve the blind source separation problem. For the proof of concepts, the focus is on the scenario where the number of mixtures is not less than that of the sources. Based on the assumption that the sources are sparsely represented by some dictionaries, we present a joint source separation and dictionary learning algorithm (SparseBSS) to separate the noise corrupted mixed sources with very little extra information. We also discuss the singularity issue in the dictionary learning process, which is one major reason for algorithm failure. Finally, two approaches are presented to address the singularity issue.

2.1 Introduction

Blind source separation (BSS) has been investigated during the last two decades; many algorithms have been developed and applied in a wide range of applications including biomedical engineering, medical imaging, speech processing, astronomical

The first two authors made equal contribution to this chapter.

X. Zhao (✉) · G. Zhou · W. Dai
Imperial College London, London, UK
e-mail: xiaochen.zhao10@imperial.ac.uk

G. Zhou
e-mail: g.zhou11@imperial.ac.uk

W. Dai
e-mail: wei.dai1@imperial.ac.uk

W. Wang
University of Surrey, Surrey, UK
e-mail: w.wang@surrey.ac.uk

G. R. Naik and W. Wang (eds.), *Blind Source Separation*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-55016-4_2,

imaging, and communication systems. Typically, a linear mixture model is assumed where the mixtures $\boldsymbol{Z} \in \mathbb{R}^{r \times N}$ are described as $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}$. Each row of $\boldsymbol{S} \in \mathbb{R}^{s \times N}$ is a source and $\boldsymbol{A} \in \mathbb{R}^{r \times s}$ models the linear combinations of the sources. The matrix $\boldsymbol{V} \in \mathbb{R}^{r \times N}$ represents additive noise or interference introduced during mixture acquisition and transmission.

Usually in the BSS problem, the only known information is the mixtures $\boldsymbol{Z}$ and the number of sources. One needs to determine both the mixing matrix $\boldsymbol{A}$ and the sources $\boldsymbol{S}$, i.e., mathematically, one needs to solve

$$\min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\|_F^2 .$$

It is clear that such a problem has an infinite number of solutions, i.e., the problem is ill-posed. In order to find the true sources and the mixing matrix (subject to scale and permutation ambiguities), it is often required to add extra constraints to the problem formulation. For example, a well-known method called independent component analysis (ICA) [1] assumes that the original sources are statistically independent. This has led to some widely used approaches such as Infomax [2], maximum likelihood estimation [3], the maximum a posterior (MAP) [4], and FastICA [1].

Sparsity prior is another property that can be used for BSS. Most natural signals are sparse under some dictionaries. The mixtures, viewed as a superposition of sources, are in general less sparse compared to the original sources. Based on this fact, the sparse prior has been used in solving the BSS problem from various perspectives since 2001, e.g., sparse ICA (SPICA) [5] and sparse component analysis (SCA) [6]. In this approach, there is typically no requirement that the original sources have to be independent. As a result, these algorithms are capable of dealing with highly correlated sources, for example, in separating two superposed identical speeches, with one being a few samples delayed version of the other. Jourjine et al. proposed an SCA-based algorithm in [7] aiming at solving the anechoic problem. SCA algorithms look for a sparse representation under predefined bases such as discrete cosine transform (DCT), wavelet, curvelet, etc. Morphological component analysis (MCA) [8] and its extended algorithms for multichannel cases, Multichannel MCA (MMCA) [9], and Generalized MCA (GMCA) [10], are also based on the assumption that the original sources are sparse in different bases instead of explicitly constructed dictionaries. However, these algorithms do not exhibit an outstanding performance since in most cases the predefined dictionaries are too general to offer sufficient details of sources when used in sparse representation.

A method to address this problem is to learn data-specific dictionaries. In [11], the authors advised to train a dictionary from the mixtures/corrupted-images and then decompose it into a few dictionaries according to the prior knowledge of the main components in different sources. This algorithm is used for separating images with different main frequency components (e.g., Cartoon and Texture images) and obtained satisfactory results in image denoising. Starck et al. proposed in [12] to learn dictionary from a set of exemplar images for each source. Xu et al. [13] proposed an algorithm, which allows the dictionaries to be learned from the sources or the mixtures. In most BSS problems, however, dictionaries learned from the mixtures or from similar exemplar images rarely well represent the original sources.

To get more accurate separation results, the dictionaries should be adapted to the unknown sources. The motivation is clear from the assumption that the sources are sparsely represented by some dictionaries. The initial idea of learning dictionaries while separating the sources was suggested by Abolghasemi et al. [14]. They proposed a two-stage iterative process. In this process each source is equipped with a dictionary, which is learned in each iteration, right after the previous mixture learning stage. Considering the size of dictionaries being much larger than the mixing matrix, the main computational cost is on the dictionary learning stage. This two-stage procedure was further developed in Zhao et al. [15]. The method was termed as SparseBSS, which employs a joint optimization framework based on the idea of SimCO dictionary update algorithm [16]. By studying the optimization problem encountered in dictionary learning, the phenomenon of singularity in dictionary update was for the first time discovered. Furthermore, from the viewpoint of the dictionary redundancy, SparseBSS uses only one dictionary to represent all the sources, and is therefore computationally much more efficient than using multiple dictionaries as in [14]. This joint dictionary learning and source separation framework is the focus of this chapter. This framework can be extended potentially to a convolutive or underdetermined model, e.g., apply clustering method to solve the the ill-posed inverse problem in underdetermined model [13]; however, discussion on such an extension is beyond the scope of this chapter. In this chapter, we focus on overdetermined/even determined model.

The remainder of this chapter is organized as follows. Section 2.2 describes the framework of the BSS problem based on dictionary learning. The recently proposed algorithm SparseBSS is introduced and compared in detail with the related benchmark algorithm BMMCA. In Sect. 2.3, we briefly introduce the background of dictionary learning algorithms and then discuss the important observation of the singularity issue, which is a major reason for the failure of dictionary learning algorithms and hence dictionary learning-based BSS algorithms. Later, two available approaches are presented to address this problem. In Sect. 2.5, we conclude our work and discuss some possible extensions.

2.2 Framework of Dictionary Learning-Based BSS Problem

We consider the following linear and instantaneous mixing model. Suppose there are s source signals of the same length, denoted by $\boldsymbol{s}_1, \ \boldsymbol{s}_2, \ldots, \boldsymbol{s}_s$, respectively, where $\boldsymbol{s}_i \in \mathbb{R}^{1\times N}$ is a row vector to denote the ith source. Assume that these sources are linearly mixed into r observation signals denoted by $\boldsymbol{z}_1, \ \boldsymbol{z}_2, \ldots, \boldsymbol{z}_r$ respectively, where $\boldsymbol{z}_j \in \mathbb{R}^{1\times N}$. In the matrix format, denote $\boldsymbol{S} = \left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \ldots, \boldsymbol{s}_s^T\right]^T \in \mathbb{R}^{s\times N}$ and $\boldsymbol{Z} = \left[\boldsymbol{z}_1^T, \boldsymbol{z}_2^T, \ldots, \boldsymbol{z}_r^T\right]^T \in \mathbb{R}^{r\times N}$. Then the mixing model is given by

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}, \tag{2.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{r \times s}$ is the mixing matrix and $\boldsymbol{V} \in \mathbb{R}^{r \times N}$ is denoted as zero mean additive Gaussian noise. We also assume that $r \geq s$, i.e., the underdetermined case will not be discussed here.

2.2.1 Separation with Dictionaries Known in Advance

For some BSS algorithms, such as MMCA [9], orthogonal dictionaries $\boldsymbol{D}_i$'s are required to be known a priori. Each source $\boldsymbol{s}_i$ is assumed to be sparsely represented by a different $\boldsymbol{D}_i$. Hence, we have $\boldsymbol{s}_i = \boldsymbol{D}_i \boldsymbol{x}_i$ with $\boldsymbol{x}_i$'s being sparse. Given the observation $\boldsymbol{Z}$ and the dictionaries $\boldsymbol{D}_i$'s, MMCA [9] aims to estimate the mixing matrix and sources, based on the following form:

$$\min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \sum_{i=1}^{n} \lambda_i \left\| \boldsymbol{s}_i \boldsymbol{D}_i^{\dagger} \right\|_1 . \tag{2.2}$$

Here $\lambda_i > 0$ is the weighting parameter determined by the noise deviation σ, $\|\cdot\|_F$ represents the Frobenius norm, $\|\cdot\|_1$ is the ℓ_1 norm and $\boldsymbol{D}_i^{\dagger}$ denotes the pseudo-inverse of $\boldsymbol{D}_i$. Predefined dictionaries generated from typical mathematical transforms, e.g., DCT, wavelets and curvelets, do not target particular sources, and thus do not always provide sufficiently accurate reconstruction and separation results. Elad et al. [11] designed a method to first train a redundant dictionary by K-SVD algorithm in advance, and then decompose it into a few dictionaries, one for each source. This method works well when the original sources have components that are largely different from each other under some unknown mathematical transformations (e.g. Cartoon and Texture images under the DCT transformation). Otherwise, the dictionaries found may not be appropriate in the sense that they may fit better the mixtures rather than the sources.

