ORIGINAL ARTICLE

Blind source separation for the analysis sparse model

Shuang Ma[1] · Hongjuan Zhang[1] · Zhuoyun Miao[1]

Abstract

Sparsity of the signal has been shown to be very useful for blind source separation (BSS) problem which aims at recovering unknown sources from their mixtures. In this paper, we propose a novel algorithm based on the analysis sparse constraint of the source over an adaptive analysis dictionary to address BSS problem. This method has an alternating scheme by keeping all but one unknown fixed at a time so that the dictionary, the source, and the mixing matrix are estimated alternatively. In order to make better use of the sparsity constrain, l_0-norm is utilized directly for a more exact solution instead of its other relaxation, such as l_p-norm ($0 < p \leq 1$). Numerical experiments show that the proposed method indeed improves the separation performance.

Keywords Analysis sparse model · Analysis dictionary learning · Blind source separation

1 Introduction

In signal and image processing, we frequently need to address the problem of recovering unknown sources from their mixtures, that is, the blind source separation (BSS) problem, which can be described as the following linear mixing model

$$Y = \mathrm{AX} + V \tag{1}$$

where $Y = [y_1, y_2, \ldots, y_m]^T \in \mathbb{R}^{m \times N}$ and $X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n \times N}$ denote the observed matrix and the source matrix, respectively, $A \in \mathbb{R}^{m \times n}$ is the mixing matrix, and $V \in \mathbb{R}^{m \times N}$ is referred to the additive Gaussian noise. Notice that X and Y contain n sources and m mixtures, respectively, and here N is the data size. BSS's task is to estimate the sources X (and the mixing matrix A) without any prior knowledge except the observation Y. As we all know that this problem does not have a unique solution in general. Therefore, it is necessary to impose some restrictions into the separation process and make the sources distinguishable.

✉ Hongjuan Zhang
  zhanghongjuan@shu.edu.cn

1   Department of Mathematics, Shanghai University,
    Shanghai 200444, People's Republic of China

Independent component analysis (ICA) [1], as one of classical approaches to solve the over-determined BSS problem ($m \geq n$), assumes that the sources are non-Gaussian and as statistically independent as possible. It has led to many well-known approaches, such as Infomax [2], maximum likelihood estimation [3], the maximum a posterior (MAP) [4] and FastICA [1]. Where Infomax using a stochastic gradient algorithm to maximize the likelihood may require some hand-tuning and often fails to converge [5], or only converges slowly. Therefore, various methods have been proposed to maximize Infomax likelihood much faster by using curvature information, especially the complete set of second-order derivatives (Hessian). However, as a large object, Hessian will cause high computational cost for large data sets. To address the cost issue, [6] adopted a truncated Newton algorithm, and another approach is to use approximations of the Hessian, such as a simple Quasi-Newton method in [7] and [8], and a trust-region method in [9]. Nevertheless, the work in [10] compared these methods [7–9] and found that the Hessian approximations do yield a low cost per iteration, but that they are not accurate enough on real data. Therefore, literature [10] proposed a Preconditioned ICA for Real Data (Picard) algorithm, which is a relative L-BFGS algorithm preconditioned with sparse Hessian approximations, to overcome this problem. However, the classical ICA algorithm is weak in processing noisy sources separation.

Meanwhile, Lee and Seung proposed another effective method, nonnegative matrix factorization (NMF) by taking advantages of the nonnegativity constraints. These constraints lead to aparts-based representation because they allow only additive, not subtractive, combinations [11]. Afterward, Boutsidis and Gallopoulos described nonnegative double singular value decomposition (NNDSVD) [12], a new method designed to enhance the initialization stage of NMF, which is well suited to initialize NMF algorithms with sparse factors and many numerical examples showed that NNDSVD made the approximation error of many NMF algorithms decrease rapidly.

During the past decade, sparsity has been widely used in many aspects [13–18]. Vast works have been in an attempt to use sparsity for BSS problem, and many sparsity-based approaches have proposed [19–24]. They are collectively coined sparse component analysis (SCA), which means that the sources can be sparsely represented using a known common basis or dictionary. For instance, morphological component analysis (MCA) [19] is a novel decomposition method based on sparse representation of signals with a signal dictionary, which showed that different dictionaries produce entirely different representations and the sparser representation, the better separation. Therefore, Jérôme Bobin described an extension of the MCA-multichannel morphological component analysis (MMCA) [20] adapted to the multichannel data. MMCA took advantage of a large over-complete dictionary to obtain different sparse representations for n sources based on their morphological diversity. And in literature [21], generalized morphological component analysis (GMCA) was proposed to the much more general case and considered that each source is a linear combination of several components and each component is sparse in a given dictionary. Moreover, Vahid Abolghasemi proposed BMMCA [22], which can adaptively learn the dictionaries from the mixed images via K-SVD, adapting MMCA to those cases that the known dictionaries are not available. Further, the work in [24] extended the BMMCA method that uses the only one dictionary to sparsely represent different sources. And it inventively incorporated SimCO optimization framework [25] into two-stage dictionary learning and mixture learning, thus significantly reducing the implementation effort.

It is noting that the above sparsity-related methods without exception are all relied on synthesis model. In this model, a signal $w \in \mathbb{R}^d$ can be composed as $w = D\alpha$, where $D \in \mathbb{R}^{d \times k}$ $(d \geq k)$ is a dictionary and the representation coefficient $\alpha \in \mathbb{R}^k$ is expected to be sparse, i.e., $\|\alpha\|_0 = c \ll k$. Just like above mentioned, the synthesis model has been extensively applied to BSS problem. As for its dual analysis angle—analysis sparse model—in sparse representation has been barely touched [26, 27]. The

analysis model is described as $z = \Omega w$, where the $\Omega \in \mathbb{R}^{p \times d}$ $(p \geq d)$ refers to a analysis dictionary. The implications of this model are the signal $w \in \mathbb{R}^d$ can be transformed into a low-dimensional subspace via a linear operation Ω. Therefore, the representation coefficient z is assumed as sparse, i.e., $\|z\|_0 = c \ll p$.

Although the two models have very similar structure, the theoretical as well as simulation results provided in [28] have indicated that there exists a respectable gap between them, especially in the over-complete case. Moreover, in some respects, analysis model has advantages over synthesis model. For example, analysis model contains a larger number of low-dimensional subspaces in the case of the same dimension $(k = d - l)$, which enrich in terms of its descriptive power [26]. In addition, in the framework of synthesis model where only a small number of atoms are used to represent each signal, the significance of every atom grows enormously; any wrong choice could potentially lead to a 'domino effect.' But in the analysis formulation, all atoms take an equal part in describing the signal, thus minimizing the dependence on each individual one, and stabilizing the recovery process [28]. More generally, analysis-based methods are a very common structure in image processing and computer vision applications. Due to the smaller dimension of the unknown, compared with the similar-sized synthesis-based method, the analysis-based method leads to a more simple optimization problem in the over-complete case $(p > d)$, which is considerably easier to be solved [28]. Therefore, fusing analysis sparse constraint to the processing of BSS problem becomes necessary.

Recently, one of the effective related attempts was the work of literature [29], in which experimental results have demonstrated that the approach based on analysis sparse model has better separation performance. Nevertheless, it should be noticed that this work replaced the l_0-norm with l_1-norm as the sparsity measure to obtain a approximate solution, which will certainly reduce sparsity. l_0-norm is the most intuitive measure of sparsity. Only when certain conditions, such as exact recovery condition (ERC) and restricted isometry property (RIP) [26] are met, can l_1-norm be considered equivalently to l_0-norm.

