An Algorithmic Framework for Sparse Bounded Component Analysis

Eren Babatas and Alper T. Erdogan

Koc University, Istanbul, Turkey

***Abstract*—Bounded Component Analysis (BCA) is a recent approach which enables the separation of both dependent and independent signals from their mixtures. This article introduces a novel deterministic instantaneous BCA framework for the separation of sparse bounded sources. The framework is based on a geometric maximization setting, where the objective function is defined as the volume ratio of two objects, namely the principal hyperellipsoid and the bounding ℓ_1-norm ball, defined over the separator output samples. It is shown that all global maxima of this objective are perfect separators. The article also provides the corresponding iterative algorithms for both real and complex sparse sources. The numerical experiments illustrate the potential benefits of the proposed approach, with applications on image separation and neuron identification.**

***Index Terms*—Sparse Source Separation, Bounded Component Analysis, Independent Component Analysis, Blind Source Separation, Sparse Component Analysis**

I. Introduction

Bounded Component Analysis (BCA) is a framework for blind separation of both independent and dependent sources from their mixtures [1]. It is demonstrated to be useful especially for settings involving dependent sources such as natural images, and/or short data records such as digital communication packets where the mutual independence assumption used by Independent Component Analysis (ICA) approach is either not directly applicable or ill-posed [2]–[5].

Having its roots in the modeling of neuro-physiological processes, ICA has been the most popular framework for the Blind Source Separation (BSS) problem [6]–[8]. The mutual independence of sources is a sufficient assumption for obtaining the solution of the linear BSS problem, and this fact has led to several tractable algorithms and analysis results for BSS (see for example [9]–[11]). However, for some applications, the source signals can be dependent/correlated. Even in the case where the generating processes may be independent, the available source samples may not be sufficiently long to reflect this underlying stochastic property.

BCA addresses this issue through relaxing the *mutual independence assumption* by replacing it with the weaker *domain separability assumption* under the side information that sources take their values from a compact set [1]. In this sense, BCA can be viewed as an extension of ICA for bounded magnitude signals allowing the separation of both dependent and independent sources. In [2], a geometric framework for developing BCA algorithms for simultaneous separation of sources was proposed. This framework was based on the assumption that the source samples appropriately fill out the rectangular region (ℓ_∞-norm ball) which is the Cartesian product of individual source domains. This seems to be a nice fit for many of the practical signals such as digital communication constellations, natural images, harmonic oscillations with subgaussian nature. However, it is not appropriate for heavy tailed or sparse signals with supergaussian distribution. It is the goal of this current article to extend the BCA framework in [2] to the settings involving sparse bounded sources.

We should note at this point that the sparsity concept has been at the focus of several different research fields including signal processing, machine learning and neuroscience for several decades. One of the major drivers of this area has been the efficient representation and denoising of signals through complete or overcomplete representations especially based on wavelet transforms [12], [13]. Connected with this research, the emergence of mathematically tractable cost functions used for measuring (non)sparsity, such as ℓ_1-norm, and related convex optimization based formulations generated significant boost and attracted interest from a wider research community. Another important driver has been the successful modeling of the visual process in brain through sparse coding [14], which led to further sparse coding based approaches in both computational neuroscience and machine learning fields [15]–[17]. The main projection of the sparsity driven research in signal processing community has been the compressed sensing field where the main interest has been the exploitation of the sparsity side information (in an appropriate basis) in the sampling of signals [18]–[20].

Sparsity side information has also been exploited to solve BSS problem especially in conjunction with the ICA approach [7]. The ICA approaches with sparsity promoting contrast functions can be listed as the main examples in this field, e.g., [21]. Sparsity side information can be also used to replace the independence assumption in separating sources, e.g. [22]. General BSS approaches exploiting sparsity are referred as Sparse Component Analysis (SCA) methods, e.g., [23], [24]. SCA algorithms have two main branches: underdetermined SCA, with more sources than mixtures, and (over)determined SCA. We can list cluster based approaches, such as [22], [25], and alternating projection based dictionary learning algorithms, such as [26], [27], as examples of underdetermined SCA. For (over)determined SCA methods, we can give Maximum Likelihood / Maximum A Posteriori based approaches [28], [29] as examples.

In this article, we expand the approach proposed in [2] to sparsely natured bounded signals for the (over)determined mixing scenario. We modify the geometric framework in [2]

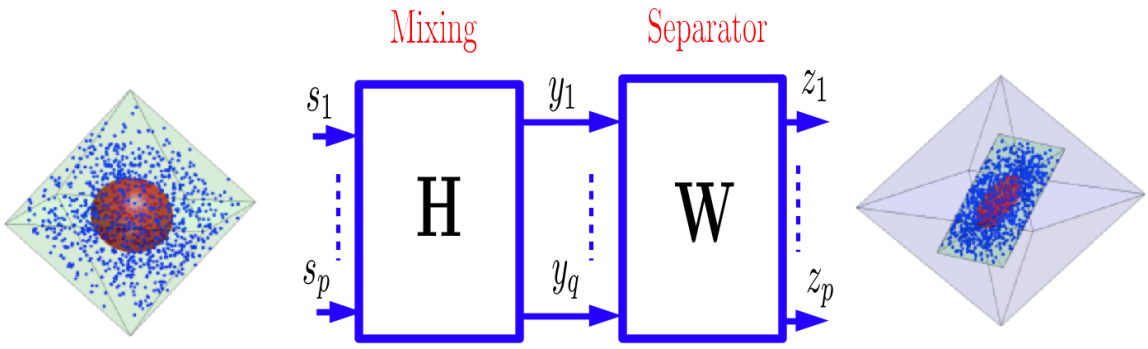

Fig. 1: Geometric objects for the proposed sparse BCA framework. Diamond shaped boxes on the left and right sides are the bounding l_1-norm-balls for the source and separator output samples respectively. Red balls on the left and right sides are the principal hyperellipsoids corresponding to the source and separator output samples respectively. Green polytope on the right is the image of the input l_1-norm-ball under the overall mapping $\mathbf{G} = \mathbf{WH}$.

to reflect the sparsity of the original source vectors. Similar to [2], we pose the Sparse BCA (SBCA) problem as the volume ratio maximization for the objects defined at the separator output domain. As the main deviation from the approach in [2], the sources are assumed to be sparse in the sense that they are located in an ℓ_1-norm ball, and they are locally dominant only for a small subset of samples. The proposed approach is completely deterministic and potentially applicable to separation of both dependent and independent sources. The contributions covered in this article were partly presented in the conference article [30]. The present article provides a comprehensive treatment of the proposed framework including new sections on

- The algorithm extensions for the complex source signals and noisy mixtures (in Section IV),
- Two proposals for the algorithm acceleration (in Section V). In particular, we provide a new algorithm update rule with significant computational simplification, based on the weighted updates. Furthermore, this form enables comparison with Infomax ICA algorithm,
- Algorithm application results on some new set of synthetic signals (in Section VI-A) as well as some Neuroimaging signals. In fact, a tool based on SBCA algorithm for the neural activity identification by using calcium (Ca^{2+}) fluorescent imaging records is introduced in Section VI-C.

The organization of the article is as follows: The SBCA setup assumed throughout the article is introduced in Section II. In Section III, SBCA framework is proposed. In the same section, it is shown that the global maximizers of the proposed objective are perfect separators. An iterative algorithm for the proposed optimization setting is also provided. The algorithmic extensions of SBCA for complex sources and noisy case are provided in Section IV. In Section V, the algorithm acceleration through weighted update rule is introduced, which turns out to have a notable resemblance to Infomax (ICA) algorithm. In Section VI, numerical examples demonstrating the potential utility of the proposed approach relative to some existing SCA algorithms, and the BCA algorithms in [1], [2] are provided. We also illustrate the application of the proposed algorithm, for image separation, and neuron/neural activity identification in Ca^{2+} fluorescent imaging. Finally, Section VII is the conclusion.

We define the following notation table for set A, $\mathbf{G} \in \mathbb{R}^{p\times p}$, $\mathbf{x} \in \mathbb{R}^p$, $a, b, k \geq 1 \in \mathbb{Z}$:

Notation	Explanation
$\|\mathbf{x}\|_a$	ℓ_a-norm defined as $\sqrt[a]{\sum_{n=1}^{p} \|x_n\|^a}$
$\|\mathbf{G}\|_{a,b}$	Induced matrix norm defined as $\sup_{\|\mathbf{x}\|_b \leq 1} \|\mathbf{Gx}\|_a$
$\mathbf{G}_{:,k}$ ($\mathbf{G}_{k,:}$)	k^{th} column (row) of $\mathbf{G}$
$\mathbf{Co}A$	Convex Hull of the set A
$1_{\text{condition}}$	Indicator function: 1 if the condition is true, 0 otherwise.
sign$\{\mathbf{A}\}$	replaces positive (negative) entries of $\mathbf{A}$ with 1 (-1)
$\mathbb{R}e\{\mathbf{A}\}$	real part of $\mathbf{A}$
$\mathcal{I}m\{\mathbf{A}\}$	imaginary part of $\mathbf{A}$
signc$\{\mathbf{A}\}$	complex sign operator: $\mathbb{R}e\{\text{sign}\{\mathbf{A}\}\} + i\mathcal{I}m\{\text{sign}\{\mathbf{A}\}\}$

TABLE I: Notation Table.