2.2.2 Separation with Unknown Dictionaries

2.2.2.1 SparseBSS Algorithm Framework

According to the authors' knowledge, BMMCA and SparseBSS are the two most recent BSS algorithms, which implement the idea of performing source separation and dictionary learning simultaneously. Due to space constraints, we focus on Sparse BSS in this chapter. In SparseBSS, one assumes that all the sources can be sparsely represented under the same dictionary. In order to obtain enough training samples for dictionary learning, multiple overlapped segments (patches) of the sources are taken. To extract small overlapped patches from the source image $\boldsymbol{s}_i$, a binary matrix $\boldsymbol{P}_k \in \mathbb{R}^{n \times N}$ is defined as a patching operator[1] [15]. The product $\boldsymbol{P}_k \cdot \boldsymbol{s}_i^T \in \mathbb{R}^{n \times 1}$ is needed to obtain and vectorize the kth patch of size $\sqrt{n} \times \sqrt{n}$ taken from image $\mathcal{S}_i$. Denote $\boldsymbol{P} = [\boldsymbol{P}_1, \ldots, \boldsymbol{P}_K] \in \mathbb{R}^{n \times KN}$, where K is the number of patches taken from each image. Then the extraction of multiple sources $\boldsymbol{S}$ is defined as $\mathcal{P}\boldsymbol{S} = ([\boldsymbol{P}_1, \ldots, \boldsymbol{P}_K]) \cdot \left(\left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \ldots, \boldsymbol{s}_s^T\right] \otimes \boldsymbol{I}_K\right) = \boldsymbol{P} \cdot (\boldsymbol{S}^T \otimes \boldsymbol{I}_K) \in \mathbb{R}^{n \times Ks}$, where symbol $\otimes$ denotes the Kronecker product and $\boldsymbol{I}_K$ indicates the identity matrix. The computational cost associated with converting from images to patches is low. Each column of $\mathcal{P}\boldsymbol{S}$ represents one vectorized patch. We sparsely represent $\mathcal{P}\boldsymbol{S}$ by using only one dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ and a sparse coefficient matrix $\boldsymbol{X} \in \mathbb{R}^{d \times Ks}$, which suggests $\mathcal{P}\boldsymbol{S} \approx \boldsymbol{D}\boldsymbol{X}$. This is different from BMMCA, where multiple dictionaries are used for multiple sources.

With these notations, the BSS problem is formulated as the following joint optimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{S},\boldsymbol{D},\boldsymbol{X}} \lambda \left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + \left\|\mathcal{P}^{\dagger}(\boldsymbol{D}\boldsymbol{X}) - \boldsymbol{S}\right\|_F^2 . \tag{2.3}$$

The parameter λ is introduced to balance the measurement error and the sparse approximation error, and $\boldsymbol{X}$ is assumed to be sparse.

To find the solution of the above problem, we propose a joint optimization algorithm to iteratively update the following two pairs of variables $\{\boldsymbol{D}, \boldsymbol{X}\}$ and $\{\boldsymbol{A}, \boldsymbol{S}\}$ over two stages until a (local) minimizer is found. Note that in each stage there is only one pair of variables to be updated simultaneously by keeping the other pair fixed.

- Dictionary learning stage

$$\min_{\boldsymbol{D},\boldsymbol{X}} \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 , \tag{2.4}$$

- Mixture learning stage

$$\min_{\boldsymbol{A},\boldsymbol{S}} \lambda \left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 . \tag{2.5}$$

Without being explicit in (2.3), a sparse coding process is involved where greedy algorithms, such as orthogonal matching pursuit (OMP) [17] and subspace pursuit (SP), [18] are used to solve

$$\min_{\boldsymbol{X}} \ \|\boldsymbol{X}\|_0 , \ \text{ s.t. } \ \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}(\boldsymbol{S})\right\|_F^2 \le \epsilon,$$

where $\|\boldsymbol{X}\|_0$ counts the number of nonzero elements in $\boldsymbol{X}$, the dictionary $\boldsymbol{D}$ is assumed fixed, and $\epsilon > 0$ is an upper bound on the sparse approximation error.

During the optimization, further constraints are made on the matrices $\boldsymbol{A}$ and $\boldsymbol{D}$. Consider the dictionary learning stage. Since the performance is invariant to scaling and permutations of the dictionary codewords (columns of $\boldsymbol{D}$), we follow the

[1] Note that in this chapter $\boldsymbol{P}_k$ is defined as a patching operator for image sources. The patching operator for audio sources can be similarly defined as well.

convention in the literature, e.g., [16], and enforce the dictionary to be updated on the set

$$\mathcal{D} = \left\{ \boldsymbol{D} \in \mathbb{R}^{n \times d} : \ \left\| \boldsymbol{D}_{:,i} \right\|_2 = 1, \ 1 \le i \le d \right\}, \tag{2.6}$$

where $\boldsymbol{D}_{:,i}$ stands for the ith column of $\boldsymbol{D}$. A detailed description of the advantage by adding this constraint can be found in [16]. Sparse coding, once performed, provides information about which elements of $\boldsymbol{X}$ are zeros and which are nonzeros. Define the sparsity pattern by $\Omega = \left\{ (i,j) : \ \boldsymbol{X}_{i,j} \neq 0 \right\}$, which is the index set of the nonzero elements of $\boldsymbol{X}$. Define $\mathcal{X}_\Omega$ as the set of all matrices conforming to the sparsity pattern Ω. This is the feasible set of the matrix $\boldsymbol{X}$. The optimization problem for the dictionary learning stage can be written as

$$\begin{aligned} \min_{\boldsymbol{D} \in \mathcal{D}} f_\mu (\boldsymbol{D}) &= \min_{\boldsymbol{D} \in \mathcal{D}} \ \min_{\boldsymbol{X} \in \mathcal{X}_\Omega} \ \| \boldsymbol{D}\boldsymbol{X} - \mathcal{P}(\boldsymbol{S}) \|_F^2 + \mu \| \boldsymbol{X} \|_F^2 , \\ &= \min_{\boldsymbol{D} \in \mathcal{D}} \ \min_{\boldsymbol{X} \in \mathcal{X}_\Omega} \ \left\| \begin{bmatrix} \mathcal{P}(\boldsymbol{S}) \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{D} \\ \sqrt{\mu} \boldsymbol{I} \end{bmatrix} \boldsymbol{X} \right\|_F^2 . \end{aligned} \tag{2.7}$$

The term $\mu \| \boldsymbol{X} \|_F^2$ introduces a penalty to alleviate the singularity issue. See more details in Sect. 2.3.3.