Hence, in this paper, we propose a new method, termed analysis BSS (ABSS), to make better use of the sparsity constrain. Different from the most BSS algorithms based on synthesis model, this method bases on analysis model. And it directly uses l_0-norm as the measure of sparsity to obtain a more exact solution. In turn we come up with a novel objective function for the BSS problem based on analysis sparse model. In order to simplify this problem, the solution process is divided into three stages: dictionary learning stage, sources estimating stage and mixing matrix

estimating stage. In the first stage, we adopt Analysis KSVD [30] to adaptively estimate the analysis dictionary from mixed sources. Next in the second and third stage, the sources and mixing matrix can be uniquely obtained separately by a simple least square linear regression. Finally, the results indicate that using l_0-norm enhances the separability of the sources.

The organization of this manuscript is as follows. In sect. 2, we are devoted to introducing the proposed framework. Then the details of our algorithm are presented in Sect. 3. Finally, the simulations compared with some typical methods to verify the competitive separation performances of the proposed algorithm are provided in Sect. 4.

2 Problem formulation

Firstly, BSS problem (1) can be written as the following optimization problem

$$\min_{A,X} \|Y - AX\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Notice that in this problem, the mixing matrix $A \in \mathbb{R}^{m \times n}$ and the source matrix $X : \{x_i\}_{i=1}^n \in \mathbb{R}^{n \times N}$ are completely unknown. We can only use the known observed matrix $Y : \{y_i\}_{i=1}^m \in \mathbb{R}^{m \times N}$ which is the input of our algorithm to estimate X and A. Therefore, this problem is ill-posed and here we will impose the analysis sparse constraint on it, that is, the forward projection of the source x_i is sparse enough on the basis elements. The sparse representation of x_i based on analysis model can be described as

$$\min_{\Omega_i, s_i} \|s_i\|_0 \quad s.t. \; s_i = \Omega_i x_i \tag{3}$$

where the s_i is referred to the representation coefficient of x_i, which most entries are zero, and the $\|\cdot\|_0$ is the ℓ_0-norm which counts the number of nonzeros. In particular, $\Omega_i \in \mathbb{R}^{p \times d}$ is an analysis dictionary corresponding to the ith source. Optimization problem (3) can be written in its Lagrangian form

$$\min_{\Omega_i, s_i} \|s_i - \Omega_i x_i\|_F^2 + \mu \|s_i\|_0 \tag{4}$$

Note that formula (4) describes the analysis model, which is different from the expression of the synthesis model in BMMCA [22]. In addition, l_0-norm is directly used as the measure of sparsity, which is different from the method proposed by Fang [29]. Overall, the BSS problem based on the analysis sparse model can be described as a joint optimization problem

$$\min_{A,\Omega_i, x_i, s_i} \lambda \|Y - AX\|_F^2 + \sum_i \mu \|s_i\|_0 + \sum_i \|s_i - \Omega_i x_i\|_F^2 \tag{5}$$

where scalars λ and μ control the noise power and sparsity degree, respectively. To simplify the problem, the dictionary, the sources and the mixing matrix are estimated alternatively. So overall optimization problem (5) is divided into the following stage

– Dictionary learning stage (with X and A fixed)

$$\min_{\Omega_i, s_i} \|s_i\|_0 \quad s.t. \; \|s_i - \Omega_i x_i\|_F^2 \le \delta \qquad i = 1, \ldots, n \tag{6}$$

– Sources estimating stage (with Ω_i, s_i, and A fixed)

$$\min_X \lambda \|Y - AX\|_F^2 + \sum_i \|s_i - \Omega_i x_i\|_F^2 \tag{7}$$

– Mixing matrix estimating stage (with X, Ω_i, and s_i fixed)

$$\min_A \|Y - AX\|_F^2 \tag{8}$$

In the overall separation process, the above three optimization subproblems should be addressed iteratively until convergence. The specific details of the proposed algorithm, coined Analysis BSS (ABSS), are described in Sect. 3.

3 ABSS algorithm

Before giving the ABSS algorithm, we make some conventions on the variables in the model in advance, which will be more conducive to the proposed method's application, such as solving large images separation problems. Here, the jth $\sqrt{N} \times \sqrt{N}$ observation image $\mathcal{Y}_j$ is presented as the vector $y_j \in \mathbb{R}^{N \times 1}$ and the ith $\sqrt{N} \times \sqrt{N}$ source image $\mathcal{X}_i$ is also vectorized by $x_i \in \mathbb{R}^{N \times 1}$. Then, we divide each x_i and each y_j into M patches of size $d \times 1$ which may be overlapped, where $x_{i,f}$ is the fth patch of x_i and $y_{j,f}$ is the fth patch of y_j. The observation matrix $y_{:,f} = [y_{1,f}, y_{2,f}, \ldots, y_{m,f}]^T \in \mathbb{R}^{m \times d}$, $f = 1, \ldots, M$ can be modeled as

$$y_{:,f} = Ax_{:,f} + v \tag{9}$$

where $x_{:,f} = [x_{1,f}, x_{2,f}, \ldots, x_{n,f}]^T \in \mathbb{R}^{n \times d}$ refers to the source matrix and v is the additive noise. Thus, large-scale images separation problem (1) can be transformed into M small-scale images separation problems (9). Solving overall BSS problem (5) amounts to solving the following problem one patch by one patch

$$\min_{A,\boldsymbol{\Omega}_i,\boldsymbol{x}_{i,f},\boldsymbol{s}_{i,f}} \lambda\|\boldsymbol{y}_{:,f} - \mathrm{A}\boldsymbol{x}_{:,f}\|_{\mathrm{F}}^2 + \sum_i \mu\|\boldsymbol{s}_{i,f}\|_0 + \sum_i \|\boldsymbol{s}_{i,f} - \boldsymbol{\Omega}_i\boldsymbol{x}_{i,f}\|_{\mathrm{F}}^2 \tag{10}$$

or equivalently,

$$\min_{A,\boldsymbol{\Omega}_i,\boldsymbol{x}_{i,f},\boldsymbol{s}_{i,f}} \lambda\|\boldsymbol{E}_{i,f} - \boldsymbol{a}_{:,i}\boldsymbol{x}_{i,f}^T\|_{\mathrm{F}}^2 + \sum_i \mu\|\boldsymbol{s}_{i,f}\|_0 + \sum_i \|\boldsymbol{s}_{i,f} - \boldsymbol{\Omega}_i\boldsymbol{x}_{i,f}\|_{\mathrm{F}}^2 \tag{11}$$

where $\boldsymbol{E}_{i,f}$ is defined as the multichannel residual of $\boldsymbol{x}_{i,f}$ and its expression is $\boldsymbol{E}_{i,f} = \boldsymbol{y}_{:,f} - \sum_{r=1,r\neq i}^n \boldsymbol{a}_{:,r}\boldsymbol{x}_{r,f}^T$. Here $\boldsymbol{a}_{:,i}$ corresponds to the ith column of A. Because the above optimization subproblems are invariant to scaling [24], we require A is column-normalized matrix and $\boldsymbol{\Omega}_i$ is row-normalized matrix, that is

$$\|\boldsymbol{a}_{:,i}\| = 1, \qquad i = 1,\ldots,n \tag{12}$$

and

$$\|\boldsymbol{w}_{j,i}\|_2 = 1, \qquad j = 1,\ldots,p \tag{13}$$

where $\boldsymbol{w}_{j,i}$ is the jth row of $\boldsymbol{\Omega}_i$. To minimize problem (11), we propose an alternating algorithm by keeping all but one unknown fixed at a time. The estimation process of sources, mixing matrix and analysis dictionary are discussed in the following three subsections, respectively. And the procedure of the proposed algorithm is given at the end of this section.

