Estimating the mixing matrix in underdetermined Spare Component Analysis (SCA) using consecutive Independent Component Analysis (ICA)

Adel Javanmard, Pedram Pad, Massoud Babaie-Zadeh, Christian Jutten

ESTIMATING THE MIXING MATRIX IN UNDERDETERMINED SPARSE COMPONENT ANALYSIS (SCA) USING CONSECUTIVE INDEPENDENT COMPONENT ANALYSIS (ICA)

A. Javanmard[1], *P. Pad*[1], *M. Babaie-Zadeh*[1] *and C. Jutten*[2]

[1] Advanced Communications Research Institute (ACRI) and Department of Electrical Engineering
Sharif University of Technology, Tehran, Iran
[2]Institute National Polytechnique de Grenoble (INPG), GIPSA-lab, Dept. of Images and Signals
Grenoble, France

Email : adeljavanmard@yahoo.com, pedram_pad@ee.sharif.edu, mbzadeh@yahoo.com, Christian.Jutten@inpg.fr

ABSTRACT

One of the major problems in underdetermined Sparse Component Analysis (SCA) is the appropriate estimation of the mixing matrix, $\mathbf{A}$, in the linear model $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, especially where more than one source is active at each instant of time (It is called 'multiple dominant problem'). Most of the previous algorithms were restricted to single dominant problem in which it is assumed that at each instant, there is at most one single dominant component. Moreover, because of high computational load, all present methods for multiple dominant problem are practical only for small scale cases (By 'small scale' we mean that the average number of active sources at each instant, k, is less than 5). In this paper, we propose a new method for estimating the mixing matrix, $\mathbf{A}$ for the large scale multiple dominant problem in SCA. Our main idea is to convert the underdetermined SCA problem into a series of determined problems, which can be solved by well-known methods like ICA. To do this, we combine both sparsity and independence assumptions to estimate the mixing matrix. Our method can solve high dimension problems in which k can be relatively large (about 8).

1. INTRODUCTION

In recent years, there has been a great interest on finding different algorithms for the problem of Blind Source Separation (BSS) due to its various applications [1, 2, 3]. In this problem, separating a set of mixed signals from their mixtures is investigated. In traditional form of this problem, there are prior information neither about the mixing system nor about the source signals, except their statistical independence. Moreover, the number of sources and the number of sensors are assumed to be the same i.e., determined problem. There exist some well-known algorithms for the determined BSS problem such as EASI [4], Eigenvalue Decomposition [1], Entropy Optimization [5] and Fast-ICA [1]. In the determined problems (i.e., where the number of sources and sensors are equal), the mixing matrix is a square matrix and may be invertible. So by finding the mixing matrix, $\mathbf{A}$, the source signals can be recovered easily by multiplying the mixture by the inverse of $\mathbf{A}$.

On the other hand, if we take the advantage of some weak prior information available about the source signals i.e., BSS becomes Semi-Blind Source Separation, the quality of separation can be significantly improved [6]. A prior information like 'sparsity' of the source signals leads to the powerful concept of Sparse Component Analysis (SCA) and is particularly efficient especially for underdetermined mixtures [7, 8]. A sparse signal is a signal whose most samples are nearly zero, and just a few percents take significant values. Consequently, at each instant ('time' slot), only a few number of sources have significant values (say they are 'active'), and most of them are almost zero (say they are 'inactive'). This prior information is important because of two reasons. Firstly, in contrast to traditional source separation methods, it permits source separation for the case in which the number of sources exceeds the number of sensors i.e., underdetermined problems [8, 9, 10, 11, 12]. Secondly, it is a very practical assumption for many sources: even if the sources are non-sparse in time domain, they may be sparse in another (linear) transformed domain. For instance, the speech sources may not be sparse enough in time, but they are usually sparser in time-frequency (using Short-Time Frequency Transform=STFT) or time-scale (using wavelet packet) domains [7].