In the mixture learning stage, similar to the dictionary learning stage, we constrain the mixing matrix $\boldsymbol{A}$ in the set

$$\mathcal{A} = \left\{ \boldsymbol{A} \in \mathbb{R}^{r \times s} : \ \left\| \boldsymbol{A}_{:,i} \right\|_2 = 1, \ 1 \le i \le s \right\}. \tag{2.8}$$

This constraint is necessary. Otherwise, if the mixing matrix $\boldsymbol{A}$ is scaled by a constant c and the source $\boldsymbol{S}$ is inversely scaled by c^{-1}, then for any $\{\boldsymbol{A}, \boldsymbol{S}\}$ we can always find a solution $\{c\boldsymbol{A}, c^{-1}\boldsymbol{S} | c > 1\}$, which further decreases the objective function (2.3) from $\lambda \| \boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S} \|_F^2 + \| \boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S} \|_F^2$ to $\lambda \| \boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S} \|_F^2 + c^{-2} \| \boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S} \|_F^2$. Now if we view the sources $\boldsymbol{S} \in \mathbb{R}^{s \times n}$ as a "sparse" matrix with the sparsity pattern $\Omega' = \{(i,j) : \ 1 \le i \le s, \ 1 \le j \le N\}$. Then, the optimization problem for the mixture learning stage is exactly the same as that for the dictionary learning stage:

$$\begin{aligned} \min_{\boldsymbol{A} \in \mathcal{A}} f_\lambda (\boldsymbol{A}) &= \min_{\boldsymbol{A} \in \mathcal{A}} \ \min_{\boldsymbol{S} \in \mathbb{R}^{s \times n}} \ \lambda \| \boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S} \|_F^2 + \left\| \mathcal{P}^\dagger (\boldsymbol{D}\boldsymbol{X}) - \boldsymbol{S} \right\|_F^2 \\ &= \min_{\boldsymbol{A} \in \mathcal{A}} \ \min_{\boldsymbol{S} \in \mathcal{X}_{\Omega'}} \ \left\| \begin{bmatrix} \sqrt{\lambda} \boldsymbol{Z} \\ \mathcal{P}^\dagger (\boldsymbol{D}\boldsymbol{X}) \end{bmatrix} - \begin{bmatrix} \sqrt{\lambda} \boldsymbol{A} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{S} \right\|_F^2 , \end{aligned} \tag{2.9}$$

where the fact that $\mathbb{R}^{s \times n} = \mathcal{X}_{\Omega'}$ has been used. As a result, the SimCO mechanism can be directly applied. Here, we do not require the prior knowledge of the scaling matrix in front of the true mixing matrix [10], as otherwise required in MMCA and GMCA algorithms.

To conclude this section, we emphasize the following treatment of the optimization problems (2.7) and (2.9). Both involve a joint optimization over two variables, i.e., $\boldsymbol{D}$ and $\boldsymbol{X}$ for (2.7) and $\boldsymbol{A}$ and $\boldsymbol{S}$ for (2.9). Note that if $\boldsymbol{D}$ and $\boldsymbol{A}$ are fixed, then the optimal $\boldsymbol{X}$ and $\boldsymbol{S}$ can be easily computed by solving the corresponding least squares problems. Motivated by this fact, we write (2.7) and (2.9) as $\min_{\boldsymbol{D}\in\mathcal{D}} f_\mu(\boldsymbol{D})$ and $\min_{\boldsymbol{A}\in\mathcal{A}} f_\lambda(\boldsymbol{A})$, respectively, when $f_\mu(\boldsymbol{D})$ and $f_\lambda(\boldsymbol{A})$ are properly defined in (2.7) and (2.9). In this way, the optimization problems, at least from the surface, only involve one variable. This helps the discovery of the singularity issue and the developments of handling singularity. See Sect. 2.3 for details.

2.2.2.2 Implementation Details in SparseBSS

Most optimization methods are based on line search strategies. The dictionaries at the beginning and the end of the kth iteration, denoted by $\boldsymbol{D}^{(k)}$ and $\boldsymbol{D}^{(k+1)}$, respectively, can be related by $\boldsymbol{D}^{(k+1)} = \boldsymbol{D}^{(k)} + \alpha^{(k)}\boldsymbol{\eta}^{(k)}$ where $\alpha^{(k)}$ is an appropriately chosen step size and $\boldsymbol{\eta}^{(k)}$ is the search direction. The step size $\alpha^{(k)}$ can be determined by *Armijo condition* or *Golden selection* presented in [19]. The search direction $\boldsymbol{\eta}^{(k)}$ can be determined by a variety of gradient methods [19, 20]. The decision of $\boldsymbol{\eta}^{(k)}$ plays the key role, which directly affects the convergence rate of the whole algorithm. Generally speaking, a Newton direction is a preferred choice (compared with the gradient descent direction) [19]. In many cases, direct computation of the Newton direction is computationally prohibitive. Iterative methods can be used to search the Newton direction. Take the Newton Conjugate Gradient (Newton CG) method as an example. It starts with the gradient descent direction $\boldsymbol{\eta}_0$ and iteratively refines it toward the Newton direction. Denote the gradient of $f_\mu(\boldsymbol{D})$ as $\nabla f_\mu(\boldsymbol{D})$. Denote $\nabla_{\boldsymbol{\eta}}\left(\nabla f_\mu(\boldsymbol{D})\right)$ as the directional derivative of $\nabla f_\mu(\boldsymbol{D})$ along $\boldsymbol{\eta}$ [21]. In each line search step of the Newton CG method, instead of computing the Hessian $\nabla^2 f_\mu(\boldsymbol{D}) \in \mathbb{R}^{md\times md}$ explicitly, one only needs to compute $\nabla_{\boldsymbol{\eta}}\left(\nabla f_\mu(\boldsymbol{D})\right) \in \mathbb{R}^{m\times d}$. The required computational and storage resources are therefore much reduced.

When applying the Newton CG to minimize $f_\mu(\boldsymbol{D})$ in (2.7), the key computations are summarized below. Denote $\tilde{\boldsymbol{D}} = \left[\boldsymbol{D}^T \; \mu\boldsymbol{I}\right]^T$ and let $\Omega(:,j)$ be the index set of nonzero elements in $\boldsymbol{X}_{:,j}$. We consider $\tilde{\boldsymbol{D}}_i = \tilde{\boldsymbol{D}}_{:,\Omega(:,i)} \in \mathbb{R}^{(m+r)\times r}$ with $m > r$. Matrix $\tilde{\boldsymbol{D}}_i$ is a full column rank tall matrix. We denote

$$f_i(\tilde{\boldsymbol{D}}_i) = \min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i - \tilde{\boldsymbol{D}}\boldsymbol{x}_i\|_2^2$$

and the optimal

$$\boldsymbol{x}_i^* = \arg\min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i - \tilde{\boldsymbol{D}}\boldsymbol{x}_i\|_2^2.$$

Denote $\tilde{\boldsymbol{D}}_i^\dagger$ as the pseudo-inverse of $\tilde{\boldsymbol{D}}_i$. Then we have $\frac{\partial f}{\partial \boldsymbol{x}_i}\mid_{\boldsymbol{x}_i^*} = \boldsymbol{0}$, where $\boldsymbol{x}_i^* = \tilde{\boldsymbol{D}}_i^\dagger \boldsymbol{y}_i$, and $\nabla f_i(\tilde{\boldsymbol{D}}_i)$ can be written as

$$\nabla f_i(\tilde{\boldsymbol{D}}_i) = \frac{\partial f}{\partial \tilde{\boldsymbol{D}}_i} + \frac{\partial f}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \tilde{\boldsymbol{D}}_i} = -2(\boldsymbol{y}_i - \tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^*)\boldsymbol{x}_i^{*T} + \boldsymbol{0} \tag{2.10}$$

To compute $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)$, we have

$$\begin{aligned}\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right) &= 2\nabla_{\boldsymbol{\eta}}\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T}\\ &= 2\nabla_{\boldsymbol{\eta}}\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\tilde{\boldsymbol{D}}_i\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T}\\ &= 2\boldsymbol{\eta}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\tilde{\boldsymbol{D}}_i\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^*\boldsymbol{x}_i^{*T} + 2\left(\tilde{\boldsymbol{D}}_i\boldsymbol{x}_i^* - \boldsymbol{y}_i\right)\nabla_{\boldsymbol{\eta}}\boldsymbol{x}_i^{*T}, \qquad (2.11)\end{aligned}$$

where $\nabla_{\boldsymbol{\eta}}\boldsymbol{x}^*$ is relatively easy to obtain,

$$\nabla_{\boldsymbol{\eta}}x^* = -\left(\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}}\right)^{-1}\left(\left(\tilde{\boldsymbol{D}}^T\boldsymbol{\eta} + \boldsymbol{\eta}^T\tilde{\boldsymbol{D}}\right)\tilde{\boldsymbol{D}}^{\dagger} - \boldsymbol{\eta}^T\right)\boldsymbol{y}. \qquad (2.12)$$

From the definition of $\tilde{\boldsymbol{D}}_i, \boldsymbol{D}_i$ is a submatrix of $\tilde{\boldsymbol{D}}_i$, therefore $\nabla f_i(\boldsymbol{D}_i)$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\boldsymbol{D}_i)\right)$ are also, respectively, submatrices of $\nabla f_i(\tilde{\boldsymbol{D}}_i)$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)$, i.e., $\nabla f_i(\boldsymbol{D}_i) = \left(\nabla f_i\left(\tilde{\boldsymbol{D}}_i\right)\right)_{1:m,:}$ and $\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\boldsymbol{D}_i)\right) = \left(\nabla_{\boldsymbol{\eta}}\left(\nabla f_i(\tilde{\boldsymbol{D}}_i)\right)\right)_{1:m,:}$.