3.1 Dictionary learning stage

During the dictionary learning stage, we fix $\{\boldsymbol{x}_{i,f}\}_{f=1}^M$ and A to estimate the dictionary Ω_i, so problem (6) can be transformed as follows

$$\{\hat{\boldsymbol{\Omega}}_i,\hat{\boldsymbol{s}}_{i,f}\} = \min_{\boldsymbol{\Omega}_i,\boldsymbol{s}_{i,f}} \sum_f \|\boldsymbol{s}_{i,f}\|_0 \quad s.t.\|\boldsymbol{s}_{i,f} - \boldsymbol{\Omega}_i\boldsymbol{x}_{i,f}\|_{\mathrm{F}}^2 \leq \delta \tag{14}$$

Note that, for each picture $\boldsymbol{x}_i$ we need learn a corresponding dictionary $\hat{\boldsymbol{\Omega}}_i$. Thus, we consider adopt Analysis KSVD to the above problem, in which the learning of ℓ_0 analysis dictionary $\boldsymbol{\Omega}_i$ and the recovery of the correct noiseless signal $\{\tilde{\boldsymbol{x}}_{i,f}\}_{f=1}^M$ from its noisy version $\{\boldsymbol{x}_{i,f}\}_{f=1}^M$ can be achieved. Note that $\boldsymbol{x}_{i,f} = \tilde{\boldsymbol{x}}_{i,f} + \boldsymbol{v}_{i,f}$ where $\boldsymbol{v}_{i,f}$ is a zero-mean white-Gaussian additive noise vector, and the analysis coefficients can be described as $\boldsymbol{s}_{i,f} = \Omega_i\tilde{\boldsymbol{x}}_{i,f}$. Then formula (14) can be transformed into the objective function of Analysis KSVD

$$\{\hat{\boldsymbol{\Omega}}_i,\hat{\tilde{\boldsymbol{x}}}_{i,f}\} = \min_{\boldsymbol{\Omega}_i,\tilde{\boldsymbol{x}}_{i,f}} \sum_f \|\boldsymbol{\Omega}_i\tilde{\boldsymbol{x}}_{i,f}\|_0 \quad s.t. \|\tilde{\boldsymbol{x}}_{i,f} - \boldsymbol{x}_{i,f}\|_{\mathrm{F}}^2 \leq \delta \tag{15}$$

or

$$\{\hat{\boldsymbol{\Omega}}_i,\hat{\tilde{\boldsymbol{x}}}_{i,f},\hat{\Lambda}_{i,f}\} = \min_{\boldsymbol{\Omega}_i,\tilde{\boldsymbol{x}}_{i,f},\Lambda_{i,f}} \sum_f \|\tilde{\boldsymbol{x}}_{i,f} - \boldsymbol{x}_{i,f}\|_{\mathrm{F}}^2 \tag{16}$$
$$s.t. \ \boldsymbol{\Omega}_{i\Lambda_{i,f}}\tilde{\boldsymbol{x}}_{i,f} = 0, \ \mathrm{Rank}(\boldsymbol{\Omega}_{i\Lambda_{i,f}}) = d - r$$

since here we assume the signal $\tilde{\boldsymbol{x}}_{i,f}$ resides in r-dimensional subspace, which implies that $d - r$ rows in $\boldsymbol{\Omega}_i$ are orthogonal to it. These rows define the co-support $\Lambda_{i,f}$ of signal $\tilde{\boldsymbol{x}}_{i,f}$ and $\boldsymbol{\Omega}_{i\Lambda_{i,f}}$ is a submatrix of $\boldsymbol{\Omega}_i$ that contains only the rows indexed in $\Lambda_{i,f}$. In problem (15), we constrain the target sparsity of $\boldsymbol{\Omega}_i\tilde{\boldsymbol{x}}_{i,f}$ to $d - r$, whereas in problem (16), we require the solution to be δ-close to the given noisy signal, where this error tolerance is derived from the noise power. Once given the correct correspondence between r and δ, problem (15) and problem (16) can be considered equivalent. Note that in the proposed method, the Analysis KSVD is based on a two-phase block-coordinate-relaxation approach to solve problem (16). The first phase is called the sparse-coding stage, using a pursuit method, such as the backward greedy (BG) algorithm and the optimized backward greedy (OBG) algorithm, to estimate $\{\tilde{\boldsymbol{x}}_{i,f}\}_1^M$ with $\boldsymbol{\Omega}_i$ fixed. It is worth noting that this method is a greedy algorithm which selecting rows from $\boldsymbol{\Omega}$ one-by-one in a greedy fashion to solve the l_0-norm problem. The problem can be described as follows

$$\{\hat{\tilde{\boldsymbol{x}}}_{i,f},\hat{\Lambda}_{i,f}\} = \min_{\tilde{\boldsymbol{x}}_{i,f},\Lambda_{i,f}} \|\tilde{\boldsymbol{x}}_{i,f} - \boldsymbol{x}_{i,f}\|_F^2 \tag{17}$$
$$s.t. \ \boldsymbol{\Omega}_{i\Lambda_{i,f}}\tilde{\boldsymbol{x}}_{i,f} = 0, \ \mathrm{Rank}(\boldsymbol{\Omega}_{i\Lambda_{i,f}}) = d - r$$

Once $\{\hat{\tilde{\boldsymbol{x}}}_{i,f}\}_1^M$ is computed, we turn to update $\boldsymbol{\Omega}_i$ in the second phase, which is called the atom update step. The optimization is carried out sequentially for each of the rows $\boldsymbol{w}_{j,i}$ in $\boldsymbol{\Omega}_i$. And the update of $\boldsymbol{w}_{j,i}$ should only depend on those signals of $\{\tilde{\boldsymbol{x}}_{i,f}\}_1^M$ that are orthogonal to it, while not on the remaining signals. Therefore, denoting $\tilde{X}_{J,i}$ whose columns are derived from the signal in $\{\tilde{\boldsymbol{x}}_{i,f}\}_1^M$ and are orthogonal to $\boldsymbol{w}_{j,i}$. In addition, denoting by $X_{J,i}$ the corresponding columns of $\{\boldsymbol{x}_{i,f}\}_1^M$. The update step for $\boldsymbol{w}_{j,i}$ can be written as

$$\{\hat{\boldsymbol{w}}_{j,i},\hat{\tilde{X}}_{J,i}\} = \min_{\boldsymbol{w}_{j,i},\tilde{X}_{J,i}} \|\tilde{X}_{J,i} - X_{J,i}\|_F^2 \tag{18}$$
$$s.t. \ \boldsymbol{\Omega}_{i\Lambda_{i,f}}\tilde{\boldsymbol{x}}_{i,f} = 0, \ \forall f \in J$$
$$\mathrm{Rank}(\boldsymbol{\Omega}_{i\Lambda_{i,f}}) = d - r, \ \forall f \in J$$
$$\|\boldsymbol{w}_{j,i}\|_2 = 1$$

Unfortunately, in general, solving problem (18) is a

difficult task. [30] proposed the following approximation to the above optimization goal

$$\hat{w}_{j,i} = \min_{w_{j,i}} \|w_{j,i} X_{J,i}\|_F^2 \quad s.t. \ \|w_{j,i}\|_2 = 1 \qquad (19)$$

For the optimization problem, the $w_{j,i}$ can be updated using the eigenvector associated with the smallest eigenvalue of $X_{J,i} X_{J,i}^T$. Finally, the process is repeated until a fixed number of iterations are reached. And the analysis coefficients of $\tilde{x}_{i,f}$ can be obtained as $\hat{s}_{i,f} = \hat{\Omega}_i \hat{\tilde{x}}_{i,f}$.