The previous methods for extracting sparse components of the mixtures, mainly relies on the sparsity of sources, and not on their independence[8]. *The main novelty of our method is that it combines both sparsity and independence assumptions and uses consecutive determined ICA transforms to estimate the mixing matrix in the underdetermined SCA problem.*

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

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{1}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathbb{R}^{n\times m}$ is the mixing matrix, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T] \in \mathbb{R}^{m\times T}$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] \in \mathbb{R}^{n\times T}$ are the matrices of m sources and n observed signals, respectively and $m > n$. Each column of $\mathbf{X}$ and $\mathbf{S}$ corresponds to an instant of time and T is the number of time samples. The sparsity assumption implies that, at each instant of time, there are a few significant values in the corresponding columns of $\mathbf{S}$ (i.e., a few active sources). Each column of the mixing matrix, i.e., each $\mathbf{a}_i, 1 \le i \le m$, is called a 'mixing vector'. These notations and terminologies are also used in [13, 14].

Although the word 'time' is used in the above paragraphs ('time' samples, instant of 'time' and 'time' slot), and will be used in the continuation of this paper, the above model may be in another domain, in which the sparsity assumption

This work has been partially supported by Iran National Science Foundation (INSF) under contract number 86/994, and also by center for International Research and Collabration (ISMO) and French embassy in Tehran in the framework of a GundiShapour program.

holds. To see this, let $\mathscr{T}$ be a linear 'sparsifying' transform (like STFT or wavelet packet transforms for speech signals), and the mixing system is stated as $\mathbf{X} = \mathbf{AS}$ in the time domain. Then, we have $\mathscr{T}\{\mathbf{X}\} = \mathbf{A}\,\mathscr{T}\{\mathbf{S}\}$ in the transformed domain, and because of the sparsity of $\mathscr{T}\{\mathbf{S}\}$, it is in the form of an SCA problem [13].

Denote the average number of active sources at each instant by k. In fact, if the probability of activity of a source is denoted by p, then $k = mp$. In single dominant SCA problems in which there is at most one dominant source at each instant of time, k is less than or approximately equal to one. In these problems, the data points are geometrically concentrated around the directions of the mixing vectors. This can be easily seen from the fact that at each instant of time,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = \sum_{i=1}^{m} s_i(t)\,\mathbf{a}_i \qquad t = 1,2,\ldots,T \qquad (2)$$

and for most instants of time, there exist just one dominant s_i and the others are almost zero. In the multiple dominant SCA problems in which there can be more than one dominant source at each instant of time, k is greater than one. In these problems, the observed data concentrate around k-dimensional subspaces spanned by a set of k mixing vectors. Throughout this paper, the subspaces spanned by r mixing vectors, for an arbitrary r, are called 'r-concentration' subspaces. This terminology is also used in [13, 14]

The SCA problem is usually solved in two steps. The first step is the estimation of the mixing matrix ($\mathbf{A}$), and the second step is the recovery of the source signals ($\mathbf{S}$) by knowing the mixing matrix. Note that in the underdetermined case, in which the number of sources exceeds the number of sensors, the mixing matrix is not square and thus not invertible any more. So these two problems are not identical. In other words, knowing the mixing matrix does not directly result in the recovery of the sources [15]. In this paper, we introduce an algorithm to solve only the problem of the estimation of the mixing matrix.

Most of the current algorithms are restricted to the single dominant problem and as far as we know, all existing algorithms for multiple dominant cases are capable only for small scale problems [7, 8, 12]. In fact, because of high computational load, these algorithms are not efficient for cases in which k is large (in all of them k is smaller than 5 [14, 16]).

In this paper, we express a method which is capable for problems with larger k. However, note that to ensure the uniqueness of the sparse solution of an underdetermined system of linear equations, k should be assumed smaller than $n/2$ [10, 17]. The main idea of this paper is to show that the underdetermined SCA problem can be converted into a series of determined problems, which can be solved by well-known methods like ICA.

Throughout the paper, we suppose that the sources are sparse and independent and the probability of activity is the same for all of them. These are very practical assumptions for many sources. Moreover, we assume that each subset of m columns of $\mathbf{A}$ is linearly independent.

The paper is organized as follows. The following section explains the main idea of the paper. We investigate the method of finding the concentration subspaces and estimating the mixing vectors in Sections 3 and 4, respectively. The final algorithm is expressed in Section 5. Experimental Results will be presented in Section 6, before conclusions and future works in Section 7.

2. THE MAIN IDEA

The main idea of our method for estimating the mixing matrix $\mathbf{A}$, in the underdetermined SCA problem is as follows:

At each step we find one of the r-dimensional concentration subspaces $\mathbf{B}$ and all data points which belong to it (r depends on dimensions of the problem, i.e., the number of sources, sensors and also on k). By confining to this subspace, we are faced to a determined problem in which all data points are in an r-dimensional space and the mixing matrix consists of those mixing vectors which belong to this subspace. Then all mixing vectors in $\mathbf{B}$ will be found by applying ICA method to this determined problem.

