Estimating the Mixing Matrix in Sparse Component Analysis Based on Converting a Multiple Dominant to a Single Dominant Problem

Nima Noorshams[1], Massoud Babaie-Zadeh[1,⋆], and Christian Jutten[2]

[1] Electrical Engineering Department, Advanced Communications Research Institute (ACRI), Sharif University of Technology, Tehran, Iran
[2] GIPSA-lab, Department of Images and Signals, National Polytechnic Institute of Grenoble (INPG), France
nima_noorshams@yahoo.com, mbzadeh@yahoo.com, Christian.Jutten@inpg.fr

Abstract. We propose a new method for estimating the mixing matrix, $\mathbf{A}$, in the linear model $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), t = 1, \ldots, T$, for the problem of underdetermined Sparse Component Analysis (SCA). Contrary to most previous algorithms, there can be more than one dominant source at each instant (we call it a "multiple dominant" problem). The main idea is to convert the multiple dominant problem to a series of single dominant problems, which may be solved by well-known methods. Each of these single dominant problems results in the determination of some columns of $\mathbf{A}$. This results in a huge decrease in computations, which lets us to solve higher dimension problems that were not possible before.

1 Introduction

Sparse Component Analysis (SCA) [1,2,3,4] is a semi-Blind Source Separation problem [5], in which it is a priori known that the source signals are 'sparse'. A sparse signal is a signal whose most samples are nearly zero, and just a few percents take significant values. It has been already shown that such a prior information permits source separation for the case the number of sources exceeds the number of sensors [6,1,2,3,4].

The problem of SCA can be stated as follows. Consider the linear model:

$$\mathbf{x}(t) = \sum_{i=1}^{n} \mathbf{s}_i(t)\mathbf{a}_i = \mathbf{A}\mathbf{s}(t) \quad t = 1, 2, \ldots, T \tag{1}$$

where $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ is the mixing matrix, $\mathbf{s}(t)$ and $\mathbf{x}(t)$ are the vectors of all samples of n sources and m observed signals (mixtures) at instant t, T is the number of 'time' samples. The goal of SCA is then to estimate $\mathbf{A}$ and $\mathbf{s}(t)$, only from $\mathbf{x}(t)$, $1 \leq t \leq T$ and the sparsity assumption. In this paper, we address only the problem of estimation of $\mathbf{A}$ (note that where there are more sources than sensors, it is not equivalent to the estimation of the sources). We call each

⋆ This work has been partially funded by Sharif University of Technology, by Center for International Research and Collaboration (ISMO) and by Iran NSF (INSF).

M.E. Davies et al. (Eds.): ICA 2007, LNCS 4666, pp. 397–405, 2007.

column of the mixing matrix, i.e. each $\mathbf{a}_i$, $1 \leq i \leq n$, a *mixing vector*. Although the word "time" will be used throughout this paper, the above model may be in another domain, in which the sparsity assumption holds. To see this, let $\mathcal{T}$ be a linear 'sparsifying' transform, and the mixing system is stated as $\mathbf{x} = \mathbf{As}$ in the time domain. Then, we have $\mathcal{T}\{\mathbf{x}\} = \mathbf{A}\,\mathcal{T}\{\mathbf{x}\}$ in the transformed domain, and because of the sparsity of $\mathcal{T}\{\mathbf{s}\}$, it is in the form of (1).

Let k denote the average number of active sources at each instant. In fact, if the probability of inactivity of a source is denoted by p (sparsity implies that $p \approx 1$), then[1] $k = n(1-p)$. Then, two different cases should be distinguished for estimating the mixing matrix: single dominant component and multiple dominant components. In the former, k is equal to one, and the scatter plot of $\mathbf{x}(t)$ $(t = 1, \ldots, T)$ geometrically shows the data concentration directions. This can be seen from the fact that at each instant, $\mathbf{x}(t) = \mathbf{As}(t) = s_1(t)\mathbf{a}_1 + \cdots + s_n(t)\mathbf{a}_n$, $t = 1, \ldots, T$; and for most instants, only one of s_i's is dominant and the others are almost zero. Consequently, in most samples, $\mathbf{x}(t)$ is in the direction of one of the mixing vectors. In the latter, k is greater than one and the mixing matrix would not be estimated easily from the scatter plot. Up to now, many papers have been addressed the former case [1,3,4], while only few researchers have considered the latter case [4,7,8]. In this paper, we focus on the case of multiple dominant components.