II. Sparse Bounded Component Analysis Setup

Throughout this article, we assume the data model shown in Fig. 1. In this setup,

- There are p sources represented by the vector $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_p \end{bmatrix}^T \in \mathbb{R}^p$. It is also assumed that there are L samples of these sources represented by the set $\mathcal{S} = \{\mathbf{s}(n) \in \mathbb{R}^p, n = 1, \dots L\}$. Furthermore, the source vectors are bounded in magnitude and lie in an ℓ_1-norm ball, i.e.,

$$\mathbf{s}(n) \in \mathcal{B}(\mathcal{S}), \qquad n = 1, \dots, L, \tag{1}$$

where

$$\mathcal{B}(\mathcal{S}) = \{\mathbf{q} \in \mathbb{R}^p : \|\mathbf{q}\|_1 \leq 1\}. \tag{2}$$

This implies that each source takes its values from the interval $[-1, 1]$ for the purpose of simplifying future

expressions, without any loss of generality. In the more general case, $\mathcal{B}_s$ could be selected as a weighted ℓ_1-norm ball. We also note that there are no stochastic assumptions made about the source vector such as mutual independence of its components.
In order to clarify the ℓ_1-norm ball choice for the sources, we can provide the following explanation: the actual indicator of non-sparseness is ℓ_0-norm which counts the number of non-zero entries in a vector. The fact that ℓ_0 is non-convex, and it is not an actual norm, led to the use of ℓ_1-norm as its algorithmic convex surrogate. It is well-known that this replacement has led to remarkable success in constructing sparseness centered algorithms. In fact, it is the goal of this article to use ℓ_1-norm as such surrogate to promote sparseness. It is clear that the sources lying in ℓ_p-norm balls for some non-negative $p < 1$, would also lie inside ℓ_1-norm ball, and therefore, fall into the domain of interest where the framework proposed in this article works.

- The mixing system is a linear and memoryless mapping which is represented by the matrix $\mathbf{H} \in \mathbb{R}^{q\times p}$. We assume that $q \geq p$, i.e., $\mathbf{H}$ is a tall or square matrix. In addition, it is also full rank.
- The memoriless mixtures of the sources are represented with the vector $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_q \end{bmatrix}^T$. The relation between the mixtures and the sources is defined by

$$\mathbf{y}(n) = \mathbf{H}\mathbf{s}(n), \qquad n = 1, \dots, L. \tag{3}$$

- $\mathbf{W} \in \mathbb{R}^{p\times q}$ is the separator matrix, and its outputs are represented by the vector $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & \dots & z_p \end{bmatrix}^T \in \mathbb{R}^p$. The relation between separator outputs and mixtures is given by

$$\mathbf{z}(n) = \mathbf{W}\mathbf{y}(n), \qquad n = 1, \dots L. \tag{4}$$

- We define $\mathbf{G} \in \mathbb{R}^{p\times p}$ as the cascade of the separator and the mixing systems, i.e., $\mathbf{G} = \mathbf{W}\mathbf{H}$. This would define the overall mapping from sources to the separator outputs in the form

$$\mathbf{z}(n) = \mathbf{G}\mathbf{s}(n) \qquad n = 1, \dots, L. \tag{5}$$

We can explicitly pose the *BSS Problem* as follows: Given only the mixture samples $\mathcal{Y} = \{\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(L)\}$, and no information about the mixing system $\mathbf{H}$, find the original source signals.

Due to its unsupervised nature, we can accomplish this goal up to some uncertainty. For the ideal case, we expect only one entry in each row of $\mathbf{G}$ to be non-zero to achieve separation. This condition can be represented as

$$\mathbf{G} = \mathbf{P}\mathbf{D}, \tag{6}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a full rank diagonal matrix. Therefore, we will refer to a $\mathbf{W}$ matrix as perfect separator if and only if its corresponding $\mathbf{G}$ matrix satisifies the condition in (6).

III. Sparse BCA Framework

In this section, we propose a new algorithmic framework for the BSS problem introduced in the previous section. The proposed approach is the adaptation of the geometric framework introduced in [2], where we pose obtaining the separator as maximizing the volume ratio of two objects defined for the separator outputs.

In Section III-A, we formulate the corresponding optimization setting. In Section III-B, we show that all global optima of this setting are perfect separators. In Section III-C, we compare the proposed objective with the quasi ML based SCA approaches. Finally, in Section III-D, we provide explicit algorithm iterations to maximize the proposed objective.

A. Geometric Optimization for Sparse BCA

As stated above, in this section we propose a geometric optimization setup for the sparse BCA problem. We start by defining the set of separator output samples for a given separator matrix $\mathbf{W}$, and the corresponding $\mathbf{G}$ as

$$\mathcal{Z} = \{\mathbf{W}\mathbf{y}(1), \mathbf{W}\mathbf{y}(2), \dots, \mathbf{W}\mathbf{y}(L)\}, \tag{7}$$
$$= \{\mathbf{G}\mathbf{s}(1), \mathbf{G}\mathbf{s}(2), \dots, \mathbf{G}\mathbf{s}(L)\}. \tag{8}$$

The members of the set $\mathcal{Z}$, i.e., the separator outputs, are represented as (blue) dots at the right side of the Fig. 1. Corresponding to this set, we define the following objects:

- **Principal Hyper-Ellipsoid**: This object reflects the "shape" of the separator output samples based on their sample covariance matrix. We first define the sample covariance matrix for $\mathcal{Z}$ as

$$\hat{\mathbf{R}}(\mathcal{Z}) = \frac{1}{L}\sum_{n=1}^{L} \mathbf{z}(n)\mathbf{z}(n)^T - \hat{\mu}(\mathcal{Z})\hat{\mu}(\mathcal{Z})^T \tag{9}$$

where $\hat{\mu}(\mathcal{Z})$ is the corresponding sample mean given by

$$\hat{\mu}(\mathcal{Z}) = \frac{1}{L}\sum_{n=1}^{L} \mathbf{z}(n). \tag{10}$$

Based on these definitions, the principal hyper-ellipsoid corresponding to $\mathcal{Z}$ is given by

$$\mathcal{E}(\mathcal{Z}) = \{\mathbf{q} : (\mathbf{q} - \hat{\mu}(\mathcal{Z}))^T \hat{\mathbf{R}}(\mathcal{Z})^{-1} (\mathbf{q} - \hat{\mu}(\mathcal{Z})) \leq 1\} \tag{11}$$

The principal hyper-ellipsoid for the separator output samples, $\mathcal{E}(\mathcal{Z})$ is illustrated as the 3-dimensional (red) ellipsoid on the right of Fig. 1. In the same figure, on the left, principal hyper-ellipsoid $\mathcal{E}(\mathcal{S})$ for the corresponding source samples is also shown. Here, $\mathcal{E}(S)$ is defined similarly as

$$\mathcal{E}(\mathcal{S}) = \{\mathbf{q} : (\mathbf{q} - \hat{\mu}(\mathcal{S}))^T \hat{\mathbf{R}}(\mathcal{S})^{-1} (\mathbf{q} - \hat{\mu}(\mathcal{S})) \leq 1\} \tag{12}$$

where $\hat{\mu}(\mathcal{S})$ is the sample mean for the source samples in $\mathcal{S}$, given by

$$\hat{\mu}(\mathcal{S}) = \frac{1}{L}\sum_{n=1}^{L} \mathbf{s}(n), \tag{13}$$

and $\hat{\mathbf{R}}(\mathcal{S})$ is the sample covariance for the set $\mathcal{S}$ which is given by

$$\hat{\mathbf{R}}(\mathcal{S}) = \frac{1}{L}\sum_{n=1}^{L}\mathbf{s}(n)\mathbf{s}(n)^T - \hat{\mu}(\mathcal{S})\hat{\mu}(\mathcal{S})^T. \tag{14}$$

Note that from the relation $\mathbf{z}(n) = \mathbf{G}\mathbf{s}(n)$, it follows that the relation $\hat{\mathbf{R}}(\mathcal{Z}) = \mathbf{G}\hat{\mathbf{R}}(\mathcal{S})\mathbf{G}^T$ holds.

- **Bounding ℓ_1-Norm Ball**: This is the smallest ℓ_1-norm ball containing a given compact set. For the separator output samples, we define the corresponding bounding ℓ_1-norm ball, $\mathcal{B}(\mathcal{Z})$ as

$$\mathcal{B}(\mathcal{Z}) = \{\mathbf{q} : \|\mathbf{q}\|_1 \leq \max_{n\in\{1,\ldots,L\}} \|\mathbf{z}(n)\|_1\}. \tag{15}$$

In Fig. 1, $\mathcal{B}(\mathcal{Z})$ is illustrated as the 3-dimensional diamond shaped box on the right. In the same figure, $\mathcal{B}(\mathcal{S})$ is shown on the left.

Based on these definitions, and following the geometric approach in [2], we define the sparse BCA objective as the volume ratio

$$\bar{J}(\mathbf{W}) = \frac{\text{Volume}(\mathcal{E}(\mathcal{Z}))}{\text{Volume}(\mathcal{B}(\mathcal{Z}))} \tag{16}$$

$$= \frac{C_e\sqrt{\det(\hat{\mathbf{R}}(\mathcal{Z}))}}{C_l(\max_{n\in\{1,\ldots,L\}}\|\mathbf{z}(n)\|_1)^p} \tag{17}$$

where $C_e = \pi^{p/2}/\Gamma(p/2+1)$ is the scaling constant for the p-dimensional hyper-ellipsoid and $C_l = 2^p/p!$ is the volume for the unity ℓ_1-norm ball. Dropping the dimension dependent constants C_e, C_l, the new objective is obtained as

$$J(\mathbf{W}) = \frac{\sqrt{\det(\hat{\mathbf{R}}(\mathcal{Z}))}}{(\max_{n\in\{1,\ldots,L\}}\|\mathbf{z}(n)\|_1)^p}. \tag{18}$$

Therefore, the sparse BCA problem is posed as the problem of choosing the separator matrix $\mathbf{W}$ to maximize the objective $J(\mathbf{W})$ in (18). The denominator of the objective includes the maximum ℓ_1 norm of the separator output samples so that maximization of the objective means minimization of the ℓ_1 norm of the separator output samples which is a well-known way for measuring sparsity, whereas the determinant term at the nominator simply acts as a regulator to avoid all zeros solution and guarantee a full rank mapping between sources and separator outputs. More detailed understanding of how the maximization of this objective function enables perfect separation is based on the proof in the next section. Therefore, this objective function is a suitable criterion for sparse source separation. In the next subsection, we prove that under a certain local dominance assumption, the global maxima of this optimization setting are perfect separators.