It is worth mentioning that before applying ICA to the subspace $\mathbf{B}$, we choose a suitable basis for $\mathbf{B}$ such that the data points belonging to $\mathbf{B}$ have an r-entry representation in this basis. Thus the mixing matrix becomes an $r \times r$ one in this coordination. (Usually the latter is implemented using Principle Component Analysis (PCA) as a preprocessing block for ICA [4]. However, we have not used PCA for this part of our algorithm).

The algorithm is run until all mixing vectors in the origin problem are obtained. Our policy to find at least one new concentration subspace at each step, is of great importance. This is investigated in the following sections.

So, in our method, both sparsity and independence assumptions are taken into account. The sparsity assumption results in concentration of the data points around the subspaces while the independence assumption leads to a good performance for ICA, applied to the corresponding subspace, in finding the mixing vectors.

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

– *Step 1:* Find a new concentration subspace $\mathbf{B}$. Following the idea proposed in [13, 14], a concentration subspace can be found by maximizing a cost function (see next section) using steepest ascent algorithm.

To guarantee to find a *new* concentration subspace at each iteration, we first omit data points which belong to at least one of the already found concentration subspaces and then apply the steepest ascent algorithm initialized by a randomly different starting point. The data points belonging to a typical concentration subspace are those whose the contribution to the cost function is larger than a specific value, α.

– *Step 2:* Determine all data points in the initial problem which belong to $\mathbf{B}$ (Note that all data points are taken into account in this step even those that were omitted in ***Step 1*** to find the new subspace $\mathbf{B}$). Then, run ICA algorithm to find the mixing vectors belonging to $\mathbf{B}$.

– *Step 3:* Considering that some of these mixing vectors may have been already found, at this step we use a clustering-merging algorithm to merge these mixing vectors with the ones obtained before. During this clustering-merging algorithm, a weight is defined for each estimated mixing vector which is proportional to the number of times that this vector (or a correlated vector to it) has been found.

– *Step 4:* If the number of iteration is l, stop the algorithm and sort the estimated mixing vectors in descending order according to their weights and choose the first m vectors as estimated mixing vectors, else go back to ***Step 1*** and continue.

In this paper, the number of sources is assumed to be known in advance.

3. FINDING CONCENTRATION SUBSPACES

Each r-dimensional subspace can be represented by an $n \times r$ matrix, whose columns form an orthonormal basis of the subspace. In this paper, we do not distinguish between a subspace and its matrix representation. Let $\mathbf{B} \in \mathbb{R}^{n \times r}$ be the matrix of an arbitrary r-dimensional subspace. The following cost function has been proposed in [13] to detect whether $\mathbf{B}$ is an r-concentration subspace or not:

$$f_\sigma(\mathbf{B}) = \sum_{i=1}^{T} \exp\left(\frac{-d(\mathbf{B}, \mathbf{x}_i)^2}{2\sigma^2}\right) \tag{3}$$

where $d(\mathbf{B}, \mathbf{x}_i)$ is the distance of $\mathbf{x}_i$ from the subspace represented by $\mathbf{B}$ and σ is a very small positive number. For small values of $d(\mathbf{B}, \mathbf{x}_i)$ compared to σ, the term

$$\exp\left(\frac{-d(\mathbf{B}, \mathbf{x}_i)^2}{2\sigma^2}\right) \tag{4}$$

is about 1 and for large values of $d(\mathbf{B}, \mathbf{x}_i)$, it is nearly zero. Thus for sufficiently small values of σ, the above cost function is approximately equal to the number of data points close to $\mathbf{B}$. Moreover, if the set of points are concentrated around several different r-dimensional subspaces, f_σ has a local maximum near the basis of each of them. We apply the same idea of [13] for finding an r-concentration subspace. That is maximizing the function f_σ for a sufficiently small σ, using a steepest ascent method. However maximizing f_σ for small σ's faces to many local maxima. The idea of [13], is to use a decreasing sequence of σ to obtain an accurate estimation. In other words, the location of the maximum of f_σ for larger σ's are interpreted as the initial guess about the location of the maximum for smaller σ.

