IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

Sparse Source Separation from Orthogonal Mixtures

Moshe Mishali and Yonina C. Eldar

CCIT Report #704
September 2008

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL

CCIT REPORT #704 September 2008

Sparse Source Separation from Orthogonal Mixtures

Moshe Mishali and Yonina C. Eldar

Abstract

This paper addresses source separation from a linear mixture under two assumptions: source sparsity and orthogonality of the mixing matrix. We propose efficient sparse separation via a two-stage process. In the first stage we attempt to recover the sparsity pattern of the sources by exploiting the orthogonality prior. In the second stage, the support is used to reformulate the recovery task as an optimization problem. We then suggest a solution based on alternating minimization. Random simulations are performed to analyze the behavior of the resulting algorithm. The simulations demonstrate convergence of our approach as well as superior recovery rate in comparison with alternative source separation methods and K-SVD, a leading algorithm in dictionary learning.

I. Introduction

Blind source separation (BSS) is a fundamental problem in data analysis where the goal is to recover a set of source signals from their linear mixture, typically in the presence of noise. The blind aspect refers to the lack of prior information about the exact mixture coefficients or source values. A BSS problem is ill-posed unless additional source properties are assumed, *e.g.,* statistical independence or an underlying deterministic structure. In sparse component analysis (SCA), only a few (active) sources contribute to each measurement. Mathematically, a vector $\mathbf{s}(t)$ of n sources is constrained to have only K non-zero entries per each time instant t. The linear mixture model is described by

$$\mathbf{x}(t) = \mathbf{\Psi}\mathbf{s}(t) + \mathbf{n}(t), \quad t = 0, 1, \ldots, T-1, \tag{1}$$

This work was supported in part by the Israel Science Foundation under Grant no. 1081/07 and by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715).

The authors are with the Technion—Israel Institute of Technology, Haifa Israel. Email: moshiko@tx.technion.ac.il, yonina@ee.technion.ac.il.

where $\mathbf{\Psi}$ is an $m \times n$ unknown mixing matrix, $\mathbf{x}(t)$ is the time-varying measurement vector, and $\mathbf{n}(t)$ is an additive noise.

In this paper, we study the complete separation problem $(m = n)$ with an orthogonal $\mathbf{\Psi}$, which amounts to finding an orthonormal basis in which the sources are sparsely represented. Previous SCA methods, which focus on the underdetermined SCA problem $(m < n)$, propose finding $\mathbf{\Psi}$ either by combinatorial search [1], [2], or by an approximate maximization program [3], [4]. A related area in which source separation arises is dictionary design, in which the goal is to recover a dictionary $\mathbf{\Psi}$ such that the representation coefficients $\mathbf{s}(t)$ are sparse. A leading dictionary learning algorithm is the K-SVD [5]. Although this method does not explicitly assume an orthogonal dictionary $\mathbf{\Psi}$, it is possible in principle to modify the basic algorithm by orthogonalizing the solution. Preliminary simulations, which we show in Section V, show that simple modifications to K-SVD do not lead to significant improvement in source separation; therefore, we have not further pursued this approach. Lesage *et. al.* [6] describe a learning method for $m < n$ and $\mathbf{\Psi}$ which is a union of orthonormal bases [6]. Experimentally, they show that if $\mathbf{\Psi}$ is a single orthonormal basis, and a union model of two bases is used in the algorithms, then in about 50% of the simulations one of the recovered bases is $\mathbf{\Psi}$. Adjusting their method to learning a single basis leads to similar performance in our simulations. A recent paper studies the same orthogonal case we consider here [7]. The authors propose a non-convex program over the manifold of orthogonal matrices with an ℓ_1 sparsity-promoting objective. This program is proved to have a strict local minimum at the correct mixing matrix under some technical condition. Unfortunately, in practice, state-of-the-art descend solvers over the orthogonal manifold [8] did not converge in our simulations for the program of [7].