B. Global Optimality of Perfect Separators

In order to ensure the global optimality of the perfect separators with respect to the objective in (18), we make use of the following assumption:

Assumption I: Source sample set $\mathcal{S}$ contains the vertices of the bounding ℓ_1-norm ball $\mathcal{B}(S)$.

This deterministic assumption implies that for some sample instants only one source is assumed to be active and for each source there exists such sample instants. Therefore, Assumption I implies a *local dominance* condition for all sources.

Following theorem shows that, under the above local dominance assumption, all global maxima of $J(\mathbf{W})$ are perfect separators:

Theorem: Given the BCA setup in Section II, if the Assumption I is correct, then all global maxima of (18) are perfect separators for which

$$\mathbf{G} = \alpha\mathbf{P}, \tag{19}$$

where $\mathbf{P}$ is a permutation matrix and $\alpha \neq 0 \in \mathbb{R}$.

Proof: We start by writing the objective function, in terms of the argument $\mathbf{G} = \mathbf{W}\mathbf{H}$:

$$J(\mathbf{G}) = \frac{\sqrt{\det(\mathbf{G}\hat{\mathbf{R}}(\mathcal{S})\mathbf{G}^T)}}{\left(\max_{n\in\{1,\ldots,L\}}\|\mathbf{G}\mathbf{s}(n)\|_1\right)^p} = \frac{|\det(\mathbf{G})|\sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))}}{\left(\max_{n\in\{1,\ldots,L\}}\|\mathbf{G}\mathbf{s}(n)\|_1\right)^p}. \tag{20}$$

For the denominator of (20) we can write,

$$\left(\max_{n\in\{1,\ldots,L\}}\|\mathbf{G}\mathbf{s}(n)\|_1\right)^p \leq \|\mathbf{G}\|_{1,1}^p\|\mathbf{s}(n)\|_1^p \tag{21}$$

where $\|\mathbf{G}\|_{1,1}$ is the induced matrix norm as defined in the notation table at the end of Section I, which can be explicitly written as [31]

$$\|\mathbf{G}\|_{1,1} = \left\|\begin{bmatrix}\|\mathbf{G}_{:,1}\|_1 & \|\mathbf{G}_{:,2}\|_1 & \ldots & \|\mathbf{G}_{:,p}\|_1\end{bmatrix}\right\|_\infty. \tag{22}$$

Since $\mathbf{s}(n)$ is in the ℓ_1-norm ball $\mathcal{B}(\mathcal{S})$, given in (2), we can rewrite the inequality in (21) as

$$\left(\max_{n\in\{1,\ldots,L\}}\|\mathbf{G}\mathbf{s}(n)\|_1\right)^p \leq \|\mathbf{G}\|_{1,1}^p \tag{23}$$

If Assumption I holds, then (23) is an equality [31], therefore, we can rewrite (20) as

$$J(\mathbf{G}) = \frac{|\det(\mathbf{G})|\sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))}}{\left\|\begin{bmatrix}\|\mathbf{G}_{:,1}\|_1 & \|\mathbf{G}_{:,2}\|_1 & \ldots & \|\mathbf{G}_{:,p}\|_1\end{bmatrix}\right\|_\infty^p} \tag{24}$$

Based on this expression, we can write

$$J(\mathbf{G}) \leq \frac{|\det(\mathbf{G})|\sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))}}{\left(\left\|\begin{bmatrix}\|\mathbf{G}_{:,1}\|_1 & \|\mathbf{G}_{:,2}\|_1 & \ldots\|\mathbf{G}_{:,p}\|_1\end{bmatrix}\right\|_1/p\right)^p} \tag{25}$$

$$\leq \frac{|\det(\mathbf{G})|\sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))}}{\|\mathbf{G}_{:,1}\|_1\|\mathbf{G}_{:,2}\|_1\ldots\|\mathbf{G}_{:,p}\|_1} \tag{26}$$

$$\leq \frac{|\det(\mathbf{G})|\sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))}}{\|\mathbf{G}_{:,1}\|_2\|\mathbf{G}_{:,2}\|_2\ldots\|\mathbf{G}_{:,p}\|_2} \tag{27}$$

$$\leq \sqrt{\det(\hat{\mathbf{R}}(\mathcal{S}))} \tag{28}$$

where

- the inequality (25) is due to norm inequality (between ℓ_1 and ℓ_∞ norms), with equality if and only if all the columns of $\mathbf{G}$ has the same ℓ_1 norm,
- (26) is due to arithmetic-geometric mean inequality, with equality if and only if all the columns of $\mathbf{G}$ has the same ℓ_1 norm,
- (27) is due to norm inequality (between ℓ_1 and ℓ_2 norms), with equality if and only if each column of $\mathbf{G}$ has only one non-zero entry,
- (28) is due to Hadamard inequality, with equality if and only if all the columns are orthogonal to each other.

As a result, the upper bound for the objective $J(\mathbf{G})$ on the right hand-side of (28) is achieved if and only if $\mathbf{G} = \alpha\mathbf{P}$ where $\mathbf{P}$ is a permutation matrix and $\alpha \neq 0$. It can be shown that if the sources have varying ranges so that their samples lie in a weighted ℓ_1-norm ball, then due to the scaling indeterminacy of sources, the global optimizers would take the more general form $\mathbf{G} = \mathbf{PD}$.

C. Comparison of Sparse BCA to Independent Component Analysis with Sparsity Promoting Marginals

Taking the logarithm of the sparse BCA objective in (18) converts the ratio form into a difference form which can be written as a modified objective

$$\begin{aligned}\mathcal{J}(\mathbf{W}) &= \log(J(\mathbf{W})) \\ &= \frac{1}{2}\log(\det(\hat{\mathbf{R}}(\mathcal{Z}))) - p\log\left(\max_{n\in\{1,\ldots,L\}} \|\mathbf{z}(n)\|_1\right) \end{aligned} \quad (29)$$

For the sample covariance of the separator outputs, we can write

$$\hat{\mathbf{R}}(\mathcal{Z}) = \mathbf{W}\hat{\mathbf{R}}(\mathcal{Y})\mathbf{W}^T, \quad (30)$$

where $\hat{\mathbf{R}}(\mathcal{Y})$ is the sample covariance matrix of mixtures given by

$$\hat{\mathbf{R}}(\mathcal{Y}) = \frac{1}{L}\sum_{n=1}^{L}\mathbf{y}(n)\mathbf{y}(n)^T - \hat{\mu}(\mathcal{Y})\hat{\mu}(\mathcal{Y})^T, \quad (31)$$

with $\hat{\mu}(\mathcal{Y}) = \frac{1}{L}\sum_{n=1}^{L}\mathbf{y}(n)$. Therefore, in the square case $p = q$, we have $\det(\hat{\mathbf{R}}(\mathcal{Z})) = |\det(\mathbf{W})|^2\det(\hat{\mathbf{R}}(\mathcal{Y}))$, and the objective simplifies to

$$\mathcal{J}(\mathbf{W}) = \log(|\det(\mathbf{W})|) - p\log\left(\max_{n\in\{1,\ldots,L\}} \|\mathbf{z}(n)\|_1\right), \quad (32)$$

where we neglected the constant term $\frac{1}{2}\log(\det(\hat{\mathbf{R}}(\mathcal{Y})))$. The expression in (32) resembles the form of the maximum likelihood formulation

$$\mathcal{L}(\mathbf{W}) = \log(|\det(\mathbf{W})|) + \sum_{k=1}^{p}\sum_{n=1}^{L}\log(f_s(z_k(n))), \quad (33)$$

used for the ICA settings with independent sources and samples where $f_s(\cdot)$ is the presumed source marginal [7]. The outer summation in (33) reflects the independence of sources and the inner summation reflects the assumption about the independence of samples. When the marginal density is Laplacian, this likelihood expression simplifies to

$$\mathcal{L}(\mathbf{W}) = \log(|\det(\mathbf{W})|) - \sum_{n=1}^{L}\|\mathbf{z}(n)\|_1, \quad (34)$$

The ML expression in (33) can be further generalized to quasi-ML expression (e.g. [28])

$$\mathcal{L}(\mathbf{W}) = \log(|\det(\mathbf{W})|) + \sum_{k=1}^{p}\sum_{n=1}^{L}v(z_k(n)), \quad (35)$$

where v is not necessarily derived from $\log(f_s(\cdot))$.

The likelihood based expressions in (33) and (35) (and therefore in (34)) assume the independence of sources as well as which is reflected by the summations over source components and samples. *However, as an important difference, the deterministic sparse BCA framework based on the objective in (32) (and more generally in (18) or (29)) does not make such independence assumptions*. Therefore, the proposed sparse BCA scheme is applicable to dependent sources with non-separable joint densities.