In addition, it is also worth noting that the SparseBSS model, using one dictionary to sparsely represent all the sources will get almost the same performance as using multiple but same-sized dictionaries when the dictionary redundancy d/n is large enough. As a result, it is reasonable to train only one dictionary for all the sources. An obvious advantage of using one dictionary is that the computational cost does not increase when the number of sources increases.

2.2.3 Blind MMCA and Its Comparison to SparseBSS

BMMCA [14] is another recently proposed BSS algorithm based on adaptive dictionary learning. Without knowing dictionaries in advance, the BMMCA algorithm also trains dictionaries from the observed mixture $\boldsymbol{Z}$. Inspired by the hierarchical scheme used in MMCA and the update method in K-SVD, the separation model in BMMCA is made up of a few rank-1 approximation problems, where each problem targets on the estimation of one particular source

$$\min_{\boldsymbol{A}_{:,i},\boldsymbol{s}_i,\boldsymbol{D}_i,\boldsymbol{X}_i} \lambda\left\|\boldsymbol{E}_i - \boldsymbol{A}_{:,i}\boldsymbol{s}_i\right\|_F^2 + \|\boldsymbol{D}_i\boldsymbol{X}_i - \mathcal{R}\boldsymbol{s}_i\|_2^2 + \mu\,\|\boldsymbol{X}_i\|_0\,. \qquad (2.13)$$

Different from the operator $\mathcal{P}$ defined earlier in SparseBSS algorithm, the operator $\mathcal{R}$ in BMMCA is used to take patches from only one estimated image $\boldsymbol{s}_i$. $\boldsymbol{D}_i$ is the trained dictionaries for representing source $\boldsymbol{s}_i$. $\boldsymbol{E}_i$ is the residual which can be written as

$$\boldsymbol{E}_i = \boldsymbol{Z} - \sum_{j\neq i}\boldsymbol{A}_{:,j}\boldsymbol{s}_j. \qquad (2.14)$$

Despite being similar in problem formulation, BMMCA and SparseBSS differ in terms of whether the sources share a single dictionary in dictionary learning. In the SparseBSS algorithm, only one dictionary is used to provide sparse representations for all sources. BMMCA requires multiple dictionaries, one for each source. In the mixing matrix update, BMMCA imitates the K-SVD algorithm by splitting the steps of update and normalization. Such two-step based approach does not bring the expected optimality of $\boldsymbol{A} \in \mathcal{A}$, thereby giving inaccurate estimation, while SparseBSS keeps $\boldsymbol{A} \in \mathcal{A}$ during the optimization process. In BMMCA, the authors claim that the ratio between the parameter λ and the noise standard deviation σ is fixed to 30, which will not guarantee good estimation results at various noise levels.

2.3 Dictionary Learning and the Singularity Issue

As is clear from previous discussions, dictionary learning plays an essential role in solving the BSS problem when the sparse prior is used, and hence is the focus of this section. We first briefly introduce the relevant background, then discuss an interesting phenomenon, the singularity issue in the dictionary update stage, and finally present two approaches to handle the singularity issue. For readers who are more interested in the SparseBSS algorithm themselves may consider this section as optional and skip to Sect. 2.4.

2.3.1 Brief Introduction to Dictionary Learning Algorithms

One of the earliest dictionary learning algorithms is the method of optimal directions (MOD) [22] proposed by Engan et al. The main idea is as follows: in each iteration, one first fixes the dictionary and uses OMP [17] or FOCUSS [23] to update the sparse coefficients, then fixes the obtained sparse coefficients and updates the dictionary in the next stage. MOD was later modified to iterative least squares algorithm (ILS-DLA) [24] and recursive least squares algorithm (RLS-DLA) [25]. Aharon et al. developed the K-SVD algorithm [26], which can be viewed as a generalization of the K-means algorithm. In each iteration, the first step is to update the sparse coefficients in the same way as in MOD. Then in the second step, one fixes the sparse pattern, and updates the dictionary and the nonzero coefficients simultaneously. In particular, the codewords in the dictionary are sequentially selected: the selected codeword and the corresponding row of the sparse coefficients are updated simultaneously by using singular value decomposition (SVD). More recently, Dai et al. [16] considered the dictionary learning problem from a new perspective. They formulated dictionary learning as an optimization problem on manifolds and developed simultaneous codeword optimization (SimCO) algorithm. In each iteration SimCO allows multiple codewords of the dictionary to be updated with corresponding rows of the sparse coefficients jointly. This new algorithm can be viewed as a generalization of both MOD and K-SVD. Some other dictionary learning algorithms are also developed in the past decade targeting on various circumstances. For example, based on stochastic approximations, Mairal et al. [27] proposed an online algorithm to address the problem with large data sets.

Theoretical or in-depth analysis about the dictionary learning problem was mean time in progress as well. Gribonval et al. [28], Geng et al. [29], and Jenatton et al. [30] studied the stability and robustness of the objective function under different probabilistic modeling assumptions, respectively. In addition, Dai et al. observed in [16] that the dictionary update procedure may fail to converge to a minimizer. This is a common phenomenon happening in MOD, K-SVD, and SimCO. Dai et al. further observed that ill-conditioned dictionaries, rather than stationary dictionaries, are the major reason that has led to the failure of the convergence. To alleviate this problem, Regularized SimCO was proposed in [16]. Empirical performance improvement was observed. The same approach was also considered in [31], however, without detailed discussion on the singularity issue. More recently, the fundamental drawback of regularized SimCO was demonstrated using an artificial example [32]. To further handle the singularity issue, a Smoothed SimCO [33] was proposed by adding multiplicative terms rather than additive regularization terms to the objective function.

2.3.2 Singularity Issue and Its Impacts

In dictionary update stage of existing mainstream algorithms, singularity is observed as the major reason leading to failures [16, 33]. Simulations in [16] suggests that the mainstream algorithms fail mainly because of singular points in the objective function rather than non-optimal stationary points. As dictionary learning is an essential part of the aforementioned SparseBSS, the singularity issue also has negative impact on the overall performance of BSS. To explain the singularity issue in dictionary update, we first formally define the singular dictionaries.

Definition 1 *A dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times d}$ is singular under a given sparsity pattern Ω if there exists an $i \in [n]$ such that the corresponding sub-dictionary $\boldsymbol{D}_i \triangleq \boldsymbol{D}_{:,\Omega(:,i)}$ is column rank deficient. Or equivalently, the minimum singular value of $\boldsymbol{D}_i$, denoted as $\lambda_{\min}(\boldsymbol{D}_i)$, is zero.*

A dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times d}$ is said to be ill-conditioned under a given sparsity pattern Ω if there exists an $i \in [n]$ such that the condition number of the sub-dictionary $\boldsymbol{D}_i$ is large, or equivalently $\lambda_{\min}(\boldsymbol{D}_i)$ is close to zero.

Definition 2 [16] *Define the condition number of a dictionary $\boldsymbol{D}$* as:

$$\kappa(\boldsymbol{D}) = \max_{i \in [n]} \frac{\lambda_{\max}(\boldsymbol{D}_i)}{\lambda_{\min}(\boldsymbol{D}_i)},$$

where $\lambda_{\max}\left(\boldsymbol{D}_i\right)$ *and* $\lambda_{\min}\left(\boldsymbol{D}_i\right)$ *represent the maximum and the minimum singular value of the sub-dictionary* $\mathbf{D}_i$ *respectively.*

The word "singular" comes from the fact that $f\left(\boldsymbol{D}\right) = \min\limits_{\boldsymbol{X}\in\mathcal{X}_{\Omega}} \left\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\right\|_F^2$ is not continuous at a singular dictionary[2] and the corresponding

$$\boldsymbol{X}\left(\boldsymbol{D}\right) \triangleq \arg\min_{\boldsymbol{X}\in\mathcal{X}_{\Omega}} \left\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\right\|_F^2$$

is not unique. The singularity of $f\left(\boldsymbol{D}\right)$ leads to convergence problems. Benchmark dictionary update procedures may fail to find a globally optimal solution. Instead they converge to a singular point of $f\left(\boldsymbol{D}\right)$, i.e., a singular dictionary.