Our main contribution is a two-stage approach for efficient source separation with orthogonal $\mathbf{\Psi}$. Our idea is to separate the support recovery from recovery of the source values. In the first stage, we determine the support of the sources, namely which entries in $\mathbf{s}(t)$ are active for each time instant t. This structure is inferred from a set of rules, that exploit the orthogonality of $\mathbf{\Psi}$. Next, this information is incorporated into an optimization program allowing to replace the difficult combinatorial constraints of SCA by a set of linear equalities corresponding to the recovered support. Any solution to the resulting program must minimize the Lagrangian, which leads to a set of equations that we solve by alternating minimization. Empirically, the alternating procedure always converges and the recovery rate of the sparse sources is superior with respect to both K-SVD and [6], even when using a modified version of these algorithms that incorporates the orthogonality constraint.

The problem model is described in Section II where uniqueness of the separation is also addressed.

The main contribution appears in Sections III, IV which describe the support recovery algorithm and the subsequent alternating procedure, respectively. Supporting numerical experiments are presented in Section V, followed by conclusions and further extensions in Section VI.

II. Problem Formulation

A. Model

Starting with fixing notation, let $\mathbf{S}_{ik}$ be the ikth entry of the matrix $\mathbf{S}$ and $\mathbf{S}^T$ be the transpose of $\mathbf{S}$. We use $\mathbf{I}$ for the identity matrix of appropriate dimensions. The ith column (row) of $\mathbf{S}$ is denoted as $\mathbf{S}_{:i}$ ($\mathbf{S}_{i:}$). A finite index set J indicates the appropriate column subset $\mathbf{S}_{:J}$ with $|J|$ columns. Similarly, $\mathbf{S}_{J:}$ is a row subset. Vector support is defined as $\mathrm{supp}(\mathbf{v}) = \{k \,|\, \mathbf{v}_k \neq 0\}$. The Frobenius norm is $\|\mathbf{X}\|_F = \mathrm{Tr}(\mathbf{X}^T\mathbf{X})$.

We concentrate on the noise-free setting $\mathbf{n}(t) = \mathbf{0}$, in which the matrix form of (1) is written as

$$\mathbf{X} = \mathbf{\Psi}\mathbf{S}, \tag{2}$$

where the source vectors $\mathbf{s}(t), 1 \leq t \leq T$ are arranged as the columns of the $n \times T$ matrix $\mathbf{S}$, and the columns of $\mathbf{X}$ are the measurement vectors $\mathbf{x}(t)$. The more general scenario $\mathbf{N} \neq \mathbf{0}$ is shortly discussed in Section VI. In addition, (1) is considered over the real numbers with orthogonal $\mathbf{\Psi}$, namely $\mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{I}$. Extension to a complex-valued model with unitary $\mathbf{\Psi}$ is immediate.

The matrix $\mathbf{S}$ is assumed to be K-sparse, namely for each column $\|\mathbf{S}_{:i}\|_0 = K$, where the ℓ_0 pseudo-norm counts the number of non-zeros of its vector argument. The sparse separation problem can be viewed as an optimization of the squared-error between the given $n \times T$ measurement matrix $\mathbf{X}$ and an approximation by an orthogonal $\mathbf{\Psi}$ and K-sparse $\mathbf{S}$:

$$\begin{aligned} (\hat{\mathbf{\Psi}}, \hat{\mathbf{S}}) = \arg\min \quad & \|\mathbf{X} - \mathbf{\Psi}\mathbf{S}\|_F \\ \text{s.t.} \quad & \mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{I}, \\ & \|\mathbf{S}_{:i}\|_0 = K, \quad 1 \leq i \leq T. \end{aligned} \tag{3}$$

Evidently, the pair $\mathbf{\Psi}, \mathbf{S}$ of (2) is an optimal solution of the program (3). The next section provides a sufficient condition ensuring a unique optimal solution to (3).

B. Uniqueness

An important question underlying any separation algorithm is whether (3) admits a unique optimal solution pair $(\mathbf{\Psi}, \mathbf{S})$. The BSS problem (2) has inherent intermediacy which does not allow to determine $\mathbf{\Psi}, \mathbf{S}$ exactly; row permutations of $\mathbf{S}$ and column scaling of $\mathbf{\Psi}$. When $\mathbf{\Psi}$ is orthogonal the intermediacy

reduces to signed row permutations of $\mathbf{S}$, and correspondingly signed column permutations of $\mathbf{\Psi}$. To address uniqueness (up to these signed permutations) we quote the following result of [5].