D. Iterative Algorithm for Sparse BCA

The modified sparse BCA objective $\mathcal{J}(\mathbf{W})$ in (29) is also convenient for the iterative algorithm derivation, due to its additive form. Although $\mathcal{J}(\mathbf{W})$ is non-convex and not differentiable everywhere, we can still utilize Clarke subdifferential [32] for deriving iterative algorithms.

The objective function $\mathcal{J}(\mathbf{W})$ is composed of two terms:

$$\mathcal{J}(\mathbf{W}) = \quad (36)$$

$$= \underbrace{\frac{1}{2}\log(\det(\hat{\mathbf{R}}(\mathcal{Z})))}_{\mathcal{J}_1(\mathbf{W})} - p\log\left(\underbrace{\max_{n\in\{1,\ldots,L\}}\|\mathbf{z}(n)\|_1}_{\mathcal{J}_2(\mathbf{W})}\right). \quad (37)$$

$\mathcal{J}_1(\mathbf{W})$ in the first term is a convex differentiable function, and $\mathcal{J}_2(\mathbf{W})$ in the second term is a convex non-smooth function. In deriving the iterative update rule for maximizing $\mathcal{J}(\mathbf{W})$, we use the gradient term for $\mathcal{J}_1(\mathbf{W})$ and the subgradient term for the $\mathcal{J}_2(\mathbf{W})$:

- *Gradient Term for $\mathcal{J}_1(\mathbf{W})$*: The gradient term to increase the volume of principal hyper-ellipsoid $\mathcal{E}(\mathcal{Z})$ can be written as

$$\nabla\mathcal{J}_1(\mathbf{W}) = \left(\mathbf{W}\hat{\mathbf{R}}(\mathcal{Y})\mathbf{W}^T\right)^{-1}\mathbf{W}\hat{\mathbf{R}}(\mathcal{Y}). \quad (38)$$

- Subdifferential Set for $\mathcal{J}_2(\mathbf{W})$: If we write down $\mathcal{J}_2(\mathbf{W})$ in the form

$$\mathcal{J}_2(\mathbf{W}) = \max_{n\in\{1,\ldots,L\}} f_n(\mathbf{W}) \quad (39)$$

where $f_n(\mathbf{W}) = \|\mathbf{W}\mathbf{y}(n)\|_1$, then the subdifferential set for $\mathcal{J}_2$ can be written as [33],

$$\partial\mathcal{J}_2(\mathbf{W}) = \mathbf{Co}\cup\{\partial f_i(\mathbf{W}) : f_i(\mathbf{W}) = \mathcal{J}_2(\mathbf{W})\}. \quad (40)$$

Now, let us calculate the derivative of the map $\mathbf{v} \rightarrow \|\mathbf{v}\|_1$. For all $\mathbf{v} = \begin{bmatrix} v_1 & \cdots & v_N \end{bmatrix} \in \mathbb{R}^N$, one has $\|\mathbf{v}\|_1 = |v_1| + |v_2| + \cdots + |v_N|$. Therefore,

$$\frac{\partial}{\partial v_j}(\mathbf{v} \rightarrow \|\mathbf{v}\|_1) = \frac{v_j}{|v_j|} = sign(v_j), \quad 1 \leq j \leq N \quad (41)$$

provided that $v_j \neq 0$. If $v_j = 0$, then we have

$$\frac{\partial}{\partial v_j}(\mathbf{v} \rightarrow \|\mathbf{v}\|_1) = \alpha_j \in [-1, 1] \quad (42)$$

Hence, the subdifferential set for $f_n(\mathbf{W})$ is given by

$$\partial f_n(\mathbf{W}) = \partial(\| \begin{bmatrix} \mathbf{W}_{1,:}\mathbf{y}(n) & \cdots & \mathbf{W}_{p,:}\mathbf{y}(n) \end{bmatrix}^T \|_1) \quad (43)$$

$$= \| \begin{bmatrix} \frac{\partial(\mathbf{W}_{1,:}\mathbf{y}(n))}{\partial(\mathbf{W}_{1,:})} & \cdots & \frac{\partial(\mathbf{W}_{p,:})\mathbf{y}(n))}{\partial(\mathbf{W}_{p,:})} \end{bmatrix}^T \|_1 \quad (44)$$

$$= \{\mathbf{q}\mathbf{y}(n)^T : q_i = \text{sign}\{z_i(n)\} + 1_{z_i(n)=0}\alpha_i, \; \alpha_i \in [-1, 1]\} \quad (45)$$

Note that $\text{sign}\{\mathbf{z}(n)\}\mathbf{y}(n)^T$ is a subgradient, a member of $\partial f_n(\mathbf{W})$. Selecting this particular subgradient, a subgradient choice for $\mathcal{J}_2(\mathbf{W})$ can be written as

$$\partial \mathcal{J}_2(\mathbf{W}) = \sum_{n \in \mathcal{I}_{\mathbf{W}}} \lambda_n \text{sign}\{\mathbf{z}(n)\}\mathbf{y}(n)^T \quad (46)$$

where $\{\mathcal{I}_W = \{n : \|\mathbf{W}\mathbf{y}(n)\|_1 = \mathcal{J}_2(\mathbf{W})\}$ is the set of sample indices for which maximum ℓ_1-norm separator output is achieved, and λ_n's are convex combination coefficients satisfying $\lambda_n \geq 0$ and $\sum_{n \in \mathcal{I}_{\mathbf{W}}} \lambda_n = 1$.

Based on the gradient expression in (38) and the subgradient choice in (46), an iterative update rule for maximizing the SBCA objective $\mathcal{J}(\mathbf{W})$ can be written as

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}((\hat{\mathbf{R}}(\mathcal{Z}^{(t)}))^{-1}\mathbf{W}^{(t)}\hat{\mathbf{R}}(\mathcal{Y}) - \frac{p}{\max_{n \in \{1,\ldots,L\}} \|\mathbf{z}^{(t)}(n)\|_1} \sum_{l \in \mathcal{I}_{\mathbf{W}^{(t)}}} \lambda_l^{(t)} \text{sign}\{\mathbf{z}^{(t)}(l)\}\mathbf{y}(l)^T, \quad (47)$$

where t is the iteration index. As a special case, if we choose only one of the convex combination coefficients to be non-zero, which corresponds to choosing an arbitrary index location $l^{(t)}$ from $\mathcal{I}_{\mathbf{W}^{(t)}}$ at every iteration, then the update rule simplifies to

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}((\hat{\mathbf{R}}(\mathcal{Z}^{(t)}))^{-1}\mathbf{W}^{(t)}\hat{\mathbf{R}}(\mathcal{Y}) - \frac{p}{\max_{n \in \{1,\ldots,L\}} \|\mathbf{z}^{(t)}(n)\|_1} \text{sign}\{\mathbf{z}^{(t)}(l^{(t)})\}\mathbf{y}(l^{(t)})^T. \quad (48)$$

IV. Algorithmic Extensions for SBCA

This part of the article provides algorithmic extensions for the Sparse BCA framework introduced in Section III. In Section IV-A, we provide the extension of the SBCA iterative algorithm for the case of complex signals. The algorithm modification for the noisy scenario is introduced in Section IV-B.

A. Extension to Complex Signals

In order to extend the algorithms for the complex signals, we follow the isomorphism based approach used in [2]. For this purpose, we define the operator $\Upsilon : \mathbb{C}^p \rightarrow \mathbb{R}^{2p}$

$$\Upsilon(\mathbf{x}) = \begin{bmatrix} \mathbb{R}e\{\mathbf{x}^T\} & \mathcal{I}m\{\mathbf{x}^T\} \end{bmatrix}^T \quad (49)$$

as an isomorphism between the p dimensional complex vectors and $2p$ dimensional real vectors. For a given complex vector $\mathbf{x}$, we use the notation $\dot{\mathbf{x}}$ to refer its real isomorphic vector, i.e., $\dot{\mathbf{x}} = \Upsilon(\mathbf{x})$. Similarly, we also define $\Gamma : \mathbb{C}^{p \times q} \rightarrow \mathbb{R}^{2p \times 2q}$

$$\Gamma(\mathbf{A}) = \begin{bmatrix} \mathbb{R}e\{\mathbf{A}\} & -\mathcal{I}m\{\mathbf{A}\} \\ \mathcal{I}m\{\mathbf{A}\} & \mathbb{R}e\{\mathbf{A}\}, \end{bmatrix} \quad (50)$$

for mapping complex matrices to real matrices, preserving the complex matrix-vector multiplication operation for the real. Based on these definitions, we can write

$$\dot{\mathbf{y}}(n) = \Gamma(\mathbf{H})\dot{\mathbf{s}}(n), \quad (51)$$

$$\dot{\mathbf{z}}(n) = \Gamma(\mathbf{W})\dot{\mathbf{y}}(n). \quad (52)$$

We use the notation $\dot{\mathcal{Y}} = \{\dot{\mathbf{y}} : \mathbf{y} \in \mathcal{Y}\}$ and $\dot{\mathcal{Z}} = \{\dot{\mathbf{z}} : \mathbf{z} \in \mathcal{Z}\}$ for the real-isomorphic mixture and separator output vector sets respectively.