3.2 Sources estimating stage

In this stage, we estimate source $x_{i,f}$ by solving the following convex minimization problem

$$\{\hat{x}_{i,f}\} = \min_{x_{i,f}} \lambda \|E_{i,f} - a_{:,i} x_{i,f}^T\|_F^2 + \|s_{i,f} - \Omega_i x_{i,f}\|_F^2 \qquad (20)$$

where Ω_i and $s_{i,f}$ were already obtained in the previous step, and the other sources in $x_{:,f}$ are assumed to be given ($E_{i,f}$ fixed). Then the solution can be obtained by setting the gradient of subproblem (20) to zero

$$\begin{aligned} 0 &= \lambda(x_{i,f} a_{:,i}^T - E_{i,f}^T) a_{:,i} + \Omega_i^T (\Omega_i x_{i,f} - s_{i,f}) \\ &= \lambda x_{i,f} + \Omega_i^T \Omega_i x_{i,f} - \lambda E_{i,f}^T a_{:,i} - \Omega_i^T s_{i,f} \end{aligned} \qquad (21)$$

leading to

$$\hat{x}_{i,f} = (\lambda I + \Omega_i^T \Omega_i)^{-1} (\lambda E_{i,f}^T a_{:,i} + \Omega_i^T s_{i,f}) \qquad (22)$$

where I is the identity matrix. With the help of these corresponding patches $\{x_{i,f}\}_1^M$, the source x_i can be obtained by averaging over the overlapping regions and then the source matrix X can be recovered.

3.3 Mixing matrix estimating stage

At this part, all the unknowns, except A, are fixed and problem (5) is simplified as

$$\hat{a}_{:,i} = \min_{a_{:,i}} \|E_i - a_{:,i} x_i^T\|_F^2 \qquad (23)$$

where the multichannel residual of x_i is defined as $E_i = Y - \sum_{r=1, r \neq i}^n a_{:,r} x_r^T$. The above minimization subproblem is easily solved by setting its gradient to zero, and then, the estimation of $a_{:,i}$ is achieved

$$a_{:,i} = (E_i x_i)(x_i^T x_i)^{-1} \qquad (24)$$

In order to maintain the column norm, the mixing matrix A needs to be normalized after each update of $a_{:,i}$.

3.4 Complexity analysis

Since the proposed algorithm directly uses l_0-norm as the measure of sparsity, the computational complexity of the proposed algorithm must be higher than Fang's method. The detailed analysis of the complexity of the proposed method per iteration is as follows. The complexity analysis of the dictionary learning stage is divided into two parts. In Analysis KSVD, the backward greedy (BG) pursuit algorithms used in sparse coding have complexity $\mathcal{O}(Md^2p)$, and atoms update cost $\mathcal{O}(Mdp)$. Therefore, line 4 costs $\mathcal{O}(Md^2p + Mdp)$. The complexity of updating the representation coefficient (line 6) and computing the residual (line 8) is $\mathcal{O}(Mdp)$ and $\mathcal{O}((n-1)mN)$, respectively. Updating sources in line 9 costs $\mathcal{O}(d^2p + d^3)$. Finally, the mixing matrix estimating stage including computing residual (line 11), updating mixing matrix (line 12) and normalization (line 13) total cost $\mathcal{O}((n-1)mN + mN + 2m)$. Since all these calculations are executed for each source (i.e., $i = 1, \ldots, n$), then the overall computation cost per each iteration of the algorithm would be $\mathcal{O}(nNd^2p + n^2mN + nd^3 + 2mn)$, where $N = M \times d$.

Algorithm 1 ABSS Algorithm

Input: Observation matrix $Y \in \mathbb{R}^{m \times N}$, patch size d, number of dictionary atoms p, number of sources n, target co-rank $d - r$, regularization parameter λ, and total number of iterations T.
Output: Dictionary Ω, sparse coefficients s_i, source matrix X, and mixing matrix A.
Initialization: Set Ω_0 to over-complete DTC, set to a random column normalized matrix A, and compute $X = A^\dagger Y$.
1: **for** $t = 1 \ldots T$ **do**
2:     **for** $i = 1, \ldots, n$ **do**
3:         Extract all the patches $\{x_{i,f}\}_1^M$ from x_i
4:         **Dictionary learning stage**

$$\{\hat{\Omega}_i, \hat{\tilde{x}}_{i,f}, \hat{\Lambda}_{i,f}\} = \min_{\Omega_i, \tilde{x}_{i,f}, \Lambda_{i,f}} \sum_f \|\tilde{x}_{i,f} - x_{i,f}\|_F^2$$

$$s.t. \ \Omega_{i\Lambda_{i,f}} \tilde{x}_{i,f} = 0, \ Rank(\Omega_{i\Lambda_{i,f}}) = d - r$$

5:         Using Analysis KSVD update Ω_i and $\{\tilde{x}_{i,f}\}_1^M$
6:         Compute $\{s_{i,f}\}_1^M$ by $s_{i,f} = \Omega_i \tilde{x}_{i,f}$
7:         **Sources estimating stage**
8:         Compute the residual of $x_{i,f}$ by $E_{i,f} = y_{:,f} - \sum_{r=1, r \neq i}^n a_{:,r} x_{r,f}^T$
9:         $\hat{x}_{i,f} = (\lambda I + \Omega_i^T \Omega_i)^{-1} (\lambda E_{i,f}^T a_{:,i} + \Omega_i^T s_{i,f})$
10:       **Mixing matrix estimating stage**
11:       Compute the residual of x_i by $E_i = Y - \sum_{r=1, r \neq i}^n a_{:,r} x_r^T$
12:       $\hat{a}_{:,i} = E_i x_i (x_i^T x_i)^{-1}$
13:       $\hat{a}_{:,i} = a_{:,i} / \|a_{:,i}\|_2$
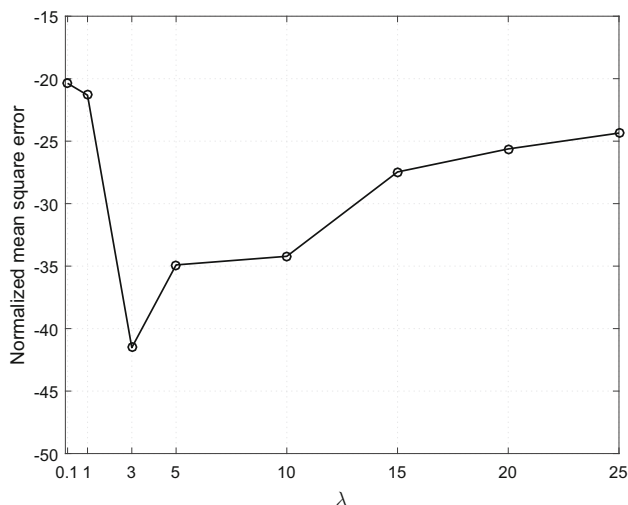14:   **end for**
15: **end for**