Theorem 1: The factorization $\mathbf{B} = \mathbf{AS}$, with normalized columns $\mathbf{A} \in \mathbb{R}^{m\times n}$ and K-sparse $\mathbf{S}$, is unique up to signed permutations under the following conditions:

- *Support*: Every $2K$ columns of $\mathbf{A}$ are linearly independent.
- *Richness*: Upon clustering the columns of $\mathbf{S}$ according to their support, every cluster consists of at least $K+1$ columns, sharing the same support. In addition, for every $1 \leq i \leq n$, there exist k, j such that the intersection $\mathrm{supp}(\mathbf{S}_{:k}) \cap \mathrm{supp}(\mathbf{S}_{:j}) = \{i\}$.
- *Non-degeneracy*: Every column subset $\mathbf{S}_{:J}$ of cardinality $|J| = d$, with r rows that are non-identically zero, has $\mathrm{rank}(\mathbf{S}_{:J}) = \min\{d, r\}$.

In our case, the factorization (2) assumes an orthogonal $\mathbf{\Psi}$. Exploiting this prior leads to the following corollary.

Corollary 1: Consider the setting of Theorem 1 with orthogonal $\mathbf{A}$. Then, uniqueness holds if $n \geq 2K$, S is non-degenerate and rich, possibly up to the following richness exceptions. The intersection property may not hold for $i = n$, and for each cluster of columns (sharing the same support), it may also not hold for one entry of the support.

A rich matrix $\mathbf{S}$ can be constructed with $T = 2n(K+1)$ columns [5], whereas in the orthogonal case, the exceptions of Corollary 1 allow to reach a similar construction with $T = 2(n-n_0)(K+1)$ columns, where $n_0 = 1 + \lfloor (n-1)/K \rfloor$ and $\lfloor x \rfloor$ stands for the integer part of x.

Proof. The proof uses the constructive procedure described in [5], which consists of three stages. The first stage clusters the columns of $\mathbf{X}$ to different K-dimensional subspaces by creating column subsets whose rank is exactly K. The richness and non-degeneracy assumptions ensures the feasibility of the clustering. Moreover, the non-degeneracy of $\mathbf{S}$ ensures that every subspace corresponds to columns of $\mathbf{S}$ that share the same sparsity support.

The second stage examines all possible pairs of column clusters of $\mathbf{X}$, searching for pairs such that upon combining them to single cluster, the rank is exactly $2K-1$. Due to the non-degeneracy every such pair corresponds to a pair of column clusters of $\mathbf{S}$ that their supports intersect at exactly one atom. The last stage discovers one atom of $\mathbf{\Psi}$ from each such pair.

When $\mathbf{\Psi}$ is orthogonal, there is no change in the clustering stage, since a feasible solution requires only that every support pattern appears at least $K+1$ times. The last two stages are also unaffected for every $K-1$ support locations entries in every cluster. Now, the orthogonality of $\mathbf{\Psi}$ is used to discover the last entry of the support in every cluster, compensating the possibility that one entry may lack the

intersection property. This allows to discover $n-1$ atoms of $\mathbf{\Psi}$. The orthogonality property is now used again to fill up the last atom. ■

Theorem 1 and Corollary 1 are proved by explicitly constructing $\mathbf{\Psi}, \mathbf{S}$. This procedure has combinatorial complexity. In the next section, we describe a concrete non-combinatorial method to reveal the supports of the columns of $\mathbf{S}$.

III. Support Recovery

The support recovery stage relies on the following properties of $\mathbf{C} = \mathbf{X}^T\mathbf{X}$:

(**R1**) $\mathbf{C}_{kj} = 0$ implies that $\mathrm{supp}(\mathbf{S}_{:k}), \mathrm{supp}(\mathbf{S}_{:j})$ are disjoint (assuming $\mathbf{S}$ is non-degenerate)

(**R2**) $\mathbf{C}_{kj} \neq 0$ implies $\mathrm{supp}(\mathbf{S}_{:k}) \cap \mathrm{supp}(\mathbf{S}_{:j})$ is not empty

(**R3**) Every column $\mathbf{S}_{:k}$ contains exactly K non-zeros

(**R4**) Row permutations of $\mathbf{S}$ are allowed.