We can obtain algorithm corresponding to the complex SBCA setting by defining the volume ratio objective in the $2p$ dimensional isomorphic real space:

$$J_c(\mathbf{W}) = \frac{\sqrt{\det(\hat{\mathbf{R}}(\dot{\mathcal{Z}}))}}{\left(\max_{n \in \{1,\ldots,L\}} \|\dot{\mathbf{z}}(n)\|_1\right)^{2p}}. \quad (53)$$

Using algebraic manipulations similar to [2], the corresponding iterative update equation for $\mathbf{W}$ can be written as

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}(\mathbf{W}_{logdet}^{(t)} - \mathbf{W}_{subg}^{(t)}) \quad (54)$$

where

$$\mathbf{W}_{logdet}^{(t)} = \frac{1}{2}\Bigg(\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}\hat{\mathbf{R}}(\dot{\mathcal{Z}}^{(t)})^{-1}\Gamma(\mathbf{W}^{(t)})\hat{\mathbf{R}}(\dot{\mathcal{Y}})\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}\hat{\mathbf{R}}(\dot{\mathcal{Z}}^{(t)})^{-1}\Gamma(\mathbf{W}^{(\mathbf{t})})\hat{\mathbf{R}}(\dot{\mathcal{Y}})\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} + j\Bigg(\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}\hat{\mathbf{R}}(\dot{\mathcal{Z}}^{(\mathbf{t})})^{-1}\Gamma(\mathbf{W}^{(\mathbf{t})})\hat{\mathbf{R}}(\dot{\mathcal{Y}})\begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}\hat{\mathbf{R}}(\dot{\mathcal{Z}}^{(t)})^{-1}\Gamma(\mathbf{W}^{(t)})\hat{\mathbf{R}}(\dot{\mathcal{Y}})\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}\Bigg)\Bigg) \quad (55)$$

and

$$\mathbf{W}_{subg}^{(t)} = \frac{2p}{\max\limits_{n \in \{1,\ldots,L\}} \|\dot{\mathbf{z}}^{(t)}(n)\|_1} \text{signc}\{\mathbf{z}^{(t)}(l^{(t)})\}\mathbf{y}(l^{(t)})^H \quad (56)$$

As an alternative, we can define the cost function,

$$J_{ca}(\mathbf{W}) = \frac{\sqrt{\det(\hat{\mathbf{R}}(\mathcal{Z}))}}{\left(\max_{n \in \{1,\ldots,L\}} \|\dot{\mathbf{z}}(n)\|_1\right)^{p}}. \quad (57)$$

where the only change is in the covariance term in the numerator, based on the equivalence $|\det(\Gamma(\mathbf{G}))| = |\det(\mathbf{G})|^2$ [2] and the degree of the denominator. In this case, the update terms simplify to

$$\mathbf{W}_{logdet}^{(t)} = \hat{\mathbf{R}}(\mathcal{Z}^{(t)})^{-1}\mathbf{W}^{(t)}\hat{\mathbf{R}}(\mathcal{Y}), \tag{58}$$

$$\mathbf{W}_{subg}^{(t)} = \frac{p}{\max\limits_{n\in\{1,\ldots,L\}} \|\dot{\mathbf{z}}^{(t)}(n)\|_1}\text{signc}\{\mathbf{z}^{(t)}(l^{(t)})\}\mathbf{y}(l^{(t)})^H \tag{59}$$

B. Iterative Algorithm for Noisy Case

When we replace the mixture model with its noisy version, i.e.,

$$\begin{aligned}\mathbf{y}(k) &= \mathbf{H}\mathbf{s}(k) + \mathbf{n}(k)\\ &= \mathbf{y}_{noiseless}(k) + \mathbf{n}(k)\end{aligned} \tag{60}$$

where $\mathbf{n}(k)$ corresponds to random noise sequence, the corresponding noisy separator output could be written as

$$\begin{aligned}\mathbf{z} &= \mathbf{W}\mathbf{y}_{noiseless}(k) + \mathbf{W}\mathbf{n}(k)\\ &= \mathbf{z}_{noiseless}(k) + \mathbf{v}(k).\end{aligned} \tag{61}$$

Here the main difficulty would be to identify the set $\{\mathcal{I}_W = \{n : \|\mathbf{z}_{noiseless}(n)\|_1 = \max_{k\in 1,\ldots,L}\|\mathbf{z}_{noiseless}(k)\|_1\}$, i.e., the peak locations for the ℓ_1-norm of the noiseless separator outputs $\mathbf{z}_{noiseless}(k)$. Due to the presence of noise, the determination of these index locations from the noisy separator outputs $\mathbf{z}(k)$ becomes a stochastic estimation problem. In such a scenario, we assign certain probability to each index in $\{1,\ldots,L\}$. More explicitly, let $p_{\mathcal{I}_\mathbf{W}}(k)$ denote the probability mass function (pmf) for index k to be in $\mathcal{I}_\mathbf{W}$, i.e.,

$$p_{\mathcal{I}_\mathbf{W}}(k) = \Pr(k \in \mathcal{I}_\mathbf{W}), \tag{62}$$

we can replace the update rule in (47) with

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}(\hat{\mathbf{R}}(\mathcal{Z}^{(t)})^{-1}\mathbf{W}^{(t)}\hat{\mathbf{R}}(\mathcal{Y}) - \frac{p}{\max_{n\in\{1,\ldots,L\}}\|\mathbf{z}^{(t)}(n)\|_1}\sum_{l=1}^{L} p_{\mathcal{I}_\mathbf{W}^{(t)}}(l)\text{sign}\{\mathbf{z}^{(t)}(l)\}\mathbf{y}(l)^T, \tag{63}$$

where the convex combination coefficients $\lambda^{(t)}$ in (47) are replaced by the pmf $p_{\mathcal{I}_{\mathbf{W}^{(t)}}}$ in (63). In this new form, we essentially calculate the expected $\mathbf{W}_{subg}^{(t)}$ term by weighted averaging of the contributions of all index points.

Obtaining expression for the exact form of $p_{\mathcal{I}_\mathbf{W}}(k)$ is a cumbersome process, even for Gaussian noise scenario. Instead, we use a simplified approximation for this pmf. We first define an estimate of the set $\mathcal{I}_\mathbf{W}$ as

$$\hat{\mathcal{I}}_\mathbf{W} = \{n : \|\mathbf{z}(n)\|_1 \geq \beta \times \max_{k\in\{1,\ldots,L\}}\|\mathbf{z}(k)\|_1, \tag{64}$$

where $0 < \beta \leq 1$ is an algorithm parameter. This choice of index set estimate corresponds to selecting index points, whose ℓ_1-norm of the noisy separator output has value at some neighborhood of the maximum ℓ_1-norm for the whole noisy separator outputs. Based on this index set estimate, we define the pmf estimate as

$$\hat{p}_{\mathcal{I}_\mathbf{W}}(k) = \begin{cases}\frac{1}{|\hat{\mathcal{I}}_\mathbf{W}|} & k \in \hat{\mathcal{I}}_\mathbf{W},\\ 0 & \text{otherwise.}\end{cases} \tag{65}$$

Note that this choice corresponds to a uniform density over the selected index set. Instead, we could potentially select the probability $\hat{p}_{\mathcal{I}_\mathbf{W}}(k)$ in proportion to the corresponding $\|\mathbf{z}(k)\|_1$ value. Although, more accurate estimates of $\hat{p}_{\mathcal{I}_\mathbf{W}}(k)$ potentially are more desirable, the estimate in (65) yields satisfactory results as illustrated by the numerical experiments in Section VI.

C. Algorithm Extension with Sparsifying Transformations

In various applications, the signals of interest are not "naturally" sparse, but they may be converted to sparse form through an appropriate linear transformation, i.e. a basis change. Let the data snapshot matrices for source and mixtures be defined as

$$\mathbf{S} = \begin{bmatrix}\mathbf{s}(1) & \mathbf{s}(2) & \ldots & \mathbf{s}(L)\end{bmatrix}, \tag{66}$$

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}(1) & \mathbf{y}(2) & \ldots & \mathbf{y}(L)\end{bmatrix}, \tag{67}$$

respectively. Clearly, we have $\mathbf{Y} = \mathbf{H}\mathbf{S}$.

Given the sparsifying transformation matrix $\mathbf{\Phi} \in \mathbb{R}^{L\times L}$, we can write the sources in transform domain as

$$\mathbf{S}_T = \mathbf{S}\mathbf{\Phi}. \tag{68}$$

It is desired that the applied transformation results in an $\mathbf{S}_T$ with sparse entries. Of course, in the application, the transformation can only be applied to mixtures in the form

$$\mathbf{Y}_T = \mathbf{Y}\mathbf{\Phi}. \tag{69}$$

Substituting $\mathbf{Y} = \mathbf{H}\mathbf{S}$,

$$\begin{aligned}\mathbf{Y}_T &= \mathbf{H}\mathbf{S}\mathbf{\Phi}, &(70)\\ &= \mathbf{H}\mathbf{S}_T. &(71)\end{aligned}$$

Therefore, the transformed mixtures are equivalent to the mixtures of transformed sources with the same mixing matrix. As a result, the proposed SBCA approach can be applied to transformed mixtures $\mathbf{Y}_T$ instead of the original mixtures. In Section VI-B, we provide an example of this procedure, where Morlet transformation is applied to image patches to convert them into a sparse form.

V. Algorithm Acceleration for SBCA

In this section, some recipes for accelerating SBCA algorithms are proposed. For this purpose, we offer two major improvements: the first one is the use of natural gradient based weighted update which particularly simplifies the update expression. The second extension is the use of accelerated gradient methods to improve the convergence speed.