Fig. 1 Normalized mean square error (NMSE) as a function of λ

Table 1 PSNRs (dB) and computation times (in seconds) of reconstructed source images

Atoms	Source 1	Source 2	Computation time
70	32.36	29.78	7.5592e+03
80	33.74	35.13	8.2359e+03
90	34.12	37.53	8.9367e+03
100	31.69	32.92	9.1082e+03
110	27.41	28.35	9.1894e+03

4 Simulations

In the first experiment, we chose two very similar source images[1] (shown in Fig.2a) mixed together using a 4×2 full rank random column normalized mixing matrix A to examine the performance of ABSS algorithm. Each 128×128 (=N) mixture and initialized source were divided into patches of size 8×8, which can avoid leading to large dictionaries and provide sufficient training signals for the dictionary learning stage. The overlap percentage of the patches was fixed to 50%. Of course, the higher the overlap percentage lead to a better the recovery result. In addition, three hundred iterations were selected as the stopping criterion. As to the selection of parameter λ, we investigated the effects of choosing a different λ on recovery quality. For this experiment, we calculated the normalized mean square error (NMSE) while varying λ. The NMSE in dB can measure the distance between the mixing matrix A and the estimated matrix $\hat{A}$ and is defined as

$$\text{NMSE} = 10 \log_{10}\left(\frac{\sum_{p,q}(\hat{a}_{p,q} - a_{p,q})^2}{\sum_{p,q}(a_{p,q})^2}\right) \qquad (25)$$

where $\hat{a}_{p,q}$ is the (p, q) element of the estimated matrix $\hat{A}$. And the smaller the value of NMSE, the better the mixing matrix estimation. Figure 1 represents the achieved results for this experiment. It can be found that $\lambda = 3$ is a optimal choice.

Then, we applied the following methods for comparison purposes: FastICA [1], Picard [10], BMMCA [22] and the algorithm proposed by Fang [29] (Computer OS: Windows 8.1, CPU: Intel (R) Core (TM) i5-5200U @ 2.20GHz RAM: 4G). And using the peak signal-to-noise ratio (PSNR) describes the reconstruction performance of the candidate algorithm. PSNR is calculated as

$$\text{PSNR} = 20 \log_{10}(\text{MAX}/\sqrt{\text{MSE}}) \qquad (26)$$

where the MAX indicates the maximum possible pixel value of the image, such as for a uint-8 image, the MAX equals to 255. And the MSE refers to mean square errors given by $\text{MSE} = (1/N)\|\mathcal{X} - \hat{\mathcal{X}}\|_{\text{F}}^2$, where $\mathcal{X}$ and $\hat{\mathcal{X}}$ are the source image and recovered image. The better the quality of the image, the higher value of PSNR. Note that in the ABSS algorithm, the number of atoms of dictionary is set to 80. Although there is not much can be theoretically said about choosing the optimum dictionary size (or redundancy factor d/p), we try to give an explanation by experiment. Table 1 shows the PSNR values for different numbers of atoms.

As shown in Table 1, the separation effect is better when the number of atoms is 80 or 90, and the computation time of 90 atoms is greater than that of 80 atoms. For comprehensive consideration, we set the number of atoms to 80. Then, Fig. 2 shows the reconstructed source images of the five test algorithms. And Table 2 presents the average PSNR results.

We observed that the proposed algorithm achieves the best performance. BMMCA and Fang have almost similar results, and the results of FastICA and Picard are not as good as the other three algorithms in processing very similar source images. As for another measure on the performance of the proposed method, the reconstruction error as a function of the number of iterations is shown in Fig. 3, which shows that a monotonic decrease in the value of the separation error is achieved. In addition, the asymptotic error decreases relatively rapidly with a very small value around the 20th iteration and is almost zero around the 300th iteration. Therefore, the proposed algorithm is convergent well.

Next, in order to make the experimental results more convincing, we did another experiment to separate four sources from six noiseless mixtures. The four selected source images[2] have very different morphologies,

---

[1] Available at: http://md.cosmostat.org/Generalized_MCA.html.

[2] Available at: http://md.cosmostat.org.

(a) Mixture

Source 1

Source 2

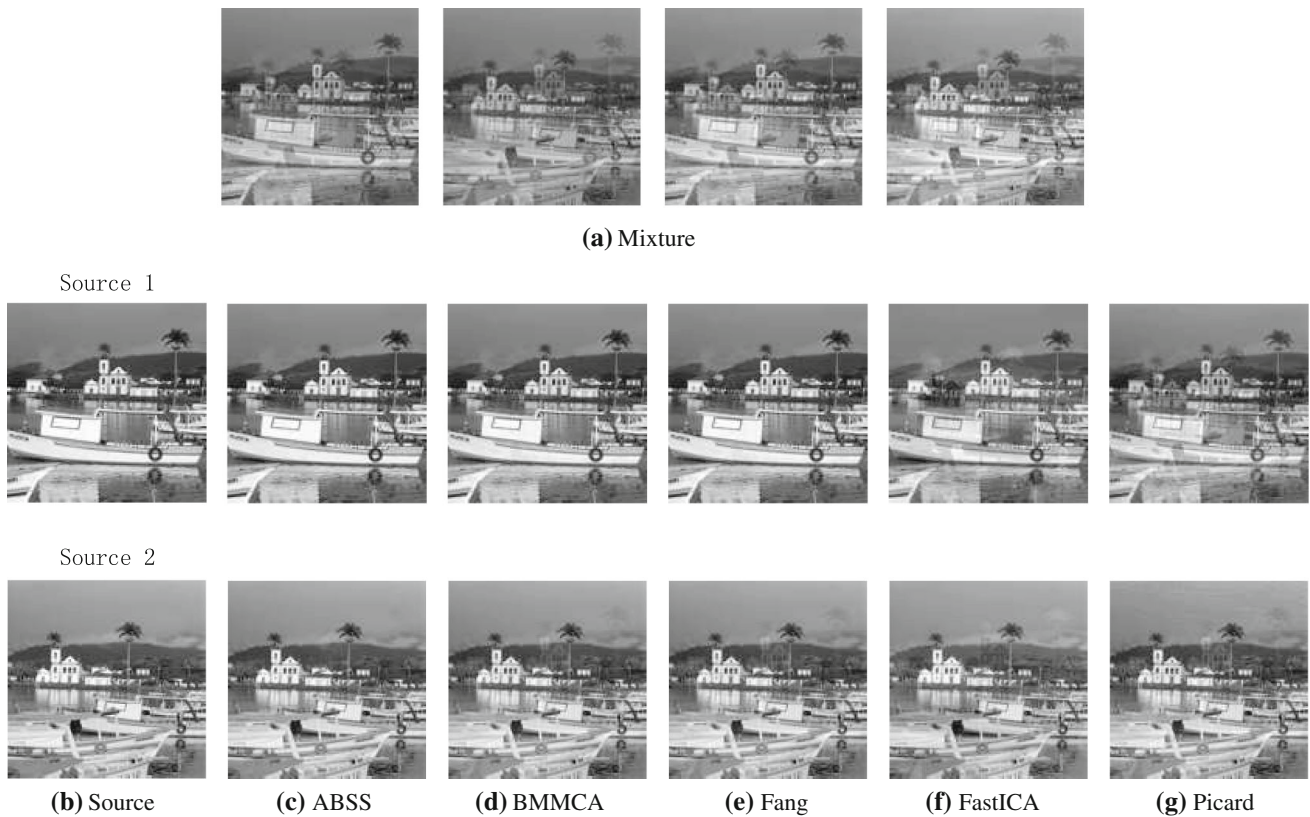| **(b)** Source | **(c)** ABSS | **(d)** BMMCA | **(e)** Fang | **(f)** FastICA | **(g)** Picard |