Algorithm 1 iteratively deduces locations of zeros and non-zeros of $\mathbf{S}$ by utilizing (**R1**)-(**R4**). An $n \times T$ matrix $\mathbf{Z}$ is used to indicate the support of $\mathbf{S}$ and is initialized to $\mathbf{Z}_{ik} = \phi$ for all entries, implying no prior knowledge. An update $\mathbf{Z}_{ik} = 0$ or 1 is performed once $\mathbf{S}_{ik}$ is identified as zero or non-zero, respectively. When a non-zero is identified, $\mathbf{Z}_{ik} = 1$, and (**R1**) leads to an update $\mathbf{Z}_{ij} = 0$ for every $\mathbf{C}_{kj} = 0$ (step 11). Once $\mathbf{Z}_{:k}$ identifies exactly K non-zeros, which determines $\mathrm{supp}(\mathbf{S}_{:k}) = \{i | \mathbf{Z}_{ik} = 1\}$, (**R2**) implies that a column j with $\mathbf{C}_{kj} \neq 0$ has $\mathbf{Z}_{ij} = 1$ for at least one $i \in \mathrm{supp}(\mathbf{S}_{:k})$. A conclusion of this type, namely a specific column j which has at least one non-zero entry within an index set $\mathcal{I}$, is referred to as a (regular) *rule*. Fig. 1 illustrates the implications of (**R1**),(**R2**).

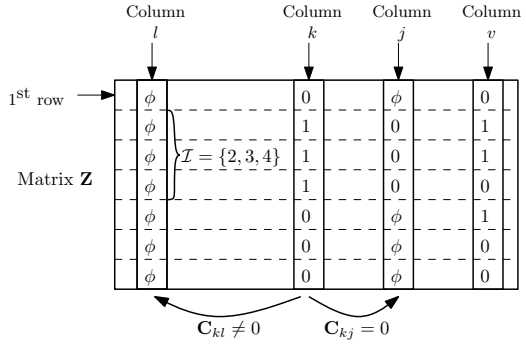


Fig. 1: The non-zero entries of column k implies zeros in column j since $\mathbf{C}_{kj} = 0$. A rule with $\mathcal{I} = \{2,3,4\}$ within column l is based on $\mathbf{C}_{lk} \neq 0$. The rule can be effectively solved out to $\mathbf{Z}_{4l} = 1$ if $\mathbf{C}_{vl} = 0$.

Rules are dynamically updated as follows. Entries $i \in \mathcal{I}$ for which $\mathbf{Z}_{ij} = 0$ are automatically discarded from $\mathcal{I}$. If $\mathbf{Z}_{ij} = 1$ for some $i \in \mathcal{I}$, then the rule does not contribute information and is removed (step 5).

Input : $\mathbf{C} = \mathbf{X}^T\mathbf{X}$, sparsity level K

Output : Support pattern $\mathbf{Z}$, rule list L

Initialization: $\mathbf{Z}_{ik} = \phi$, L with T special rules (see text)

1 **for** $k = 1$ **to** T **do**
2 **foreach** *Unsolved rule* $r \in L$ *pointing column* k **do**
3 Let $\mathcal{I}$ be the entries pointed by r
4 **if** $\mathbf{Z}_{ik} = 1$ *for some* $i \in \mathcal{I}$ *and* r *is regular* **then**
5 Remove r from L
6 **end**
7 **if** *Symmetry holds with respect to* $\mathcal{I}$ **then**
8 Choose $i \in \mathcal{I}$ and set $\mathbf{Z}_{ik} = 1$
9 **end**
10 **end**
11 **foreach** $\{i, j \,|\, \mathbf{C}_{jk} = 0, \mathbf{Z}_{ik} = 1\}$ **do** $\mathbf{Z}_{ij} = 0$
12 **if** $\mathbf{Z}_{:k}$ *indicates exactly* K *non-zero values* **then**
13 $\mathrm{supp}(\mathbf{S}_{:k}) = \{i | \mathbf{Z}_{ik} = 1\}$
14 $\mathbf{Z}_{ik} = 0$ for every $i \notin \mathrm{supp}(\mathbf{S}_{:k})$
15 Remove the kth special rule from L
16 **foreach** $\{j \,|\, \mathbf{C}_{kj} \neq 0\}$ **do**
17 Add a (regular) rule on column j with $\mathcal{I} = \mathrm{supp}(\mathbf{S}_{:k})$
18 **end**
19 **end**
20 **end**
21 Repeat steps 1-20 till no further change in $\mathbf{Z}, L$