4. ESTIMATING MIXING VECTORS AND MIXING MATRIX

Consider an r-dimensional concentration subspace $\mathbf{B}$ which is found by the method, mentioned in the previous section. The data points whose the contribution to the cost function $f_\sigma(\mathbf{B})$ i.e., (4), is greater than a specific threshold α are assumed to belong to $\mathbf{B}$.

$\mathbf{B}$ is an r-dimensional subspace and ideally it contains r mixing vectors. So by confining the problem to this subspace, we are faced to a determined BSS, since $r \leq n$. We apply ICA method to find the r mixing vectors in this determined problem. Denote the set of these mixing vectors by Ω. Then we omit all the data points belonging to $\mathbf{B}$ and find another r-dimensional concentration subspace $\mathbf{B}'$. Considering that we have omitted points of $\mathbf{B}$, the subspace $\mathbf{B}'$ will be different from the subspace $\mathbf{B}$ and this approach is likely to find a new mixing vector at each iteration.

Next, determine all data points belonging to $\mathbf{B}'$. Using ICA in the subspace $\mathbf{B}'$, the mixing vectors belonging to it are recovered. The obtained mixing vectors are added to the set Ω.

Since in the determined BSS, corresponded to the subspace $\mathbf{B}'$, all data points in the entire problem are taken into account, $\mathbf{B}$ and $\mathbf{B}'$ may contain some common mixing vectors. We perform the following clustering algorithm in order to partition the obtained vectors into subsets which are correlated internally, but different from each other. Then we merge the vectors in each subset.

The clustering-merging algorithm is as follows:

Suppose that $\Omega = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t\}$ is a set containing t vectors.

– *Step 1:* Normalize the vectors in Ω and define a weight to each of them. Initially, weight of each vector is 1.

– *Step 2:* Find a pair of vectors which are most correlated, i.e., the angle between them is the least among all other pairwise angles. If their angle is less than a specific threshold θ, put this couple in a subset. Otherwise stop the algorithm.

– *Step 3:* Merge the vectors of this subset, i.e., replace them by their weighted-mean vector. Set the weight of newly obtained vector to sum of the weights of the two vectors. In fact, the weight of a vector $\mathbf{v}$ indicates the number of vectors in Ω which are merged together to make $\mathbf{v}$.

– *Step 4:* Go to ***Step 2***.

By applying clustering-merging algorithm we provide a set of non-correlated vectors corresponding to different mixing vectors. Then we continue the above procedure by finding another concentration subspace. The algorithm is run for l iterations where l is sufficiently large such that we get a good approximation of the m mixing vectors. Then, we finish by sorting the obtained vectors in descending order according to their weights and by choosing the first m vectors as estimated mixing vectors. Intuitively, vectors with large weight are those which have been found more times than the other vectors. It means that they belong to more concentration subspaces comparing to the other vectors. So they are more likely to be a good estimate of mixing vectors.

5. FINAL ALGORITHM

The final algorithm based on the main idea, presented in section 2, is given in Fig. 1. As seen in the algorithm, in order to find an r-concentration subspace, the final value of the previous estimation is used for initialization of the next steepest ascent. By choosing a slowly decreasing sequence of σ, we try to escape from getting trapped into local maxima. This phenomenon i.e., concentration of datapoints around the concentration subspaces, is a consequence of sparsity assumption.

Then by applying ICA to the found concentration subspace, the mixing vectors belonging to this subspace are recovered. Here, the independence assumption of the data points plays the key role.

6. EXPERIMENTAL RESULTS

In this section, we present a simulation to justify the algorithm. It has been already shown that, a necessary condition for separating source signals using the ICA method (without temporal prior) is that the sources must have a non-gaussian distribution [4]. Because of this, in this simulation, sparse sources are independently and identically distributed, generated according to the Bernoulli-Uniform model. In other words, sources are active with probability p and are inactive with probability $1-p$. In the active case, their value has a uniform distribution in the interval $(-A_{\text{on}}, A_{\text{on}})$ and in inactive case it is a uniform signal with amplitude in the interval $(-A_{\text{off}}, A_{\text{off}})$. Consequently:

$$s_i \sim p\mathbf{U}(-A_{\text{on}}, A_{\text{on}}) + (1-p)\mathbf{U}(-A_{\text{off}}, A_{\text{off}}) \tag{5}$$