Fig. 2 **a** Mixtures (the first row). **b** Source images (the first column). **c–g** are separation results by ABSS, BMMCA, Fang, FastICA and Picard (the second–sixth column), respectively

Table 2 PSNRs (dB) of the five test algorithms

Algorithm	ABSS	BMMCA	Fang	FastICA	Picard
Source 1	36.89	35.59	32.81	23.11	21.14
Source 2	35.76	31.72	31.15	27.35	27.04

including noiselike texture and bricklike texture. As found from Fig. 4, the parameter λ should set to 3. Moreover, the rest of parameters were the same as those in the first experiment. Figure 5 and Table 3 illustrate the separation results of applying all these methods mentioned above (BMMCA, Fang, FastICA and Picard) and the corresponding MSEs. As shown in Fig. 5c the proposed method could successfully recover the image sources. And the achieved MSEs, given in Table 3, are lowest for the proposed method for all image sources except for the cartoon boy. Other methods do not perform as well as the proposed method. For example, as shown in Fig. 5d, although BMMCA has perfectly recovered the cartoon boy with the lowest MSE among all other methods, it has a problem in recovering *bricklike* texture. In addition, Fang, FastICA and Picard all have some difficulties in separation of some images, separating all the image sources but adding some interference from other sources.
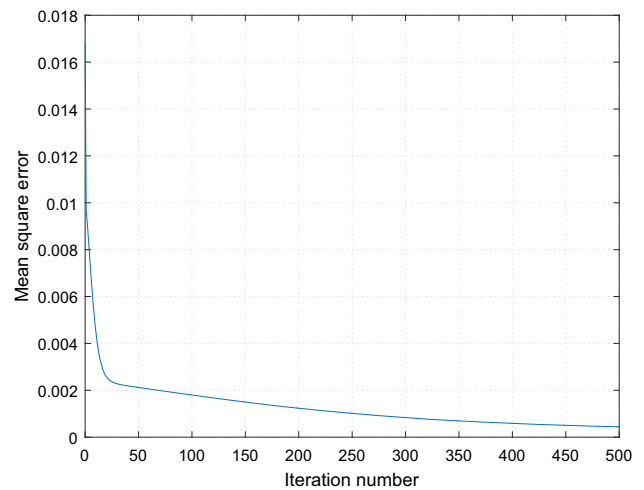

Fig. 3 MSE varies with the number of iterations, note that the elements of image sources have amplitude in the range [0, 1]

In the last simulation, we investigated the performance of all the methods in different noise levels. Gaussian noises which standard deviations varied from 5 to 15 ($\sigma = 15$) are added to the mixtures which are mixed from another two image sources (Lena and Boat)[3]. The noisy mixtures are
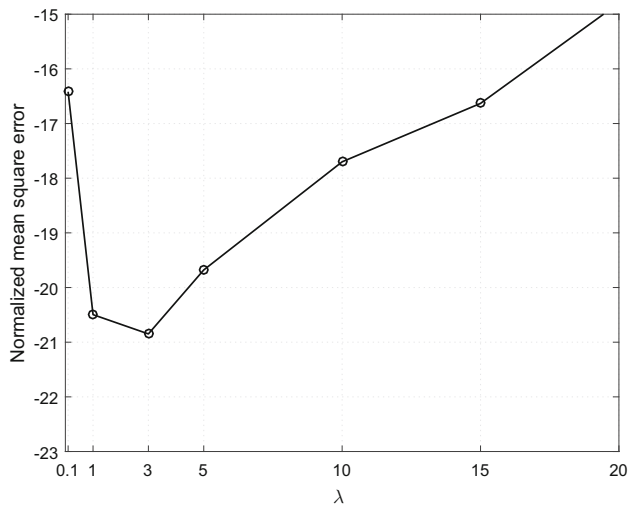
---

[3] Available at: https://elad.cs.technion.ac.il/software/.

Fig. 4 Normalized mean square error (NMSE) as a function of λ

Table 3 MSEs of the five test algorithms

Algorithm	ABSS	BMMCA	Fang	FastICA	Picard
Bricklike texture	0.0608	0.1394	0.2164	0.0933	0.0720
Cartoon boy	0.1814	0.0427	0.2776	0.1278	0.2652
Barbara	0.0459	0.0463	0.1466	0.1713	0.2288
Noiselike texture	0.0773	0.0988	0.0889	0.2346	0.0859

shown in Fig. 5a. As shown in Fig. 6, selection of λ is dependent on the noise standard deviation. Therefore, at noise levels ranging from 5 to 15, the optimal choices of λ were 1, 5 and 10, respectively. The other experimental settings and procedure were the same as the previous experiment. Table 4 presents the average PSNRS results in
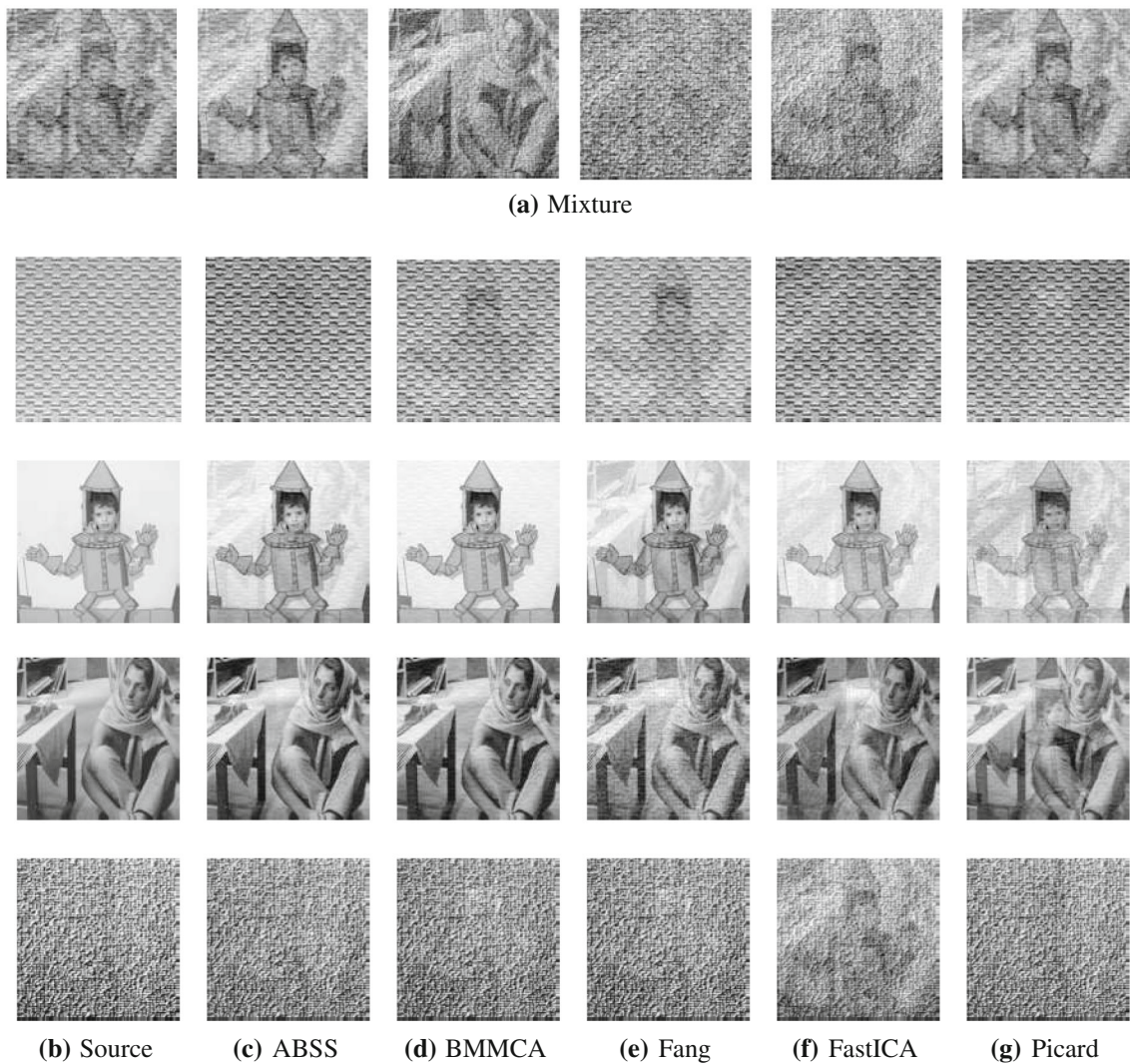