Algorithm 1: Support Recovery

If the rule is reduced to $I = \{i\}$, then $\mathbf{Z}_{ij} = 1$ (step 8). This step is also executed when $|I| > 1$ and the rows $\mathbf{Z}_{\mathcal{I}:}$ are equivalent, in the sense that zeros and non-zeros are identified in the same locations and that the rule list L reflects a similar symmetry. Then, (**R4**) implies that $\mathbf{Z}_{ij} = 1$ for some $i \in \mathcal{I}$. To use (**R3**), T special rules are created on initialization. The kth special rule, $1 \leq k \leq T$, has $\mathcal{I} = \{1, \ldots, n\}$ for column k. Special rules are not removed in step 5 until K non-zeros are identified.

The main loop of Algorithm 1 identifies new non-zero locations by resolving rules, which in turn identifies the locations of zeros. The feedback from the latter effects the rule list (step 17) and vice versa. Note that the first special rule triggers the first update: $\mathbf{Z}_{i1} = 1$ for $1 \leq i \leq K$ and $\mathbf{Z}_{i1} = 0$ for $i > K$.

A rigorous complexity analysis of Algorithm 1 is beyond the current scope. Nonetheless, a few guarantees can easily be verified:

- The algorithm always reaches the termination point, step 21. This property holds due to the termination condition and since every $\mathbf{Z}_{ik}$ can be updated only once to 0 or 1.
- The main loop, steps 2-19, identifies at least one new non-zero entry of $\mathbf{S}$, in every sweep of k from 1 to T. Thus, Algorithm 1 may execute these steps nKT times at the most before terminating.
- For $T = 1$, the support is fully recovered. More generally, the support is determined whenever $\mathbf{S}$ consists of column subsets whose supports are pairwise disjoint and each column subset share the same support.

It is interesting to note that full support recovery is possible even when there is no unique solution to (3), *e.g.,* $T = 1$.

The underlying logic behind Algorithm 1 can be extended by taking into account more sophisticated decision rules. For example: once termination is reached, solve one of the remaining regular rules manually by setting $\mathbf{Z}_{ik} = 1$ for some $i \in \mathcal{I}$, and then return to step 1. If this choice does not lead to full support recovery, roll changes back and try another $i \in \mathcal{I}$. We have not pursued this approach since in certain pathological supports it generates an exponential-time search. However, our experiments show that in many cases L contains only small sets of unsolved rules at termination, giving practical reasoning to this strategy.

We point out that partial support recovery may happen (*e.g.,* a rich $\mathbf{S}$ with the minimal number of columns $T = 2(n - n_0)(K + 1)$). Fortunately, as further described in the next section, in many cases partial support recovery still allows exact source separation. Since the next recovery stage relies on the zero locations of $\mathbf{S}$, we may slightly modify Algorithm 1 in order to maximize the number of zeros that are identified: The sequential sweep of k in step 1 can be replaced by choosing the column k, which identifies the maximal number of new zeros locations in $\mathbf{S}$. A naïve implementation of this strategy requires to execute steps 2-19 virtually for every $1 \leq k \leq T$ and then commit the changes only for the selected column. However, as none of these steps is recursive, the complexity is approximately multiplied by T. Our experiments show clear advantage for this approach, in the sense that at termination more zeros are identified compared to selecting sequential columns in step 1.

IV. Source Separation

Using the additional information of $\mathbf{Z}$, the combinatorial sparsity constraints $\|\mathbf{S}_{:i}\|_0 = K$ of (3) are now translated to a set of linear equalities, indexed by the locations of the zeros:

$$\begin{aligned}(\mathbf{\Psi}, \mathbf{S}) = \arg\min \quad & \|\mathbf{X} - \mathbf{\Psi}\mathbf{S}\|_F \\ \text{s.t.} \quad & \mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{I}, \\ & \mathbf{S}_{ik} = 0,\ \forall i,k \text{ with } \mathbf{Z}_{ik} = 0.\end{aligned} \tag{4}$$