Ill-conditioned dictionaries are in the neighborhood of singular ones. Algorithmically when one of the $\lambda_{\min}\left(\boldsymbol{D}_i\right)$s is ill-conditioned, the curvature of $f\left(\boldsymbol{D}\right)$ is quite large and the value of the gradient fluctuates dramatically. This seriously affects the convergence rate of the dictionary update process.

Furthermore, ill-conditioned dictionaries also bring negative effect on the sparse coding stage. Denote $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ as the ith column of $\boldsymbol{Y}$ and $\boldsymbol{X}$ respectively. Consider a summand of the formulation in sparse coding stage [16, 26], i.e.,

$$\min_{\boldsymbol{x}_i} \left\|\boldsymbol{y}_i-\boldsymbol{D}\boldsymbol{x}_i\right\|_F^2 + \left\|\boldsymbol{x}_i\right\|_0 .$$

An ill-conditioned $\boldsymbol{D}$ corresponds to a very large condition number, which breaks the restricted isometry condition (RIP) [34], and results in the unstable solutions: with small perturbations added on the training sample $\boldsymbol{Y}$, the solutions of $\boldsymbol{X}$ deviate significantly.

2.3.3 Regularized SimCO

The main idea of Regularized SimCO lies in the use of an additive penalty term to avoid singularity. Consider the objective function $f_{\mu}\left(\tilde{\boldsymbol{D}}\right)$ in (2.7),

$$\begin{aligned} f_{\mu}\left(\tilde{\boldsymbol{D}}\right) &= \min_{\boldsymbol{X}\in\mathcal{X}_{\Omega}} \left\|\boldsymbol{D}\boldsymbol{X}-\mathcal{P}\left(\boldsymbol{S}\right)\right\|_F^2 + \mu\left\|\boldsymbol{X}\right\|_F^2 , \\ &= \min_{\boldsymbol{X}\in\mathcal{X}_{\Omega}} \left\| \begin{bmatrix} \mathcal{P}\left(\boldsymbol{S}\right) \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{D} \\ \sqrt{\mu}\boldsymbol{I} \end{bmatrix} \boldsymbol{X} \right\|_F^2 . \end{aligned} \tag{2.15}$$

As long as $\mu \neq 0$ ($\mu > 0$ in our case), the block $\mu\boldsymbol{I}$ guarantees the full column rank of $\tilde{\boldsymbol{D}} = \left[\boldsymbol{D}^T\ \mu\boldsymbol{I}\right]^T$. Therefore, with the modified objective function $f_{\mu}\left(\tilde{\boldsymbol{D}}\right)$, there is

[2] An illustration: take $\boldsymbol{Y}$, $\boldsymbol{D}$, $\boldsymbol{X}$ as scalars. If $\boldsymbol{Y} \neq 0$, there exists a singular point at $\boldsymbol{D} = 0$ on $f\left(\boldsymbol{D}\right) = \min\limits_{\boldsymbol{X}} \left\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\right\|_F^2$, where $\boldsymbol{X}$ can be assigned as any real number.

no singular point so that gradient descent methods will only converge to stationary points.

This regularization technique is also applicable to MOD [16]. It is verified that this technique effectively mitigates the occurrence of ill-conditioned dictionary although at the same time some stationary points might be generated. To alleviate this problem, one can decrease gradually the regularization parameter μ during the optimization process [16]. In the end μ will decrease to zero. Nevertheless, it is still not guaranteed to converge to a global minimum. The explicit example constructed in [32] shows a failure of the Regularized SimCO. As a result, another method to address the singularity issue is introduced below.

2.3.4 Smoothed SimCO

Also aiming at handling the singularity issue, Smoothed SimCO [33] is to remove the singularity effect by adding multiplicative functions. The intuition is explained as follows. Write $f\left(\boldsymbol{D}\right)$ into a summation of atomic functions

$$\begin{aligned} f(\boldsymbol{D}) &= \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2 \\ &= \sum_i \|\boldsymbol{Y}_{:,i}-\boldsymbol{D}_i\boldsymbol{X}_{\boldsymbol{\Omega}(:,i),i}\|_2^2 \\ &= \sum_i f_i(\boldsymbol{D}_i), \end{aligned} \tag{2.16}$$

where each $f_i(\boldsymbol{D}_i)$ is termed as an atomic function and $\boldsymbol{D}_i$ is defined in Definition 1. Let $\mathcal{I}$ be the index set corresponding to the $\boldsymbol{D}_i$'s of full column rank. Define an indicator function $\mathcal{X}_{\mathcal{I}}$ s.t. $\mathcal{X}_{\mathcal{I}}\left(i\right) = 1$ if $i \in \mathcal{I}$ and $\mathcal{X}_{\mathcal{I}}\left(i\right) = 0$ if $i \in \mathcal{I}^c$. Use $\mathcal{X}_{\mathcal{I}}\left(i\right)$ as a multiplicative modulation function and apply it to each $f_i\left(\boldsymbol{D}_i\right)$. Then one obtains

$$\bar{f}(\boldsymbol{D}) = \sum_i f_i(\boldsymbol{D}_i)\mathcal{X}_{\mathcal{I}}\left(i\right) = \sum_{i\in\mathcal{I}} f_i(\boldsymbol{D}_i). \tag{2.17}$$

This new function $\bar{f}$ is actually the best possible lower semi-continuous approximation of f and there is no new stationary point created.

Motivated from the above, we define

$$\tilde{f}(\boldsymbol{D}) = \sum_i f_i(\boldsymbol{D}_i) g\left(\lambda_{\min}\left(\boldsymbol{D}_i\right)\right), \tag{2.18}$$

where the shape of g is given in Fig. 2.1. The function g has the following properties: (1) $g\left(\lambda_{\min}\right) = 0$ for all $\lambda_{\min} \leq 0$; (2) $g\left(\lambda_{\min}\right) = 1$ for all $\lambda_{\min}\left(\boldsymbol{D}_i\right) > \delta > 0$, where δ is a threshold; (3) g is monotonically increasing; (4) g is second order differentiable. When using $\lambda_{\min}\left(\boldsymbol{D}_i\right)$ as the input variable for g and the positive threshold $\delta \to 0$,

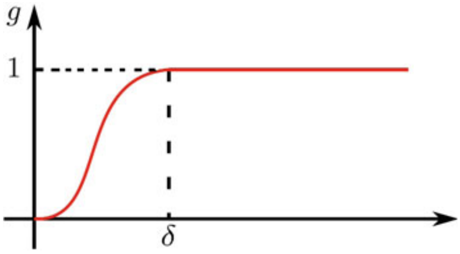

Fig. 2.1 A shape of function $g(\cdot)$

$\lambda_{\min}(\boldsymbol{D}_i)$ becomes an indicator function indicating whether $\boldsymbol{D}_i$ has a full column rank, i.e.,

$$\begin{cases} g(\lambda_{\min}(\boldsymbol{D}_i)) = 1 & \text{if } \boldsymbol{D}_i \text{ has full column rank;} \\ g(\lambda_{\min}(\boldsymbol{D}_i)) = 0 & \text{otherwise.} \end{cases}$$

The modulated objective function $\tilde{f}$ has several good properties, which do not exhibit in the regularized objective function (2.15). In particular, we have the following theorems.

Theorem 1 *Consider the smoothed objective function $\tilde{f}$ and the original objective function f defined in* (2.18) *and* (2.16) *,respectively.*

1. *When $\delta > 0, \forall i, \tilde{f}(\boldsymbol{D})$ is continuous.*
2. *Consider the limit case where $\delta \to 0$ with $\delta > 0, \forall i$. The following statements hold:*
   a. *$\tilde{f}(\boldsymbol{D})$ and $f(\boldsymbol{D})$ differ only at the singular points.*
   b. *$\tilde{f}(\boldsymbol{D})$ is the best possible lower semi-continuous approximation of $f(\boldsymbol{D})$.*

Theorem 2 *Consider the smoothed objective function $\tilde{f}$ and the original objective function f defined in* (2.18) *and* (2.16) *,respectively. For any $a \in \mathbb{R}$, define the lower level set $\mathcal{D}_f(a) = \{\boldsymbol{D} : f(\boldsymbol{D}) \leq a\}$. It is provable that when $\delta \to 0$, $\mathcal{D}_{\tilde{f}}(a)$ is the closure of $\mathcal{D}_f(a)$.*

In practice, we always choose a $\delta > 0$. The effect of a positive δ, roughly speaking, is to remove the barriers created by singular points, and replace them with “tunnels”, whose widths are controlled by δ, to allow the optimization process to pass through. The smaller the δ is, the better $\tilde{f}$ approximates f, but the narrower the tunnels are, and the slower the convergence rate will be. As a result, the threshold δ should be properly chosen. A detailed discussion of choosing δ is presented in [32]. Compared with the choice of the parameter (μ) in the Regularized SimCO [16], the choice of the smoothing threshold δ is easier: one can simply choose a small $\delta > 0$ without decreasing it during the process.