A. Weighted Update

In BSS literature, natural gradient based learning rules attracted particular attention due to its improved convergence behavior [34]. Following this approach, the update components in SBCA algorithm could be weighted by the positive matrix $\mathbf{W}^T\mathbf{W}$, which would yield

$$\begin{aligned}\mathbf{W}_{logdet}\mathbf{W}^T\mathbf{W} &= \left(\left(\mathbf{W}\hat{\mathbf{R}}(\mathcal{Y})\mathbf{W}^T\right)^{-1}\mathbf{W}\hat{\mathbf{R}}(\mathcal{Y})\right)\mathbf{W}^T\mathbf{W}\\ &= \mathbf{W}\end{aligned} \tag{72}$$

and

$$\mathbf{W}_{subg}\mathbf{W}^T\mathbf{W} = \frac{p}{\max\limits_{n\in\{1,\ldots,L\}}\|\mathbf{z}(n)\|_1}\text{sign}\{\mathbf{z}(l)\}\mathbf{y}(l)^T\mathbf{W}^T\mathbf{W}$$
$$= \frac{p}{\max\limits_{n\in\{1,\ldots,L\}}\|\mathbf{z}(n)\|_1}\text{sign}\{\mathbf{z}(l)\}\mathbf{z}(l)^T\mathbf{W} \quad (73)$$

As a result, SBCA update rule in (48) can be replaced with

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}\left(\mathbf{I} - \frac{p}{\max\limits_{n\in\{1,\ldots,L\}}\|\mathbf{z}^{(t)}(n)\|_1}\text{sign}\{\mathbf{z}^{(t)}(l^{(t)})\}\mathbf{z}^{(t)}(l^{(t)})^T\right)\mathbf{W}^{(t)}. \quad (74)$$

We note that this form interestingly resembles the update rule for Infomax (ICA) algorithm for supergaussian sources:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}(\mathbf{I} - 2\tanh\{\mathbf{z}^{(t)}(l^{(t)})\}\mathbf{z}^{(t)}(l^{(t)})^T)\mathbf{W}^{(t)}, \quad (75)$$

where the main difference is that the update rule in (74) is only applied to indices corresponding to peak (or near-peak in the noisy case) ℓ_1-norm separator outputs for that iteration and the update rule for Extended Infomax is sequentially applied to all sample points, i.e., $l^{(t+1)} = l^{(t)} + 1$.

B. Nesterov Update Rule

In optimization algorithms, Nesterov's acceleration method is a common approach to improve the speed of convergence [35]. This approach is originally proposed for smooth and strongly convex functions, however, in our work we apply it to a non-smooth and non-convex function. While the underlying theory for the justification of the original acceleration approach is still an area of active research [36], our motivation for the application of this rule is based on the empirical observation that this update rule significantly improves the speed of convergence when applied to SBCA updates.

If we represent the original SBCA algorithm in the form

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mu^{(t)}\mathbf{U}^{(t)} \quad (76)$$

where the update term $\mathbf{U}^{(t)}$ can be replaced with any of the update terms proposed in the previous sections, the Nesterov acceleration based update rule can be written as

$$\mathbf{X}^{(t+1)} = \mathbf{W}^{(t)} + \upsilon^{(t)}\mathbf{U}^{(t)} \quad (77)$$
$$\mathbf{W}^{(t+1)} = \mathbf{X}^{(t+1)} + \kappa^{(t)}(\mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}) \quad (78)$$

where $\mathbf{X}^{(t)}$ is an intermediate algorithm variable, $\kappa^{(t)} = \frac{t-1}{t+2}$ and $\upsilon^{(t)}$ is the algorithm parameter to be selected. In this formulation, the update rule essentially has memory of previous update expressions represented by the first expression.

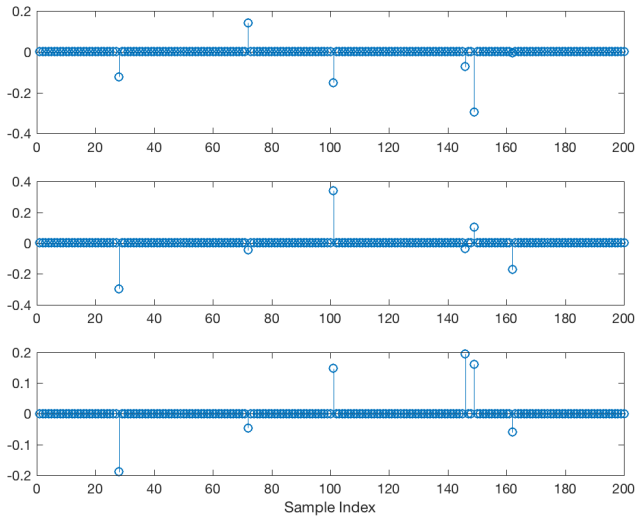

Fig. 2: Illustration of the synthetically generated random sparse sequences of Example 1.

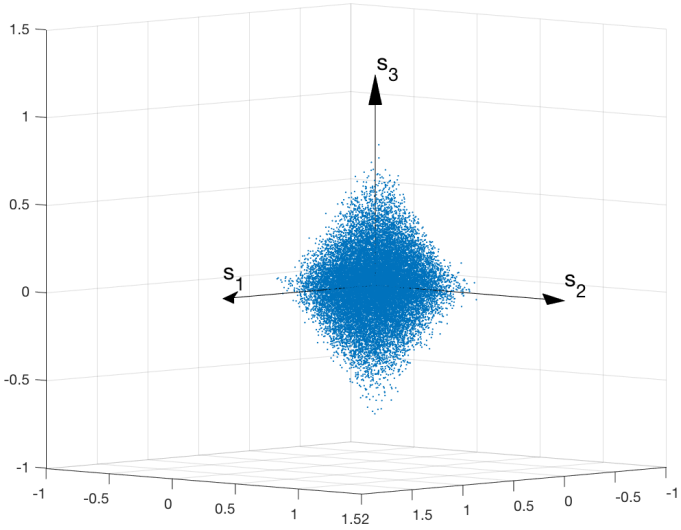

Fig. 3: Scatter diagram for the random sparse sources of Example 1.

VI. Numerical Experiments

In the experiments, we measure the Signal to (residual) Interference and Noise power Ratio (SINR) for the separator outputs after the convergence of the algorithm. SINR term is mathematically defined as,

$$SINR = \frac{P}{I+N} \quad (79)$$

where $P = \| \left[\mathbf{G}(m_1,1), \quad \cdots \quad, \mathbf{G}(m_p,p)\right] \|_2^2$ represents the signal power, $I = \| \left[\mathbf{G}(m_1,:)^T, \quad \cdots \quad, \mathbf{G}(m_p,:)^T\right]^T \|_F^2 - \| \left[\mathbf{G}(m_1,1), \quad \cdots \quad, \mathbf{G}(m_p,p)\right] \|_2^2$ represents the interference power, $N = \sigma_{noise}\|\mathbf{W}\|_F^2$ represents the noise power, and the indices $\{m_1, \cdots, m_p\}$ represent the maximum locations in the columns of the system response matrix $\mathbf{G}$.

We compare the proposed algorithm's SINR versus Sample Size (L) performance to Infomax [21], Relative Newton [28], Generalized Morphological Component Analysis (GMCA) [37], Bounded Component Analysis (BCA) methods in [1] and [2], and Nonnegative Least-Correlated Component Analysis (nLCA) [38] (where we used the publicly available codes for these algorithms as noted in the corresponding references).

A. Synthetic Sparse Sources

In the first numerical example, we synthetically generate sparse sources by transforming i.i.d. uniform random vector $\mathbf{u} \in [-1,1]^p$, through the mapping

$$\mathbf{s} = \begin{cases} \mathbf{u} & \mathbf{u} \in \mathcal{B}_r \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{80}$$

where $\mathcal{B}_r = \{\mathbf{x} : \|\mathbf{x}\|_r \leq 1\}$ with $0 < r \leq 1$. Fig. 2 illustrates the discrete time sequences for $p = 3$ sources generated through this mapping with $r = 0.8$, and Fig.3 shows their corresponding scatter diagram. These sources are mixed through a random mixing matrix $\mathbf{H}$ with i.i.d. Gaussian entries (with zero mean and unit variance). The mixtures are also perturbed by i.i.d. Gaussian noise. As a special case, for

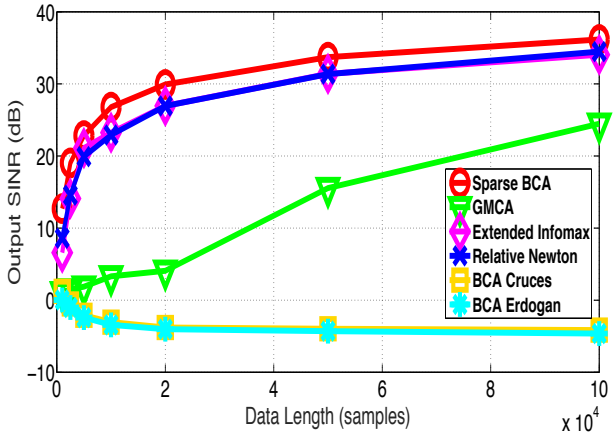

Fig. 4: Output SINR vs.Data Length for Example 1, when the random variables used in generation process are independent.

$p = 4$ sources, $q = 8$ mixtures and $r = 0.8$, the corresponding SINR vs Data Length curves are shown in Fig. 4. Based on this figure, we can comment that the Sparse BCA algorithm provides a performance improvement in the range $3 - 5$dB. There are no performance results for nLCA-IVM since nLCA-IVM assumes that number of mixtures is equal to number of sources, and sources are non-negative while we use 8 mixtures for 4 sources which can have negative values as well. We note that BCA approaches in [1] and [2] perform worse, as they assume the sources to be contained in ℓ_∞-norm ball.