Any solution to (4) will minimize the Lagrangian:

$$\mathcal{L}(\mathbf{\Psi}, \mathbf{S}) = \|\mathbf{X} - \mathbf{\Psi}\mathbf{S}\|_F^2 + \mathrm{Tr}\left[\mathbf{\Gamma}(\mathbf{\Psi}^T\mathbf{\Psi} - I)\right] + \mathrm{Tr}(\mathbf{\Pi}^T\mathbf{S}), \tag{5}$$

where $\mathbf{\Gamma}, \mathbf{\Pi}$ are matrices of Lagrange multipliers with appropriate dimensions and $\mathbf{\Pi}_{ik} = 0$ for $\mathbf{Z}_{ik} \neq 0$. Calculating the gradients with respect to each variable yields

$$\nabla_{\mathbf{\Pi}}\mathcal{L} = \mathbf{0} \quad \rightarrow \quad \mathbf{S}_{ik} = 0,\ \forall i,k \text{ with } \mathbf{Z}_{ik} = 0 \tag{6a}$$

$$\nabla_{\mathbf{\Gamma}}\mathcal{L} = \mathbf{0} \quad \rightarrow \quad \mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{I} \tag{6b}$$

$$\nabla_{\mathbf{\Psi}}\mathcal{L} = \mathbf{0} \quad \rightarrow \quad (\mathbf{X} - \mathbf{\Psi}\mathbf{S})\mathbf{S}^T = \mathbf{\Psi}\mathbf{\Gamma} \tag{6c}$$

$$\nabla_{\mathbf{S}}\mathcal{L} = \mathbf{0} \quad \rightarrow \quad -2\mathbf{\Psi}^T\mathbf{X} + 2\mathbf{\Psi}^T\mathbf{\Psi}\mathbf{S} + \mathbf{\Pi} = \mathbf{0}. \tag{6d}$$

To find $\mathbf{\Psi}, \mathbf{S}$ satisfying (6a)-(6d), we propose an alternating minimization between $\mathbf{\Psi}, \mathbf{S}$. Fixing the current estimate of $\mathbf{S}$, then the optimal $\mathbf{\Psi}$ satisfying (6b)-(6c) is given by $\mathbf{\Psi} = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U}, \mathbf{V}$ are orthogonal matrices taken from the singular value decomposition (SVD) $\mathbf{X}\mathbf{S}^T = \mathbf{U}\Sigma\mathbf{V}^T$ [6]. Next, it can be verified from (6a) and (6d) that the optimal $\mathbf{S}$, when keeping an orthogonal $\mathbf{\Psi}$ fixed, is computed as $\mathbf{S}_{ik} = [\mathbf{\Psi}^T\mathbf{X}]_{ik}$ where $\mathbf{Z}_{ik} \neq 0$ and $\mathbf{S}_{ik} = 0$ otherwise.

Algorithm 2 summarizes the recovery of $\mathbf{\Psi}, \mathbf{S}$ given the support matrix $\mathbf{Z}$. Note that (4) is not convex, and thus the system (6) only guarantees a stationary point. Moreover, the alternating approach does not guarantee convergence (even to a stationary point) as (6a)-(6d) are not solved simultaneously. Nonetheless, empirically, Algorithm 2 always converged to an optimal point of (4). Interestingly, in many cases a K-sparse $\mathbf{S}$ is found even when the support is partially recovered. The relation between identification of zeros and source recovery is evident from (4) and is further studied, empirically, in the next section.

V. Numerical Experiments

In order to examine recovery based on our algorithm, the following setup is used. For fixed problem dimensions (n, T, K), a K-sparse matrix $\mathbf{S}$ of size $n \times T$ is constructed by selecting the non-zero locations

Input : $\mathbf{X}$, support pattern $\mathbf{Z}$

Output : $\mathbf{\Psi}, \mathbf{S}$

Initialization: $\mathbf{\Psi}$ is a random $n \times n$ orthogonal matrix

repeat

  $\mathbf{S}_{ik} = [\mathbf{\Psi}^T \mathbf{X}]_{ik}$ where $\mathbf{Z}_{ik} \neq 0$ and $\mathbf{S}_{ik} = 0$ otherwise