In the multiple dominant components SCA, the observed data concentrate around k-dimensional subspaces which are spanned by a set of k mixing vectors. We call these subspaces *concentration subspaces* throughout this paper. In a multiple dominant problem finding a k-dimensional concentration subspace is not equivalent to find some of the mixing vectors. All of the existing methods [7,9] need to find most of the concentration subspaces and then estimate the mixing matrix from them (this is not the case for our algorithm).

The main idea of this paper is to show that the multiple dominant problem can be converted to a series of single dominant problems, which may be solved by simple algorithms of the single dominant problem to estimate the mixing matrix. Moreover, by estimating each concentration subspace, some of the mixing vectors are found (contrary to [7,9] in which all or many concentration subspaces were needed to be estimated before starting the estimation of mixing vectors). This results in a low computational cost in comparison to the methods of [7,9] and therefore, problems with higher dimensions can be solved by this algorithm. Up to our best knowledge there is no practical algorithm for solving this problem when $k \geq 3$ but our method can handle dimensions more than this.

Throughout the paper, we suppose that the sources are sparse enough so $k < m/2$ (where m is the number of mixtures), the sources are independent and the probability of activity are the same for all of them. Finally, we assume also that each subset of m columns of $\mathbf{A}$ is linearly independent.

[1] More precisely, in this paper, by the average number of active sources we mean an integer. If $n(1-p)$ is slightly greater than an integer $k = \lfloor n(1-p) \rfloor$ (for example is $n(1-p) = 1.05$, then the k-means algorithm, which has been designed for $k = 1$, still works). In other cases, $k = \lceil n(1-p) \rceil$.

2 The Main Idea

The main idea of converting a multiple dominant problem to a single dominant one comes from the following theorem (the proof is left to the appendix).

Theorem 1. *If $k \leq \frac{m}{2}$ and the sources are statistically independent then the average number of active sources in a k dimensional concentration subspace (denoted by $\widetilde{k}$) is $\widetilde{k} = k(1-p)$.*

The above theorem states that although the average number of active sources $k = n(1-p)$ may be greater than 1, the average number of active sources *within a concentration subspace* $\boldsymbol{B}$ (that is, $\widetilde{k} = k(1-p) = n(1-p)^2$) is *one level sparser*. In other words, a multiple dominant problem in the original space may be transformed into a single dominant problem within the subspace $\mathbf{B}$. Consequently in the subset of data points which lie in $\mathbf{B}$, we can use a single dominant algorithm (like that of [2]) for estimating the mixing vectors which are a subset of the mixing vectors of the main problem. If $n(1-p)^2$ does not less than or approximately equal to one, then the single dominant assumption does not hold and the above technique should be used one or several levels.

In summary, our approach for estimating the mixing matrix consists of the following steps:

- **Step 1:** Find a new concentration subspace. A concentration subspace can be found by maximizing a cost function (see Sec. 3). For finding a 'new' concentration subspace, the steepest ascent is initialized by a randomly different starting point (note that there are a lot of concentration subspaces).
- **Step 2:** Determine all data points which lie in this concentration subspace, and run a single dominant algorithm to find the mixing vectors in that subspace, *which are a subset of the mixing vectors of the main problem*. The points whose distances to the desired subspace are less than a specific value are supposed to belong to this subspace.
- **Step 3:** If all of the mixing vectors have been found, the search has been finished. Otherwise, go to step one, and continue. In this paper the number of sources is supposed to be known in advance.