As a modified experiment, if we use dependent uniform variables in the source generation process of (80) based on a Copula distribution with four degrees of freedom and a correlation constant $\rho = 0.7$, we obtain the curves in Fig. 5. The dependency introduced in the source generation process appears to cause leveling in the performances of Extended Infomax and Relative Newton algorithms, which emphasizes

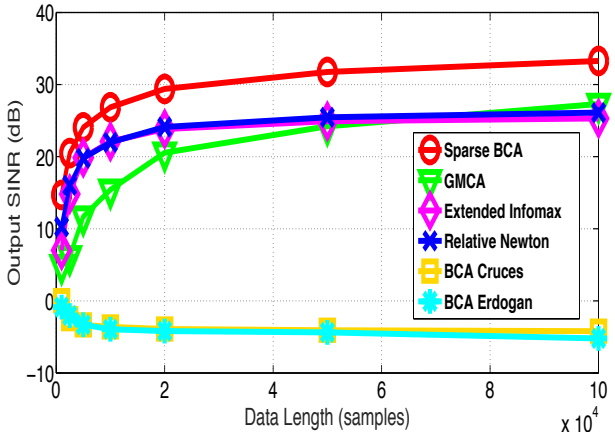

Fig. 5: Output SINR vs.Data Length for Example 1, when the random variables used in generation process are correlated.

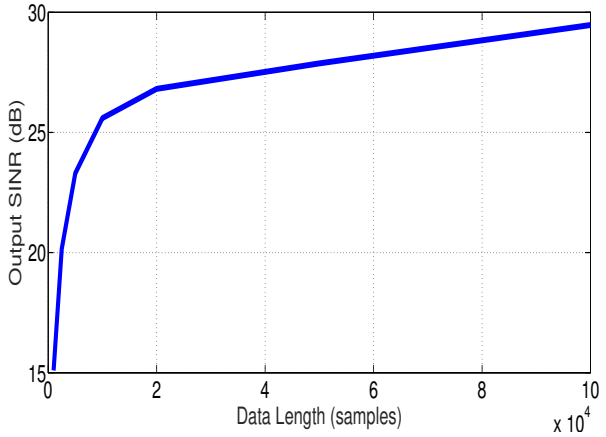

Fig. 6: Performance of complex SBCA, when the random variables used in generation process are correlated.

the performance gain of the Sparse BCA Algorithm, especially for larger data sizes.

In order to evaluate the performance of Sparse BCA's complex extension, we again use the Copula distibuted synthetic sources. We generate complex sources for complex Sparse BCA by multiplying the Copula distributed signals with independent and unit magnitude complex signals with uniformly distributed (in $[-\pi, \pi]$) phase terms. As seen in Fig. 6, the proposed SBCA framework successfully works for complex sparse sources as well.

B. Natural Images

In the second example, three 204x204 pixel images are mixed by a random Gaussian 3×3 matrix and corrupted by Gaussian noise. In Fig. 7, the input and mixed images for the proposed algorithm are illustrated for a random mixing matrix. As the first stage of separation, we apply Morlet wavelet transform [39] to the mixtures for the sparsification, as proposed in Section IV-C. On the other hand, we don't apply Morlet wavelet transform to the inputs of non-sparse BCA methods in [1], [2], and nLCA-IVM method since they

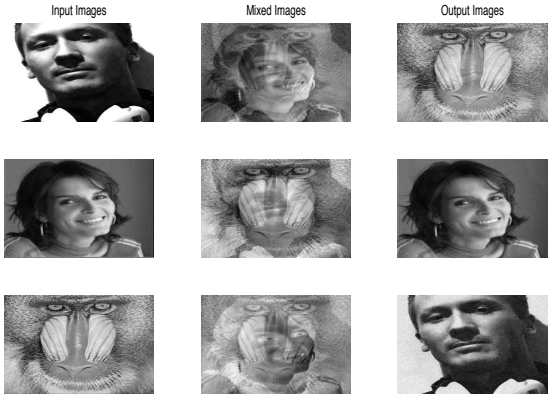

Fig. 7: Image Separation Example - Original, noisy mixed and output images for SBCA (SNR=20 dB).

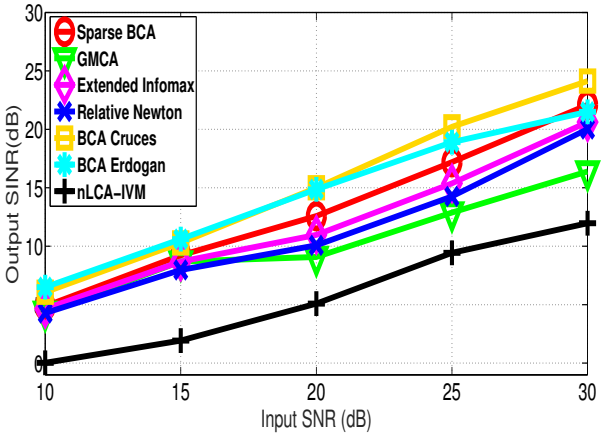

Fig. 9: Image Separation Example - Output SINR vs Input SNR.

were designed for non-sparse sources. In order to obtain sparse source images, we divide the mixtures into 8×8 patches and apply the transform to these patches. The scatter plot of the Morlet transformation coefficients for the three images can be seen in Fig. 8, which appears to conform the assumptions made in the SBCA setup about the source samples. After this transformation, we apply the same algorithms in the first example to the transformed mixtures. Fig. 9 shows the mean output SINR vs. input SNR performance for each algorithm (averaged over random Gaussian 3×3 mixture matrices) which illustrates that SBCA outperforms the other sparse methods at all SNRs. SBCA do not exceed the non-sparse BCA methods' performance because they use natural images directly while SBCA needs the images to be sparsified which may cause information loss. Also, comparison of the maximum output SINRs is provided since Cruces' BCA method extracts only one source as the output while the other methods extract all sources so that the comparison in Fig. 9 is not fair. As can be seen in Fig. 10, maximum output SINRs of all the other methods but nLCA-IVM exceed that of Cruces' BCA.

To evaluate the acceleration performance of Nesterov Update method explained in section V, we compare SINR change rates of the update methods with respect to iteration number. In the analysis, the images are mixed by a random Gaussian 6×3 matrix, i.e., there are 6 mixtures. The mixtures are corrupted by Gaussian noise with respect to input SNR=30 dB. As can been seen in Fig.11, this study yields that Nesterov Update method get Sparse BCA to converge to the maximum output SINR much faster than ordinary update method does.

C. Neuroimaging

As the last example, we consider the problem of identifying neurons and their activities from a calcium (Ca^{2+}) imaging based video recording. In calcium imaging, temporal activations of neurons are reflected as fluorescent emissions and recorded via a digital camera. Since neurons can be active simultaneously, their activity signals are mixed which yields a typical mixing problem. Neural activities denote the time courses of the neurons' relative strength, i.e., temporal weights. Multiplying the calcium imaging data by the estimated separator matrix gives a data matrix whose rows present the neural activities. The columns of the inverse

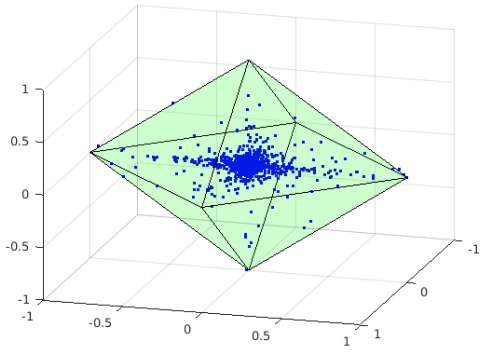

Fig. 8: Image Separation Example - Scatter plot (with bounding ℓ_1-norm ball) of the transform coefficients corresponding to the 3 input images.

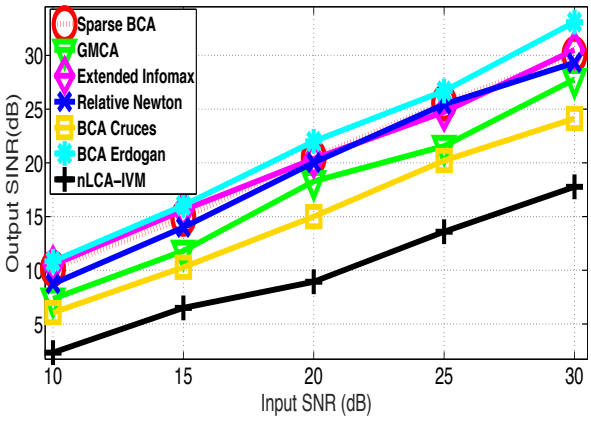

Fig. 10: Image Separation Example - Maximum Output SINR vs Input SNR.

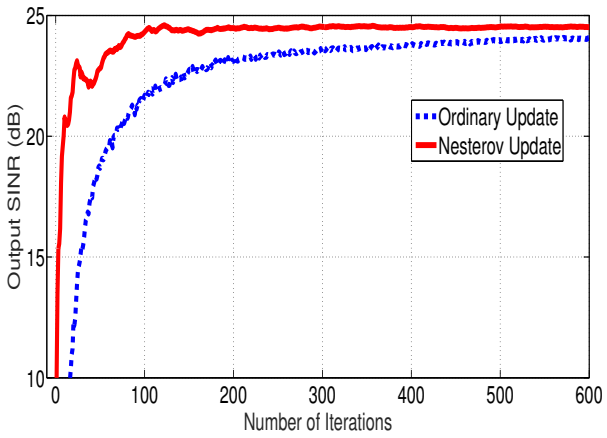

Fig. 11: Image Separation Example - SBCA with Nesterov Update vs SBCA with Ordinary Update.

separator matrix give the corresponding neurons generating the activities. Since only a small group of the neurons are active at one time, the problem conforms the sparsity assumption.