- Initialization:
  1. Assume an appropriate value for the free parameters of the problem (r, l, α and θ).
  2. Choose a suitable decreasing sequence for σ, $[\sigma_1 \dots \sigma_K]$.
  3. Let $\mathbf{B}^{(0)}$ be a random subspace (an $n \times r$ matrix with orthonormal columns).
  4. $\Omega := \emptyset$.
- For $\ell = 1, \dots, l$
  1. For $k = 1, \dots, K$:
     a) Let $\sigma = \sigma_k$.
     b) Maximize $f_\sigma(\mathbf{B})$ with steepest ascent algorithm in several steps:
        - Initialization: $\mathbf{B}_0^{(k)} := \mathbf{B}^{(k-1)}$
        - Loop:
          i) Set $\mathbf{B}_{j+1}^{(k)} = \mathbf{B}_j^{(k)} + \mu \nabla f_\sigma$.
          ii) Choose an orthonormal basis for range space of $\mathbf{B}_{j+1}^{(k)}$.
          iii) If $\left| f_\sigma\left(\mathbf{B}_{j+1}^{(k)}\right) - f_\sigma\left(\mathbf{B}_j^{(k)}\right) \right| \geq \varepsilon$, go to Loop.
     c) Denote the present subspace by $\mathbf{B}^{(k)}$.
  2. Consider all points in the entire set of data points which belong to $\mathbf{B}^{(K)}$ i.e., the points $\mathbf{x}_i$ that $\exp\left(\frac{-d(\mathbf{B}^{(K)}, \mathbf{x}_i)^2}{2\sigma^2}\right) < \alpha$.
  3. Apply ICA method to the points belonging to $\mathbf{B}^{(K)}$ to find the mixing vectors which are in subspace $\mathbf{B}^{(K)}$. Add the obtained mixing vectors to the set Ω.
  4. Apply the clustering-merging algorithm, with parameter θ, to the set Ω.
  5. Omit all data points belonging to the concentration subspaces found up to now.
- Sort the obtained vectors in descending order according to their weights and choose the first m vectors as estimated mixing vectors.

Figure 1: The final algorithm.

where $\mathbf{U}(a,b)$ indicates the uniform distribution in interval (a,b). To ensure the sparsity assumption, the conditions $A_{\text{on}} \gg A_{\text{off}}$ and smallness of p are applied. In the simulation, the value $A_{\text{on}} = 1$ and $A_{\text{off}} = 0.01$ have been used. Also each entry of the mixing matrix is generated uniformly random in the $[-1,1]$ interval.

We use the CPU time as a measure of complexity. Although the CPU time is not an exact measure, it can give us a rough estimation of the complexity. The simulation is performed in MATLAB 7.5 under Linux OS, using an Intel Pentium IV 2.4GHz processor.

Following simulation is performed to show the capability of our algorithm for large scale problems. In this simulation, $m = 40$, $n = 20$, $k = 8$ (or equivalently $p = 0.2$) and $T = 100000$ (Noting that m, n and k are relatively large, the requirement of large T is predictable). The parameters are chosen as $r = 9$, $\theta \simeq 22^\circ$, $\alpha = 0.1$, σ-sequence$= \{0.3, 0.2\}$, $\mu = 1$, $\varepsilon = 10^{-3}$ and $l = 40$. For implementing ICA in the proposed algorithm, detailed in the previous section, the FastICA algorithm has been used.

The entire process took about 30 minutes. At our best knowledge, there is no algorithm able to estimate the mixing vectors in a realizable time, especially with problem of this dimension ($m = 40$ and $n = 20$) and with a reduced sparsity

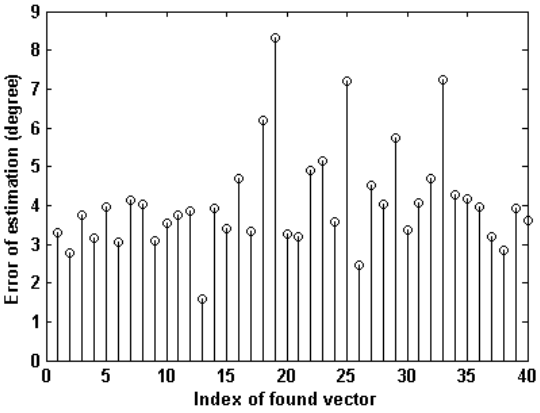

Figure 2: Error of the estimated vectors (in degree) in a simulation of the algorithm proposed in section 5 for the case $m = 40$, $n = 20$, $k = 8$.