As final remarks, Smoothed SimCO has several theoretical advantages over Regularized SimCO. However, the computations of $(\lambda_{\min}(\boldsymbol{D}_i))$'s introduce extra cost. The choice between these two methods will depend on the size of the problem under consideration.

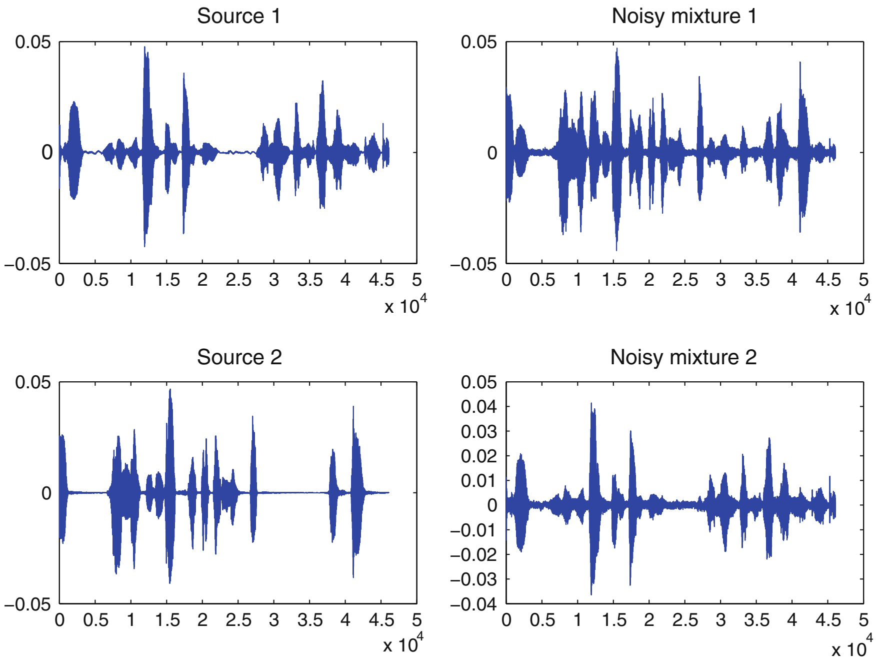

Fig. 2.2 Two speech sources and the corresponding noisy mixtures (20 dB Gaussian noise)

2.4 Algorithm Testing on Practical Applications

In this section, we present numerical results of the SparseBSS method compared with some other mainstream algorithms. We first focus on speech separation where an equal determined case will be considered. Then, we show an example for blind image separation, where we will consider an overdetermined case.

In the speech separation case two mixtures are used, which are the mixtures of two audio sources. Two male utterances in different languages are selected as the sources. The sources are mixed by a 2×2 random matrix $\boldsymbol{A}$ (with normalized columns). For the noisy case, a 20 dB Gaussian noise was added to the mixtures. See Fig. 2.2 for the sources and mixtures.

We compare SparseBSS with two benchmark algorithms including FastICA and QJADE [35]. The BSSEVAL toolbox [36] is used for the performance measurement. In particular, an estimated source $\hat{\boldsymbol{s}}$ is decomposed as $\hat{\boldsymbol{s}} = \boldsymbol{s}_{\text{target}} + \boldsymbol{e}_{\text{interf}} + \boldsymbol{e}_{\text{noise}} + \boldsymbol{e}_{\text{artif}}$, where $\boldsymbol{s}_{\text{target}}$ is the true source signal, $\boldsymbol{e}_{\text{interf}}$ denotes the interferences from other sources, $\boldsymbol{e}_{\text{noise}}$ represents the deformation caused by the noise, and $\boldsymbol{e}_{\text{artif}}$ includes all other artifacts introduced by the separation algorithm. Based on the decomposition, three performance criteria can be defined: the source-to-distortion ratio $\text{SDR} = 10 \log_{10} \frac{\|\boldsymbol{s}_{\text{target}}\|^2}{\|\boldsymbol{e}_{\text{interf}} + \boldsymbol{e}_{\text{noise}} + \boldsymbol{e}_{\text{artif}}\|^2}$, the source-to-artifact ratio

Table 2.1 Separation performance of the SparseBSS algorithm as compared to FastICA and QJADE

	△SDR	△SIR	△SAR
(a) The noiseless case			
QJADE	60.661	60.661	−1.560
FastICA	57.318	57.318	−0.272
SparseBSS	**69.835**	**69.835**	**1.379**
(b) The noisy case			
QJADE	7.453	58.324	−1.245
FastICA	7.138	40.789	−1.552
SparseBSS	**9.039**	**62.450**	**0.341**

The proposed SparseBSS algorithm performs better than the benchmark algorithms. Table 2.1a. For the same algorithm, the △*SDR* and △*SIR* are the same in noiseless case. The △*SDRs* and △*SIRs* for all the tested algorithms are large and similar, suggesting that all the compared algorithms perform very well. The artifact introduced by SparseBSS is small as its △*SAR* is positive. Table 2.1b. In the presence of noise with SNR = 20 dB, SparseBSS excels the other algorithms in △*SDR*, △*SIR*, and △*SAR*. One interesting phenomenon is that the △*SDRs* are much smaller than those in the noiseless case, implying that the distortion introduced by the noise is trivial. However, SparseBSS still has better performance

$\mathrm{SAR} = 10\log_{10}\frac{\|\boldsymbol{s}_{\text{target}}+\boldsymbol{e}_{\text{interf}}+\boldsymbol{e}_{\text{noise}}\|^2}{\|\boldsymbol{e}_{\text{artif}}\|^2}$, and the source-to-interference ratio $\mathrm{SIR} = 10\log_{10}\frac{\|\boldsymbol{s}_{\text{target}}\|^2}{\|\boldsymbol{e}_{\text{interf}}\|^2}$. Among them, the SDR measures the overall performance (quality) of the algorithm, and the SIR focuses on the interference rejection. We investigate the gains of SDRs, SARs, and SIRs from the mixtures to the estimated sources. For example, $\triangle SDR = SDR_{out} - SDR_{in}$, where SDR_{out} is calculated from its definition and SDR_{in} is obtained by letting $\hat{\boldsymbol{s}} = \boldsymbol{Z}$ with the same equation. The results (in dB) are summarized in Table 2.1.

The selection of λ is an important practical issue since it is related to the noise level and largely affects the algorithm performance. From the optimization formulation (2.3), it is clear that with a fixed SNR, different choices of λ may give different separation performance. To show this, we use the estimation error $\left\|\boldsymbol{A}_{\text{true}} - \hat{\boldsymbol{A}}\right\|_F^2$ of the mixing matrix to measure the separation performance, where $\boldsymbol{A}_{\text{true}}$ and $\hat{\boldsymbol{A}}$ are the true and estimated mixing matrices, respectively. The simulation results are presented in Fig. 2.3. Consistent with the intuition, simulations suggest that the smaller the noise level, the larger the optimal value of λ. The results in Fig. 2.3 help in setting λ when the noise level is known a priori.