Remark: Assuming that the probability of inactivity (p) is identical for all sources, p^n is the probability of no source being active, and hence p can be estimated as $\hat{p} = (\frac{N}{T})^{1/n}$, where T is the total number of data points, and N is the number of 'active' data points (i.e., $\mathbf{x}$'s whose distances from the origin is greater than a threshold). However, in this paper, p is assumed already known.

3 Finding Concentration Subspaces

Each k-dimensional subspace can be represented by an m by k matrix, whose columns form an orthonormal basis for the subspace. In this paper, we do not distinguish between a subspace and its matrix representation. Let $\mathbf{B} \in \mathbb{R}^{m \times k}$ be the orthonormal matrix representation of an arbitrary k-dimensional subspace.

The following cost function has been presented in [9] to detect whether $\mathbf{B}$ is a concentration subspace or not:

$$f_\sigma(\mathbf{B}) = \sum_{i=1}^{T} \exp\left(\frac{-d^2(\mathbf{x}_i, \mathbf{B})}{2\sigma^2}\right), \tag{2}$$

where $d(\mathbf{x}_i, \mathbf{B})$ is the distance of $\mathbf{x}_i$ from the subspace represented by $\mathbf{B}$ [9].

For small values of $d(\mathbf{x}_i, \mathbf{B})$ compared to σ, $\exp(-d^2(\mathbf{x}_i, \mathbf{B})/2\sigma^2)$ is about 1 and for large values of $d(\mathbf{x}_i, \mathbf{B})$, it is nearly zero. Therefore, for sufficiently small values of σ, the above function *is approximately equal to the number of data points close to* $\mathbf{B}$. Therefore, by maximizing the function f, we actually maximize the number of data points close to $\mathbf{B}$ thus we find a concentration subspace. Moreover, if the set of points are concentrated around several different k-dimensional concentration subspaces, f has a local maximum where $\mathbf{B}$ is close to the basis of each of them.

The idea of [9] for finding a concentration subspace is to maximize the function f_σ for a sufficiently small σ, using steepest ascent method. For very small σ, many local maxima exist which do not correspond to any concentration subspaces. These local maxima correspond to spaces which contains $r < k$ mixing vectors instead of k. On the other hand if σ is large, then the peaks are mixed together. In contrast to [9] which uses an iterative method by considering a sequence of decreasing σ to prevent getting trapped in local maxima, in this paper we use only a medium value for σ. In each step, we find a subset of k mixing vectors which are related to the estimated concentration subspace. As will be discussed in Sec. 7, if an incorrect concentration subspace with r $(r < k)$ mixing vectors is estimated, the algorithm detects r mixing vector rather than k and therefore it is robust to these errors.

4 Estimating Mixing Vectors and the Mixing Matrix

Consider a concentration subspace $\mathbf{B}$ and suppose that the points $\mathbf{x}_i$ for $i \in \mathbf{I} \subset \{1...T\}$ belong to this subspace. The fact that $\widetilde{k} < 1$ ensure us that most of these points concentrate along k, 1-dimensional subspaces. Then, we use the same idea of [2] designed for finding the mixing vector in the case $k = 1$: Firstly, data samples are normalized by dividing them by their norms ($\bar{\mathbf{x}}_i = \mathbf{x}_i / \|\mathbf{x}_i\|$), that is, the points are projected onto the unit sphere. Moreover, the sign of the first component is forced to be positive. Then, we have a point distribution on a unit hemisphere. Note that most of these points are concentrated around k points, and hence the mixing vectors (which corresponded to the centroid of these clusters) may be found by a clustering algorithm.

However, there are numerous outliers which do not belong to any clusters. Outlier points make the clustering algorithms inaccurate and increase the probability of error in detecting cluster centers, therefore they have to be removed as more as possible. We say that two points are neighbor if the distance between them is less than a specific value r which is dependent to the energy of

the sources. For outlier detection the fact that outliers are alone in the space is used. In other words, they do not have any neighbor, but this is not true for cluster centers because the density around them is high. By this definition, a point is considered as an outlier if it does not have any neighbor.