(a) Mixture

(b) Source **(c)** ABSS **(d)** BMMCA **(e)** Fang **(f)** FastICA **(g)** Picard

Fig. 5 a Mixtures (the first row). **b** Source images (the first column). **c–g** are separation results by ABSS, BMMCA, Fang, FastICA and Picard (the second–sixth column), respectively
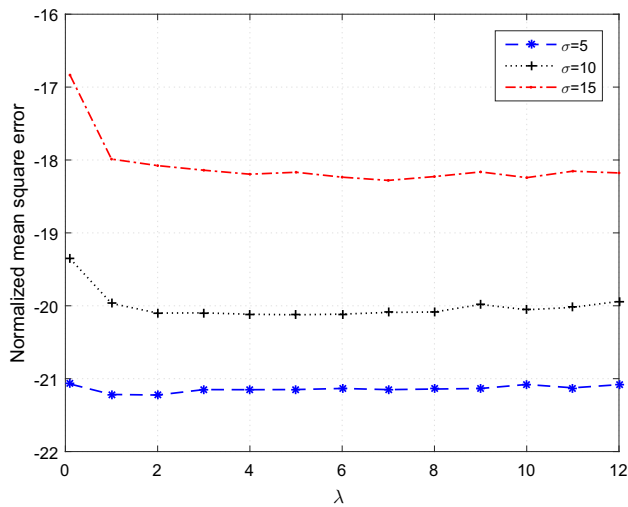
Fig. 6 Normalized mean square error (NMSE) as a function of λ

Table 4 PSNRs (dB) of the five test algorithms

σ	Source	ABSS	BMMCA	Fang	FastICA	Picard
5	Lena	24.19	21.77	23.55	23.30	23.39
	Boat	27.78	26.75	27.30	26.37	24.78
10	Lena	21.30	19.00	20.36	19.72	19.64
	Boat	24.24	21.89	21.98	18.01	21.72
15	Lena	20.25	18.00	18.28	18.18	17.19
	Boat	21.52	20.38	19.25	17.00	20.04

different noise levels. And the separation results in the noise level 10 are shown in Fig. 7c–g. We observed that all algorithms successfully separated the noise mixtures, but the proposed algorithm achieves a better separation performance than the other algorithms. When the noise level is 5, the separation results of all methods are similar except Picard, but as the noise level increases, FastICA performs obviously worse and Picard has superior performance than FastICA. In addition, ABSS presents successful performance improvement in denoising, for example, Lena's facial details are the most legible among the five methods. Overall, the above experimental results have indicated that ABSS has a significant performance improvement in both separation and denoising.

5 Conclusions and future work

In this paper, we proposed a novel algorithm based on the analysis sparse constraint of the source over an adaptive analysis dictionary to address BSS problem, where the number of mixtures is not less than the number of sources. Moreover, l_0-norm was chosen as the measure of sparsity to make better use of the sparsity constrain. The simulation results on both noisy and noiseless mixtures have been confirmed that the proposed method improves the separation performance significantly.

In fact, the way of updating dictionaries and sources is completely different from the literature [29]. In the analysis
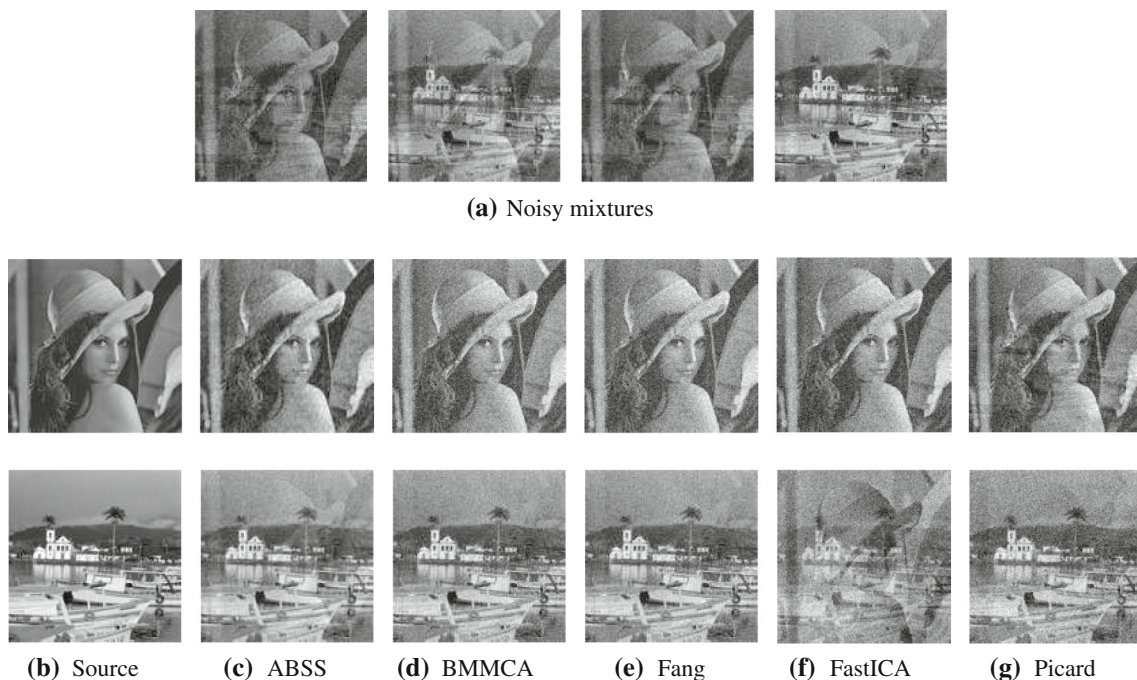