In this example, we used the video recording provided as a supplementary for the article [40], where the CA1 hippocampal place cell activity of the freely moving mouse is studied (This movie file is available in https://www.nature.com/neuro/journal/v16/n3/full/nn.3329.html.) As the main process-

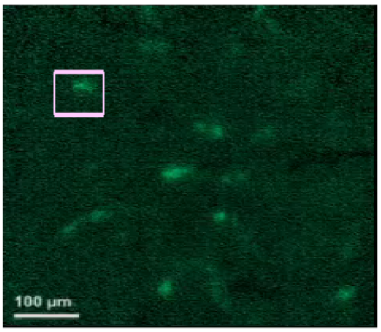

Fig. 12: A frame of the Ca^{2+} imaging movie of [40] (Use by the permission of Nature Publishing Group). In the sam figure, a rectangular subwindow is also marked.

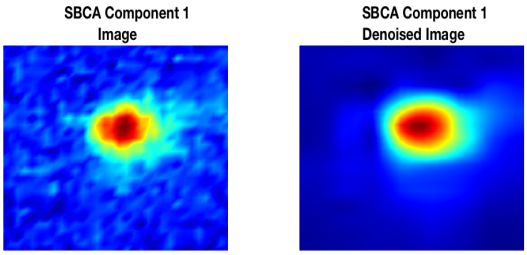

Fig. 13: A component map obtained by applying SBCA algorithm to the subwindow marked in Fig. 12, and wavelet denoised version of the component map.

ing step, we divide each frame into 32×32 subwindows. Fig. 12 illustrates a frame from the movie and a sample subwindow. Let's assume that the corresponding movie can be represented by a three dimensional array $R[m, n, f]$, where $(m, n, f) \in \{1, \ldots, M\} \times \{1, \ldots, N\} \times \{1, \ldots, F\}$, where m, n are pixel location indices, row and column respectively, f is the frame index, M (N) is the number of rows (columns) for each frame, and F is the number of frames. Then for each $K \times K$ sub-movie can be represented as

$$R_{(i,j)}[m, n, f] = R[(i-1)S + m, (j-1)S + n, f]$$

where $(i, j) \in \{1, \ldots, \lfloor \frac{M-K}{S} \rfloor + 1\} \times \{1, \ldots, \lfloor \frac{N-K}{S} \rfloor + 1\}$, and $m, n \in \{1, \ldots, K\}$.

The vectorized version of each subwindow of a frame constitutes observation vector for the SBCA algorithm (The vectorization is performed through the *vec* operator which stacks the columns of the matrix corresponding to the subwindow image):

$$\mathbf{Y}_{(i,j)} = \left[\ \mathrm{vec}(R_{(i,j)}[:, :, 1]), \ldots, \mathrm{vec}(R_{(i,j)}[:, :, F]) \ \right].$$

In this recording, the number of observations (frames) is equal to $F = 1339$. SBCA algorithm is applied to each subwindow. Fig. 13 shows the component map obtained by converting a column of the pseudoinverse of $\mathbf{W}$ to 32×32 image. We apply wavelet denoising based on (Symlet wavelet) using *wdencmp* function of MATLAB, using 'sym4' as the wavelet choice. The resulting denoised component map is shown in Fig. 13 also.

In Fig. 14, sample denoised component maps and their activations are shown. The activation structures indeed confirm the sparsity property exploited by the SBCA algorithm. The presented component analysis approach can be further developed into a full neuron-sorting algorithm registering neuron locations and their activations for the full movie [41].

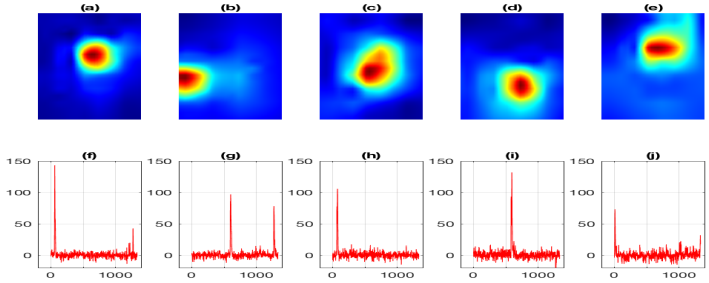

Fig. 14: Some SBCA (denoised) component maps corresponding to the subwindow in Fig. 12 (in the first row (a)-(e)), and their activations (in the second row (f)-(i))

VII. Conclusion

In this article, we proposed a deterministic Bounded Component Analysis approach for sparse-bounded sources. The resulting framework is the extension of the algorithmic BCA framework in [2] to sparse case. As in the case of [2], no assumption is made about the independence of sources or samples. All the global maxima of the corresponding geometric optimization setting are proven to be perfect separators, and the update rule derived from this setting is shown to resemble the form of the Infomax algorithm for supergaussian sources. The numerical examples demonstrate the potential practical merit of the proposed framework.

REFERENCES

[1] S. Cruces, "Bounded component analysis of linear mixtures: A criterion for minimum convex perimeter," *IEEE Trans. on Signal Process.*, vol. 58, no. 4, pp. 2141–2154, April 2010.

[2] A.T. Erdogan, "A class of bounded component analysis algorithms for the separation of both independent and dependent sources," *Signal Processing, IEEE Transactions on*, vol. 61, no. 22, pp. 5730–5743, 2013.

[3] P. Aguilera, S. Cruces, I. Duran, A. Sarmiento, and D.P. Mandic, "Blind separation of dependent sources with a bounded component analysis deflationary algorithm," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 709–712, 2013.

[4] H.A. Inan and A.T. Erdogan, "Convolutive bounded component analysis algorithms for independent and dependent source separation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 697–708, April 2015.

[5] H.A. Inan and A.T. Erdogan, "A convolutive bounded component analysis framework for potentially nonstationary independent and/or dependent sources," *Signal Processing, IEEE Transactions on*, vol. 63, no. 1, pp. 18–30, Jan 2015.

[6] Christian Jutten and Jeanny Herault, "Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1 – 10, 1991.

[7] Pierre Comon and Christian Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, 2010.

[8] P. Comon, "Independent component analysis, A new concept?," *Signal Processing*, vol. 36, pp. 287–314, April 1994.

[9] Anthony J Bell and Terrence J Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[10] Tzyy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin J Mckeown, Vicente Iragui, and Terrence J Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 2, pp. 163–178, 2000.

[11] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja, *Independent Component Analysis*, John Wiley and Sons Inc., 2001.

[12] Stephane Mallat, *A wavelet tour of signal processing: the sparse way*, Academic press, 2008.

[13] David L Donoho and Jain M Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[14] Bruno A Olshausen and David J Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607, 1996.

[15] Bruno A Olshausen and David J Field, "Sparse coding of sensory inputs," *Current opinion in neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[16] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801–808.

[17] Quoc V Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8595–8598.

[18] David L Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] Emmanuel J Candes and Terence Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[20] Richard Baraniuk, Emmanuel Candès, Robert Nowak, and Martin Vetterli, "Compressive sampling [from the guest editors]," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 12–13, 2008.

[21] Te-Won Lee, Mark Girolami, and Terrence J Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources," *Neural computation*, vol. 11, no. 2, pp. 417–441, 1999.

[22] Frédéric Abrard and Yannick Deville, "A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources," *Signal Processing*, vol. 85, no. 7, pp. 1389–1403, 2005.

[23] Rémi Gribonval and Sylvain Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges," in *ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks*. d-side publi., 2006, pp. 323–330.

[24] P. Georgiev, F. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 992–996, July 2005.

[25] Yuanqing Li, Shun-Ichi Amari, Andrzej Cichocki, Daniel WC Ho, and Shengli Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 2, pp. 423–437, 2006.

[26] Bruno A Olshausen and David J Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[27] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra, "Simple, efficient, and neural algorithms for sparse coding," in *Conference on Learning Theory*, 2015, pp. 113–149.

[28] Michael Zibulevsky and Barak A Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural computation*, vol. 13, no. 4, pp. 863–882, 2001.

[29] Dinh-Tuan Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Trans. on Signal Processing*, vol. 45, pp. 1712 – 1725, July 1997.

[30] Eren Babatas and Alper T Erdogan, "Sparse bounded component analysis," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–6.

[31] Lloyd N Trefethen and David Bau III, *Numerical linear algebra*, vol. 50, Siam, 1997.

[32] Adil Bagirov, Napsu Karmitsa, and Marko M Mäkelä, *Introduction to Nonsmooth Optimization: theory, practice and software*, Springer, 2014.

[33] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Stanford University, 2001.

[34] Shun-Ichi Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[35] Yurii Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[36] Weijie Su, Stephen Boyd, and Emmanuel J Candes, "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[37] Jérôme Bobin, Jean-Luc Starck, Jalal M Fadili, and Yassir Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2662–2674, 2007.

[38] F.-Y. Wang, Chong-Yung Chi, T.-H. Chan, and Y. Wang, "Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 875–888, May 2010.

[39] Alexander Grossmann and Jean Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[40] Yaniv Ziv, Laurie D Burns, Eric D Cocker, Elizabeth O Hamel, Kunal K Ghosh, Lacey J Kitch, Abbas El Gamal, and Mark J Schnitzer, "Long-term dynamics of ca1 hippocampal place codes," *Nature neuroscience*, vol. 16, no. 3, pp. 264–266, 2013.

[41] Eren Babatas and Alper T Erdogan, "Sparse bounded component analysis based neuron sorting for ca^{2+} imaging," in *preparation*.