  Decompose $\mathbf{X}\mathbf{S}^T = \mathbf{U}\Sigma\mathbf{V}^T$ using SVD

  $\mathbf{\Psi} = \mathbf{U}\mathbf{V}^T$

until *convergence*

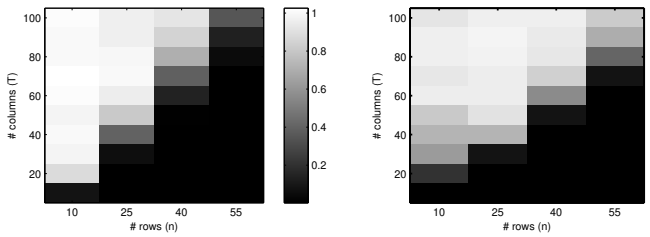
Algorithm 2: Source and Mixing Matrix Recovery

uniformly at random for each column, then drawing non-zero values from a normal distribution. The non-degeneracy assumption holds with probability one for this choice. Next, an orthogonal $\mathbf{\Psi}$ is constructed by taking the left orthogonal matrix in the SVD of a square-n matrix whose entries are drawn independently from the normal distribution. The measurement matrix is computed as $\mathbf{X} = \mathbf{\Psi}\mathbf{S}$.

Algorithm 1 is executed in order to recover the support of $\mathbf{S}$. The (possibly partial) support information is passed to Algorithm 2 which recovers a candidate pair $\hat{\mathbf{\Psi}}, \hat{\mathbf{S}}$. A success recovery is announced once $\hat{\mathbf{S}}$ is K-sparse and equals $\mathbf{S}$ up to signed row permutations, and $\hat{\mathbf{\Psi}} = \mathbf{\Psi}$ up to the corresponding signed column permutations. The average recovery success over 100 trials is reported in Fig. 2 for several combinations of (n, K, T).

As evident from the figure, increasing the number of measurements improves the recovery rate. In addition, as n grows, a higher number of columns T is required to achieve the same recovery rate. This behavior conforms with the dependency of $T = 2(n - n_0)(K + 1)$ on n, which implies a possible construction of $\mathbf{S}$ satisfying the richness assumption. Interestingly, none of the examples $\mathbf{S}$ in the simulations satisfies richness. Yet, as Fig. 2 reveals, our approach which is much faster than a combinatorial recovery, shows similar dependency of the recovery rate on n, T.

We compare our method to K-SVD [5], an iterative algorithm which finds a candidate pair of normalized columns (though not necessarily orthogonal) $\hat{\mathbf{\Psi}}$ and K-sparse $\hat{\mathbf{S}}$. The performance of the original K-SVD algorithm is far from the results depicted in Fig. 2. We slightly modified the algorithm by adding an orthogonalization step between every $d = 5$ consecutive iterations, in the same way performed in Algorithm 2. The choice $d = 5$ was found to provide fast convergence to an orthogonal solution. In addition, our method is compared to [6], which in case of an orthogonal $\mathbf{\Psi}$ reduces to K-SVD with



(a) $K = 2$ (b) $K = 3$

Fig. 2: Image intensity represents the average recovery of $\mathbf{\Psi}, \mathbf{S}$ over 100 trials per each value of n, T and for two values of K.

orthogonalization step after every iteration. A simple modification (along the lines proposed in [6]) is used to improve recovery rate; the threshold parameter λ of the sparse coding stage is gradually decreased (in 100 iterations) towards zero, see [6, Eq. (5)]. Fig. 3 depicts the results and reveals a superior recovery rate for our method. We note that our method and the modified version of [6] have comparable run times, whereas the modified K-SVD is faster.

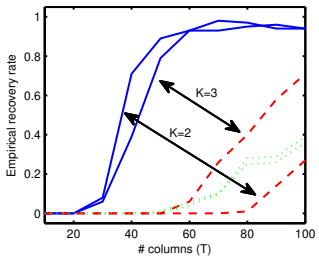
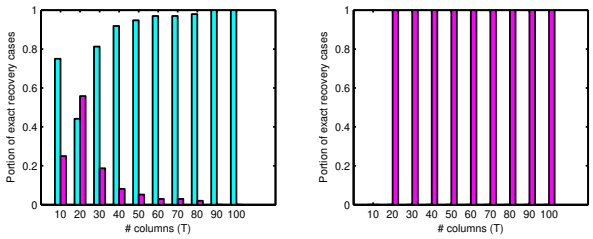