(a) Noisy mixtures

(b) Source   **(c)** ABSS   **(d)** BMMCA   **(e)** Fang   **(f)** FastICA   **(g)** Picard

Fig. 7 **a** Noisy mixtures when $\sigma = 10$ (the first row). **b** Source images (the first column). **c**–**g** are separation results by ABSS, BMMCA, Fang, FastICA and Picard (the second–sixth column), respectively

dictionary learning stage, we adopt Analysis KSVD [30] to adaptively estimate the analysis dictionary from mixed sources, whereas Fang uses SP-ADL directly proposed by the literature [31], which is an efficient algorithm for obtaining the analysis dictionary from a given data set in a KSVD-like manner. And in the sources estimating stage, since the subproblem we need to solve is a convex function, the sources can be uniquely obtained by a simple least square linear regression instead of the split Bergman iteration (SBI) algorithm introduced by [32], which use directly in the literature [29] and need five iterations to converge to an approximate solution. Thus, the convergence rate of the proposed algorithm is faster than the Fang's method. Finally, the simulation results on both noisy and noiseless mixtures have been confirmed that using l_0-norm enhances the separability of the sources and the proposed method improves the separation performance significantly.

However, due to the greedy algorithm—backward greedy algorithm is used in the dictionary learning stage, it inevitably slows down the proposed algorithm speed, which is also the reason why the computation time in Table 1 is so high. Further work is required to speed up the algorithm and make it suitable for large-scale problems, for example, using some novel and more efficient dictionary learning algorithms [33, 34] for BSS problem. In addition, this work only focuses on the general overdetermined BSS problem and tries to find some new angle to deal with it perfectly. It would be desired to extend this related method to resolve the underdetermined BSS problem in the future, in which the number of mixtures is less than that of the sources, and this kind of problem is harder to deal with than the problem mentioned in this work.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural Netw 13(4):411–430
2. Bell A, Sejnowski T (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Comput 7(6):1129–1159
3. Gaeta M, Lacoume JL (1990) Source separation without prior knowledge: the maximum likelihood solution. In: Proceeding EUSIPCO'90, pp 621–624
4. Belouchrani A, Cardoso JF (1994) Maximum likelihood source separation for discrete sources. In: Proceeding EUSIPCO'94, pp 768–771
5. Montoya-Martínez J, Cardoso JF, Gramfort A (2017) Caveats with stochastic gradient and maximum likelihood based ica for eeg. Lecture Note Computer Sci 10169:279–289
6. Tillet P, Kung HT, Cox D (2017) Infomax-ica using hessian-free optimization. Proceeding ICASSP'2017, pp 2537–2541
7. Zibulevsky M (2003) Sparse source separation with relative newton method. Proceeding ICA 2003:897–902
8. Choi H, Choi S (2007) A relative trust-region algorithm for independent component analysis. Neurocomputing 70(7):1502–1510
9. Byrd R, Lu P, Nocedal J, Zhu C (2003) A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 16(5):1190–1208
10. Ablin P, Cardoso J, Gramfort A (2018) Faster independent component analysis by preconditioning with hessian approximations. IEEE Trans Signal Process 66(15):4040–4049
11. Lee D, Seung H (1999) Learning the parts of objects by nonnegative matrix factorization. Nature 401(6755):788–791
12. Boutsidis C, Gallopoulos E (2008) Svd based initialization: a head start for nonnegative matrix factorization. Pattern Recognit 41(4):1350–1362
13. Deng Z, Zhang S, Yang L, Zong M, Cheng D (2018) Sparse sample self-representation for subspace clustering. Neural Comput Appl 29:43–49
14. Zhu X, Suk HI, Huang H, Shen D (2017) Low-rank graph-regularized structured sparse regression for identifying genetic biomarkers. IEEE Trans Big Data 3(4):405–414
15. Zhang H, Wang S, Xu X, Chow TWS, Wu QMJ (2018) Tree2vector: learning a vectorial representation for tree-structured data. IEEE Trans Neural Netw Learn Syst 29(11):5304–5318
16. Alon U, Barkai N, Notterman D, Gish K, Ybarra S, Mack D, Levine A (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci United States of America 96(12):6745–6750
17. Shipp MA et al (2002) Diffuse large b-cell lymphoma outcome prediction by gene-expressionprofilingandsupervisedmachine-learning. NatureMed 8(1):68–74
18. Huang H, Zhao H, Li X, Ding S, Zhao L, Li Z (2018) An accurate and efficient device-free localization approach based on sparse coding in subspace. IEEE Access 6:61782–61799
19. Starck JL, Elad M, Donoho D (2004) Redundant multiscale transforms and their application for morphological component separation. Adv Imagin Electron Phys 132(4):287–348
20. Bobin J, Moudden Y, Starck JL, Elad M (2006) Morphological diversity and source separation. IEEE Signal Process Lett 13(7):409–412
21. Bobin J, Starck JL, Fadili J, Moudden Y (2007) Sparsity and morphological diversity in blind source separation. Image Process, IEEE Trans Image Process 16(11):2662–2674
22. Abolghasemi V, Ferdowsi S, Sanei S (2012) Blind separation of image sources via adaptive dictionary learning. IEEE Trans Image Process 21(6):2921–2930
23. Aharon M, Elad M, Bruckstein A (2006) K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322
24. Zhao X, Zhou G, Dai W, Xu T, Wang W (2013) Joint image separation and dictionary learning. Proceedings DSP'2013, pp 1–6
25. Dai W, Xu T, Wang W (2011) Simultaneous codeword optimization (simco) for dictionary update and learning. IEEE Trans Signal Process 60(12):6340–6353

26. Nam S, Davies M, Elad M, Gribonval R (2011) The cosparse analysis model and algorithms. Appl Comput Harmonic Anal 34(1):30–56

27. Giryes R, Nam S, Elad M, Gribonval R, Davies M (2012) Greedylike algorithms for the cosparse analysis model. Linear Algebra Appl 441(1):22–60

28. Elad M, Milanfar P, Rubinstein R (2007) Analysis versus synthesis in signal priors. Inverse Probl 23(3):947

29. Fang W, Wang H, Xu B, Zhang Y (2016) Blind source separation using analysis sparse constraint. Electron Lett 52(13):1112–1114

30. Rubinstein R, Peleg T, Elad M (2013) Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. IEEE Trans Signal Process 61(3):661–677

31. Zhang Y, Wang H, Yu T, Wang W (2013) Subset pursuit for analysis dictionary learning. Proceeding EUSIPCO'2013, pp 1–5

32. Cai JF, Osher S, Shen Z (2009) Split bregman methods and frame based image restoration. Multiscale Model and Simul 8(2):337–369

33. Wu F, Jing X, Sun Y, Sun J, Huang L, Cui F, Sun Y (2018) Cross-project and within-project semisupervised software defect prediction: a unified approach. IEEE Trans Reliab 67(2):581–587

34. Wu F, Xi W, Han L, Jing X, Ji Y (2019) Multi-view synthesis and analysis dictionaries learning for classification. IEICE Trans Inf Syst E102 D(3):659–662