The method we used in this paper for the clustering is subtractive clustering [10]. In this method each point is considered as a cluster center and its potential for being a cluster center is computed. The point with highest potential is considered as a center and that cluster is removed. This process continues to find all clusters.

5 The Final Algorithm

Putting all together, the final algorithm is summarized as follows.

1. Remove the data samples ($\mathbf{x}(t)$) which are near the origin. In these samples, all of the sources are probably inactive.
2. Estimate k to set the dimension of the concentration subspaces and also p to check that if $\tilde{k}$ is smaller than 1.
3. Assume an appropriate value for the free parameter of the cost function (σ).
4. Maximize $f_\sigma(\mathbf{B})$ with the steepest ascent algorithm in several steps:
   (a) Choose a random starting subspace (an orthonormal m by k matrix $\mathbf{B}_1$).
   (b) Set $\mathbf{B}_{j+1} = \mathbf{B}_j + \mu \partial f_\sigma / \partial \mathbf{B}_j$.[2]
   (c) Orthonormalize $\mathbf{B}_{j+1}$.
   (d) If $\|\mathbf{B}_{j+1} - \mathbf{B}_j\| < 10^{-3}$ go to (5) else $j = j + 1$ and go to (b).
5. Consider the points whose distances to $\mathbf{B}$ are less than a specific value (d) and ignore the other points.
6. Normalize the points and force the sign of the first component to be positive.
7. Remove the points with no adjacent (outlier points) by preprocessing.
8. Detect the cluster centers with subtractive clustering algorithm (these vectors are some of the mixing vectors).
9. Compare obtained vectors (in the previous step) with former mixing vectors. If each of these vectors is new[3], then add it up to the list of estimated vectors, else throw it away.
10. If the number of estimated mixing vectors is n, then stop the algorithm, else go back to (4).

6 Experimental Results

In this section, 2 simulations are presented to justify the algorithm. In all of these simulations, sparse sources are generated independently and identically distributed (i.i.d) by the Bernoulli-Gaussian model. In other words, the sources

[2] In all simulations we consider $\mu = .01$.

[3] Two vectors are considered identical if the angle between them is less than a certain amount (5 degree in our simulations).

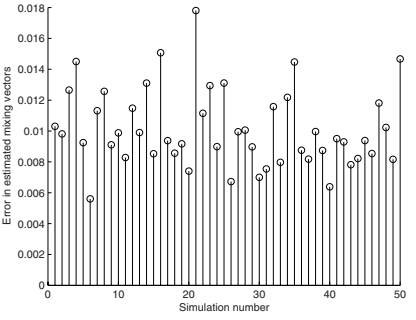

Fig. 1. Error of the overall algorithm for all simulations in the case $n = 10$, $m = 6$, $k = 2$ and $T = 10000$ for 50 different simulations

are inactive with probability p and are active with probability $1 - p$. In the inactive case, their value is a zero mean Gaussian with standard deviation σ_{off}, and in active case it is a zero mean Gaussian with standard deviation σ_{on}. Consequently $s_i \sim\ (1-p)\,\mathcal{N}(0, \sigma_{\text{on}}) + p\,\mathcal{N}(0, \sigma_{\text{off}})$.

In order to have sparse sources, the conditions $\sigma_{\text{on}} \gg \sigma_{\text{off}}$ and $p \approx 1$ should be applied (σ_{off} is to model the noise). In all simulations, the values $\sigma_{\text{on}} = 1$ and $\sigma_{\text{off}} = 0.005$ have been used and each component of the mixing matrix is generated randomly in the $[0, 1]$ interval after that each column of it, is normalized.

All simulations were performed in MATLAB 7 under WindowsXP, using an Intel Pentium IV 2.4 GHz processor with 1 Gigabyte RAM.