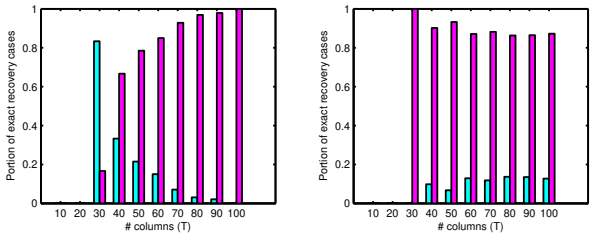
Fig. 3: Comparison between our method (solid lines), the modified K-SVD algorithm (dashed lines), and the modified version of [6] (dotted lines) as T varies from 10 to 100, $n = 10$ and two choices of K.

We also tried to compare our method with [7], who proposed to minimize $J(\mathbf{\Psi}) = \|\mathbf{\Psi}^T\mathbf{X}\|_1 = \sum_{ik} |[\mathbf{\Psi}^T\mathbf{X}]_{ik}|$ over the manifold of orthogonal $\mathbf{\Psi}$, which is a non-convex program and requires a special solver. We used both descend algorithms of [8] for that purpose. However, neither converged in our simulations.

Algorithm 2 uses the identified zero locations to update $\hat{\mathbf{S}}$ in the alternating iterations. To explore the relation between zeros identifications and source separation, we use the results of Fig. 2 to count the cases of exact recovery whose support is fully recovery. We compare these cases to exact recovery with partial support recovery. Fig. 4 reports the results. Clearly, a full support recovery is not necessary for exact



(a) $n = 10, K = 2$ (b) $n = 10, K = 3$



(c) $n = 25, K = 2$ (d) $n = 25, K = 3$

Fig. 4: The bars indicate the portion of simulations with exact recovery of $\mathbf{\Psi}, \mathbf{S}$, for which full support recovery occurs (cyan) and for which only partial is recovered (magenta).

recovery. However, the number of identified zero is not the only factor for exact separation. Specifically, we encountered cases with same number of identified zeros, where some lead to exact recovery and others do not.

VI. Conclusions and Extensions

We developed a method for sparse source separation in an orthogonal mixture model. Algorithm 1 exploits invariance of inner products under orthogonal transforms to gather information about the support of the sources. Using this knowledge, we can substitute the original combinatorial sparsity constraints by linear equalities. A saddle point of the resulting optimization program can then be found by minimizing the Lagrangian. Algorithm 2 introduces an alternating minimization procedure for this purpose. Numerical experiments demonstrate the potential of our strategy, although further study and comparisons are required.

The current version of Algorithm 1 is highly sensitive to noise. Straightforward adjustments involve thresholding the correlation matrix $\mathbf{C}$ prior to running the algorithm. In contrast, the development of Algorithm 2 does not assume a noise-free setting and thus it may also be used in the presence of noise.

Finally, we point out that although a strict theoretical richness assumption is used to prove uniqueness, high empirical recovery performance is noticed even when richness does not hold.

REFERENCES

[1] P. Georgiev, F. Theis, and A. Cichocki, "Blind source separation and sparse component analysis of overcomplete mixtures," *ICASSP'04*, vol. 5.

[2] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: Principles, perspectives, and new challenges," *Proceedings of ESANN*, vol. 6, pp. 323–330, 2006.

[3] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.

[4] Y. Li, A. Cichocki, and S. Amari, "Analysis of sparse representation and blind source separation," *Neural Computation*, vol. 16, no. 6, pp. 1193–1234, 2004.

[5] M. Aharon, M. Elad, and A. M. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *Linear Algebra and Its Applications*, vol. 416, no. 1, pp. 48–67, 2006.

[6] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," Tech. Rep., 2004.

[7] R. Gribonval and K. Schnass, "Dictionary identifiability from few training samples," *in Proc. 16th EUSIPCO08*, Aug. 2008.

[8] T. E. Abrudan, J. Eriksson, and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1134, 2008.

[9] M. Mishali and Y. C. Eldar, "Sparse Source Separation from Orthogonal Mixture," *CCIT Report no. 704, EE Dept., Technion - Israel Institute of Technology*, Sep. 2008.