Next, we show an example for blind image separation, where we consider an overdetermined case. The mixed images are generated from two source images using a 4×2 full rank column normalized mixing matrix $\boldsymbol{A}$ with its elements generated randomly according to a Gaussian process. The mean squared errors (MSEs) are used to compare the reconstruction performance of the candidate algorithms when no noise is added. MSE is defined as $MSE = (1/N)\,\|\chi - \tilde{\chi}\|_F^2$, where χ is the source image and $\tilde{\chi}$ is the reconstructed image. The lower the MSE, the better the reconstruction

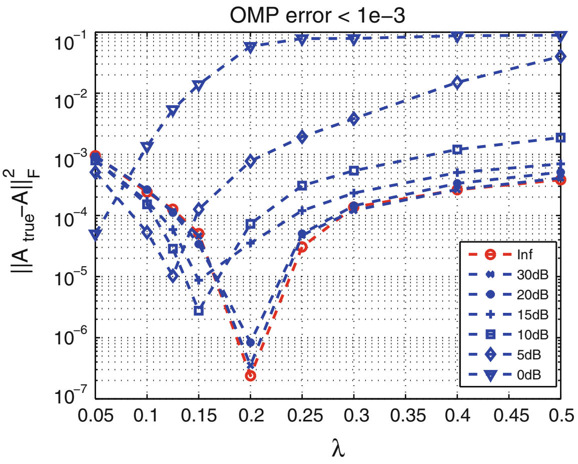

Fig. 2.3 Relation of the parameter λ to the estimation error of the mixing matrix under different noise levels. The signal-to-noise ratio (SNR) is defined as $\rho = 10\log_{10} \|\boldsymbol{AS}\|_F^2 \,/\, \|\boldsymbol{V}\|_F^2$ dB

Table 2.2 Achieved MSEs of the algorithms in a noiseless case

	FastICA	GMCA	BMMCA	SparseBSS
Lena	8.7489	4.3780	3.2631	**3.1346**
Boat	18.9269	**6.3662**	12.5973	6.6555

performance. Table 2.2 illustrates the results of four tested algorithms. For the noisy case, a Gaussian white noise is added to the four mixtures with $\sigma = 10$. We use the Peak Signal-to-Noise Ratio (PSNR) to measure the reconstruction quality, which is defined as, $PSNR = 20\log_{10}(MAX/\sqrt{MSE})$, where MAX indicates the maximum possible pixel value of the image, (e.g., $MAX = 255$ for a uint-8 image). Higher PSNR indicates better quality. The noisy observations are illustrated in Fig. 2.4b.[3]

Finally, we show another example of blind image separation to demonstrate the importance of the singularity-aware process. In this example, we use two classic images *Lena* and *Texture* as the source images (Fig. 2.6a). Four noiseless mixtures were generated from the sources. The separation results are shown in Fig. 2.6b and c. Note that images like *Texture* contain a lot of frequency components corresponding to a particular frequency. Hence, an initial dictionary with more codewords corresponding to the particular frequency may perform better for the estimation of these images. Motivated by this, in Fig. 2.6b the initial dictionary is generated from an overcomplete DCT dictionary, but contains more high frequency codewords. Such choice

[3] For the BMMCA test, a better performance was demonstrated in [14]. We point out that here a different true mixing matrix is used. And furthermore, in our tests the patches are taken with a 50 % overlap (by shifting 4 pixels from the current patch to the next) while in [14] the patches are taken by shifting only one pixel from the current patch to the next.

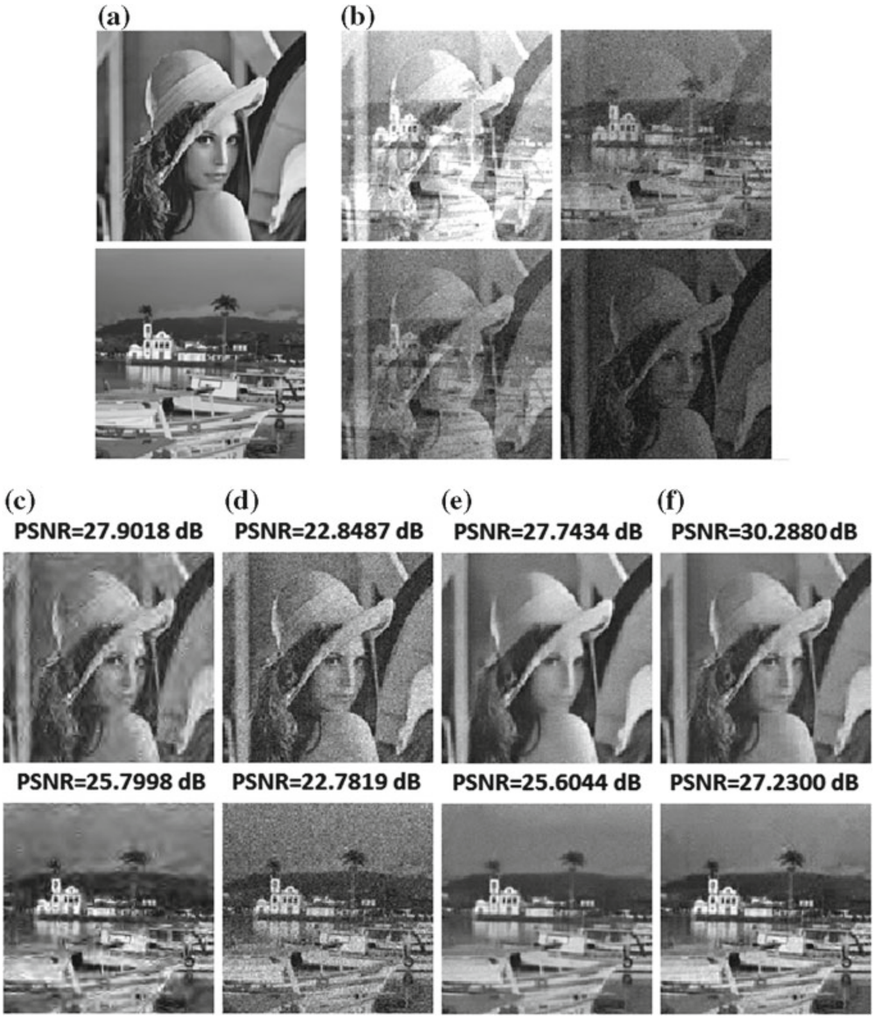

Fig. 2.4 Two classic images, *Lena* and *Boat* were selected as the source images, which are shown in (**a**). The mixtures are shown in (**b**). The separation results are shown in (**c**–**f**). We compared SparseBSS with other benchmark algorithms: FastICA [37], GMCA [10], and BMMCA [14]. We set the overlap percentage equal to 50 % for both BMMCA and SparseBSS. The recovered source images by the SparseBSS tend to be less blurred compared to the other three algorithms

can lead to better separation results. At the same time, the very similar dictionary codewords may introduce the risk of singularity issue (Fig. 2.5).

The major difference between Fig. 2.6b and c is that: in Fig. 2.6b the Regularized SimCO process ($\mu = 0.05$) is introduced, while in Fig. 2.6c there is no regularization term in the dictionary learning stage. As one can see from the numerical results, Fig. 2.6b performs much better than Fig. 2.6c. By checking the condition number

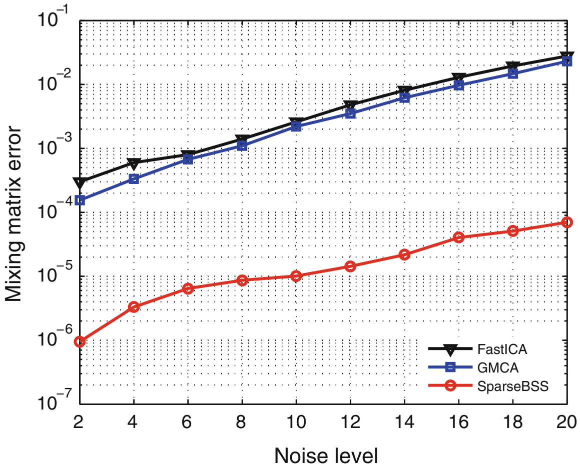

Fig. 2.5 Compare the performance of estimating the mixing matrix for all the methods in different noise standard deviation σs. In this experiment, σ varies from 2 to 20. The performance of GMCA is better than that of FastICA. The curve for BMMCA is not available as the setting for the parameters is too sophisticated and inconsistent for different σ to obtain a good result. SparseBSS outperforms the compared algorithms