Experiment 1: Performance
In this experiment, the performance of our algorithm is demonstrated. 50 simulations for 50 different mixing matrixes are performed for the case $n = 10$, $m = 6$, $k = 2$ ($p = 0.8$) and $T = 10000$. The parameters are chosen as $\sigma = 1/40$, $d = .01$ and $r = .02$.

In all cases, the obtained vectors are compared with the mixing vectors. For comparison the criterion $\mathcal{E} = \min_{\mathbf{P}\in\mathcal{P}} \|\mathbf{A} - \hat{\mathbf{A}}\mathbf{P}\|_2$ is used, where $\mathcal{P}$ is the set of all permutation matrices (this is the same criterion used in [7]). This estimation error is shown in Fig. 1 for all simulations.

The average number of iterations for successfully finding all mixing vectors is around 30, but in 3 simulations this number exceeded 100 iterations and in 1 case more than 500 iterations was required. This may increase the run time of the algorithm. By considering this inefficiency the processes took less than 90 sec in average for estimating a mixing matrix. Moreover the maximum error in the mixing matrix estimation is .018, therefore, the error is negligible.

Experiment 2: Middle and large scale problems
To show that the method is capable of solving medium scale problems, two simulations are performed. In the first simulation, the parameters were $n = 25$,

$m = 15$, $k = 5$ and $T = 100000$, whereas in the second experiment, they were $n = 35$, $m = 20$, $k = 4$ and $T = 50000$. The process took about 1 hour for the first case and 3 hours for the second case. As far as we know there is no algorithm to estimate the mixing vectors in these dimensions ($k = 4$ or 5). In these scales the sources are not so sparse but our algorithm can handle this situation.

To measure the accuracy of the estimation, the angle between each estimated vector and its corresponding actual mixing vector (i.e. inverse cosine of their dot product) were calculated. These n angles were all less than 0.01 radian, showing that all of the mixing matrix have been correctly estimated.

7 Conclusion and Discussion

In this paper, we introduced a method for estimating the mixing matrix in the multiple dominant SCA problem which can handle larger k in comparison to other methods ([7,9]).

At our best knowledge, all existing SCA methods are unable to estimate mixing matrix in large and even medium scales, for the multiple dominant case. However, our method solves the problem at least in the medium scale cases and maybe it can handel larger scales in comparison to other existing methods till now (our algorithm is capable of solving this problem when the averaged number of active sources is up to 5). As observed in the experimental results, all mixing vectors may be detected with good accuracy. However, some mixing vectors might not be found in few iterations, either because of lack of sufficient data, or because some of the actual mixing vectors are close to each other.

As was mentioned in the section 3 a medium value for σ must be considered and for very small σ the chance of error in finding a concentration increases. The subtractive clustering method does not need any prior information about the number of cluster centers, therefore, if the estimated subspace contains r ($r < k$) mixing vectors rather than k, the projected data on the positive normal hemisphere concentrate around r clusters and the clustering method detects r centers instead of k, thus our algorithm is somehow robust to these errors and consequently to σ.

Unfortunately, our algorithm is not efficient to some extent, because some of the mixing vectors are detected several times in order to find all vectors. This may lead to a greater number of iterations and consequently a longer run time. Finding an efficient method for estimating the mixing matrix is a future work.

References

1. Gribonval, R., Lesage, S.: A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In: Proceedings of ESANN'06, April 2006, pp. 323–330 (2006)
2. Zibulevsky, M., Pearlmutter, B.A.: Blind source separation by sparse decomposition in a signal dictionary. Neural Computation 13(4), 863–882 (2001)

3. Bofill, P., Zibulevsky, M.: Underdetermined blind source separation using sparse representations. Signal Processing 81, 2353–2362 (2001)
4. Georgiev, P.G., Theis, F.J., Cichocki, A.: Sparse component analysis and blind source separation of underdetermined mixtures. IEEE Transactions of Neural Networks 16(4), 992–996 (2005)
5. Babaie-Zadeh, M., Jutten, C.: Semi-blind approaches for source separation and independent component analysis. In: Proceedings of ESANN'06 (April 2006)
6. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution, Tech. Rep (2004)
7. Washizawa, Y., Cichocki, A.: on-line k-plane clustering learning algorithm for sparse comopnent analysis. In: Proceedinds of ICASSP'06, Toulouse (France), pp. 681–684 (2006)
8. Li, Y., Amari, S., Cichocki, A., Ho, D.W.C., Xie, S.: Underdetermined blind source separation based on sparse representation. IEEE Transactions on Signal Processing 54(2), 423–437 (2006)
9. Movahedi, F., Mohimani, G.H., Babaie-Zadeh, M., Jutten, C.: Estimating the mixing matrix in sparse component analysis (SCA) based on multidimensional subspace clustering. In: ICT'07, Malaysia (May 2007)
10. Chiu, S.: Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems, 2(3) (September 1994)

Appendix: Proof of Theorem 1

Consider a concentration subspace $\mathbf{B}$. Then, by definition, it is formed by a linear combination of k mixing vectors. Let $\mathbf{a}_{l_1} \cdots \mathbf{a}_{l_k}$ be these mixing vectors. Then for every point $\mathbf{x}$ in this subspace, we have $\mathbf{x} = \sum_{i=1}^{k} s_{l_i} \mathbf{a}_{l_i}$ where s_{l_i} $1 \leq i \leq k$ are the sources.

Lemma 1. *If $k \leq \frac{m}{2}$ and the mixing matrix is full rank then, each point in a concentration subspace ($\boldsymbol{B}$), for the sparsest solution, is almost always the linear combination of a set of k fixed mixing vectors. Precisely if $\boldsymbol{x} = \sum_{i=1}^{n} \tilde{s}_i \boldsymbol{a}_i$ then $s_i = 0$ for $i \in \{1, 2, ..., n\} - \{l_1, l_2, ..., l_k\}$.*

To prove this lemma suppose that there is another set of mixing vectors $\{\mathbf{a}_{t_1} \cdots \mathbf{a}_{t_h}\}$ and real valued variables $s'_{t_1}, \cdots, s'_{t_h}$ such that $\mathbf{x} = \sum_{i=1}^{h} s'_{t_i} \mathbf{a}_{t_i}$. Then $\mathbf{x} = \sum_{i=1}^{k} s_{l_i} \mathbf{a}_{l_i} = \sum_{i=1}^{h} s'_{t_i} \mathbf{a}_{t_i}$ shows that the set $\{a_{l_1}, \cdots, a_{l_k}, a_{t_1}, \cdots, a_{t_h}\}$ is not linearly independent. $\mathbf{A}$ is assumed to be full rank thus each $\acute{k} \leq m$ different mixing vectors are linearly independent. From this comment we can conclude that $k + h > m$, moreover, $k \leq m/2$ (see section 1) thus $h > m/2$ and we have $h > k$. This is in contrast to our basic assumption that we want to find the sparsest solution to BSS problem. This lemma is in fact similar to the theorem of uniqueness of the sparsest solution [6].

Using the above lemma, the expected value of active sources in $\mathbf{B}$ is

$$\tilde{k} = \sum_{i=0}^{k} iP\{i \text{ sources from } l_1, \cdots, l_k \text{ active} | \text{ remaining } n-k \text{ sources inactive}\}$$

where $P\{\cdot\}$ denotes the probability and $\tilde{k}$ is the expected value of the number of active sources in a concentration subspace. Since the sources (and hence their activity status) are assumed to be independent, the above equation is reduced to:

$$\widetilde{k} = \sum_{i=0}^{k} iP\{i \text{ sources of } l_1, \cdots, l_k \text{ active}\} = \sum_{i=0}^{k} i\binom{n}{k}(1-p)^i p^{k-i}$$

This is the expected value of a binomial random variable and hence $\widetilde{k} = k(1-p)$.