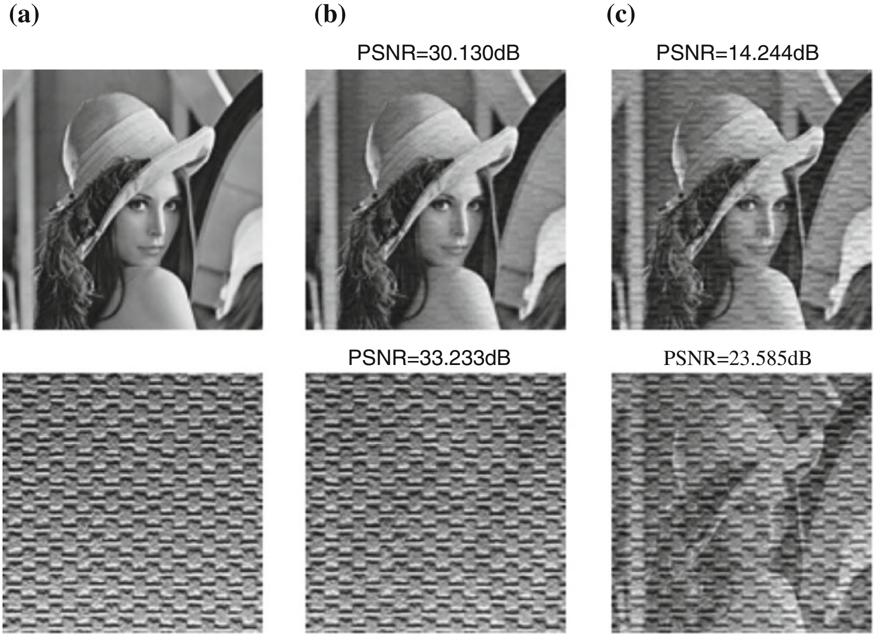

Fig. 2.6 The two source images *Lena* and *Texture* are shown in (**a**). The separation results are shown in (**b**) and (**c**). The comparison results demonstrate the importance of the singularity -aware process

when the regularized term is not introduced ($\mu = 0$), the value stays at a high level as expected (larger than 40 in this example). This confirms the necessity of considering the singularity issue in BSS and the effectiveness of the proposed singularity-aware approach.

2.5 Conclusions and Prospective Extensions

In conclusion, we briefly introduced a development of the blind source separation algorithms based on dictionary learning. In particular, we focus on the SparseBSS algorithm and the optimization procedures. The singularity issue might lead to the failure of these algorithms. At the same time there are still some open questions to be addressed.

In dictionary learning, it remains open how to find an optimum choice of the redundancy factor $\tau = d/n$ of the over-complete dictionary. A higher redundancy factor leads to either more sparse representation or more precise reconstruction. Moreover, one has to consider the computational capabilities when implementing the algorithms. From this point of view, it is better to keep the redundancy factor low. In the simulation, we have used a 64 by 256 dictionary, which gives the redundancy factor $\tau = 256/64 = 4$. This choice is empirical: the sparse representation results are good and the computational cost is limited. A rigorous analysis on the selection of τ is still missing.

The relation between the parameters λ, ϵ, and noise standard deviation σ is also worth investigating. As presented in the first experiment on blind audio separation, the relation between λ and σ is discussed when the error bound ϵ is fixed in the sparse coding stage. One can roughly estimate the value of the parameter λ assuming the noise level is known a priori. Similar investigation is undertaken in [14], where the authors claim that when $\lambda \approx \sigma/30$, the algorithm achieved similar reconstruction performance under various σ's. From another perspective, the error bound ϵ is proportional to the noise standard deviation. It turns out that once a well-approximated relation between ϵ and σ is obtained, one may get more precise estimation of parameter λ, rather than keeping ϵ fixed. This analysis, therefore, is counted as another open question.

References

1. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, New York (2001)
2. Bell, J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Comput. **7**, 1129–1159 (1995)
3. Gaeta, M., Lacoume, J.L.: Source separaion without prior knowledge: the maximum likelihood solution. In: Proceedings of European Signal Processing Conference, pp. 621–624 (1990)

4. Belouchrani, A., Cardoso, J.F.: Maximum likelihood source separation for discrete sources. In: Proceedings of European Signal Processing Conference, pp. 768–771 (1994)
5. Bronstein, M., Zibulevsky, M., Zeevi, Y.: Sparse ica for blind separation of transmitted and reflected images. Int. J. Imaging Sci. Technol. **15**, 84–91 (2005)
6. Gribonval, R., Lesage, S.: A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In: Proceedings of European Symposium on Artificial, Neural Networks, pp. 323–330 (2006)
7. Jourjine, A., Rickard, S., Yilmaz, O.: Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2985–2988 (2000)
8. Starck, J., Elad, M., Donoho, D.: Redundant multiscale transforms and their application for morphological component analysis. Adv. Imaging Electron. Phys. **132**, 287–348 (2004)
9. Bobin, J., Moudden, Y., Starck, J., Elad, M.: Morphological diversity and source separation. IEEE Sign. Process. Lett. **13**(7), 409–412 (2006)
10. Bobin, J., Starck, J., Fadili, J., Moudden, Y.: Sparsity and morphological diversity in blind source separation. IEEE Trans. Image Process. **16**(11), 2662–2674 (2007)
11. Eladl, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, 1st edn. Springer, New York (2010). Incorporated
12. Peyré, G., Fadili, J., Starck, J.-L.: Learning adapted dictionaries for geometry and texture separation. In: Proceedings of SPIE Wavelet XII, vol. 6701, p. 67011T (2007)
13. Xu, T., Wang, W., Dai, W.: Sparse coding with adaptive dictionary learning for underdetermined blind speech separation. Speech Commun. **55**(3), 432–450 (2013)
14. Abolghasemi, V., Ferdowsi, S., Sanei, S.: Blind separation of image sources via adaptive dictionary learning. IEEE Trans. Image Process. **21**(6), 2921–2930 (2012)
15. Zhao, X., Xu, T., Zhou, G., Wang, W., Dai, W.: Joint image separation and dictionary learning. In: Accepted by 18th International Conference on Digital Signal Processing, Santorini, Greece (2013)
16. Dai, W., Xu, T., Wang, W.: Simultaneous codeword optimization (simco) for dictionary update and learning. IEEE Trans. Sign. Process. **60**(12), 6340–6353 (2012)
17. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, pp. 40–44 (1993)
18. Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans. Inf. Theory **55**(5), 2230–2249 (2009)
19. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
20. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**, 303–353 (1999)
21. Hildebrand, F.B.: Advanced Calculus for Applications. Prentice-Hall, Upper Saddle River (1976)
22. Engan, K., Aase, S.O., Husoy, J.H.: Method of optimal directions for frame design. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2443–2446 (1999)
23. Gorodnitsky, I.F., George, J.S., Rao, B.D.: Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm. Electroencephalogr. Clin. Neurophysiol. **95**, 231–251 (1995)
24. Engan, K., Skretting, K., Husoy, J.: Family of iterative is-based dictionary learning algorithms, ils-dla, for sparse signal representation. Digit. Sign. Process. **17**(1), 32–49 (2007)
25. Skretting, K., Engan, K.: Recursive least squares dictionary learning algorithm. IEEE Trans. Sign. Process. **58**(4), 2121–2130 (2010)
26. Aharon, M., Elad, M., Brucketein, A.: K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Sign. Process. **54**(11), 4311–4322 (2006)
27. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res. **11**, 19–60 (2010)
28. Gribonval, R., Schnass, K.: Dictionary identification: sparse matrix-factorisation via l1-minimisation. CoRR, vol. abs/0904.4774 (2009)

29. Geng, Q., Wang, H., Wright, J.: On the local correctness of l1 minimization for dictionary learning. CoRR, vol. abs/1101.5672 (2011)
30. Jenatton, R., Gribonval, R., Bach, F.: Local stability and robustness of sparse dictionary learning in the presence of noise. CoRR (2012)
31. Yaghoobi, M., Blumensath, T., Davies, M.E.: Dictionary learning for sparse approximations with the majorization method. IEEE Trans. Sign. Process. **57**(6), 2178–2191 (2009)
32. Zhao, X., Zhou, G., Dai, W.: Dictionary learning: a singularity problem and how to handle it (in preparation)
33. Zhao, X., Zhou, G., Dai, W.: Smoothed SimCO for dictionary learning: handling the singularity issue. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (2013)
34. Candes, E., Tao, T.: Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
35. Cardoso, J.F., Souloumiac, A.: Blind beamforming for non-Gaussian signals. IEE Proc. **140**(6), 362–370 (1993)
36. Vincent, E., Gribonval, R., Fevotte, C.: Performance measurement in blind audio source separation. IEEE Trans. Audio Speech Lang. Process. **14**(4), 1462–1469 (2006)
37. Hyvarinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Networks **10**(3), 626–634 (1999)

http://www.springer.com/978-3-642-55015-7

Blind Source Separation
Advances in Theory, Algorithms and Applications
Naik, G.R.; Wang, W. (Eds.)
2014, IX, 551 p. 189 illus., Hardcover
ISBN: 978-3-642-55015-7