($k = 8$).

The accuracy of the algorithm is measured by calculating the angle between each estimated vector and its corresponding actual mixing vector. As depicted in Fig. 2, the largest angle is less than 9 degree indicating an accurate estimation of mixing vectors.

7. CONCLUSION AND FUTURE WORKS

In this paper, we introduced a new method for solving multiple dominant SCA problems. This algorithm combines both sparsity and independence assumptions to estimate the mixing matrix. As it is verified by simulations, the proposed algorithm is capable in handling large scale problems in which the average number of active sources in each instant of time, k, is large (up to the border of sparsity). As observed in the experimental results, all mixing vectors may be detected with high accuracy (Fig. 3). Also, the problem of having small number of data samples $\mathbf{x}_i$'s can be compensated by increasing l, i.e., finding more concentration subspaces. An important property of this algorithm is that the knowledge of k is not essential.

Another potential advantage of our method is that we can implement the algorithm in such a way that the knowledge of m is not required and we estimate it through the algorithm. As it can be seen in Fig. 4, there is a jump in the weights of the elements of Ω from m to $m+1$. This phenomenon suggests that for estimating m we can repeat the algorithm proposed in Section 5 until a jump in the weights of elements of Ω appears.

As a future work, we are working on the estimation of l, r and σ-sequence theoretically from dimensions of the problem (such as n, m, k and T).

REFERENCES

[1] A. Hyvarinen, J. Karhunen, and E. Oja, "Independent component analysis," *John Wiley and Sons*, 2001.

[2] A. Cichocki and S.-I. Amari, "Adaptive blind signal and image processing: Learning algorithms and applications," *John Wiley and sons*, 2002.

[3] M. Zibulevsky, B. Pearlmutter, P. Bofill, and P. Kisilev, *Independent Component Analysis: Principles and*

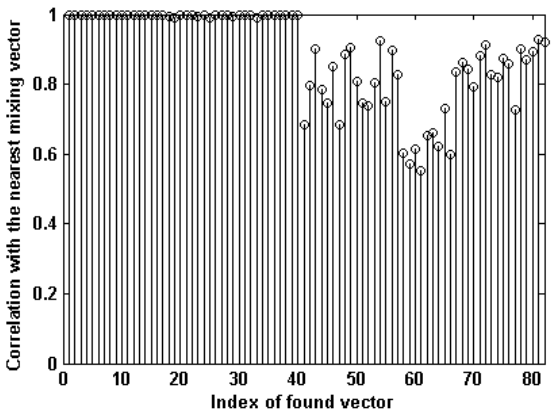

Figure 3: Correlation of the found vectors by the algorithm with the nearest mixing vector in the simulation proposed in Fig. 2.

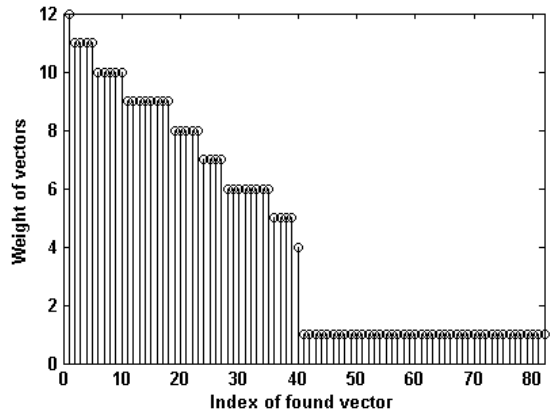

Figure 4: Weights of the found vectors (determined by clustering-merging algorithm proposed in section 4) in the same simulation as in Fig. 3.

Practice, chapter Blind Source Separation by Sparse Decomposition, 2001.

[4] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proceedings of the IEE*, vol. 9, no. 10, pp. 2009–2025, October 1998.

[5] A. Taleb and C. Jutten, "Entropy optimization - application to blind source separation," in *Proceedings of the 7th International Conference on Artificial Neural Networks, ICANN'97*, Switzerland, October 1997, pp. 529–534.

[6] M. Babaie-Zadeh and C. Jutten, "Semi-blind approaches for source separation and independent component analysis," in *Proceedings of ESANN'06*, April 2006, pp. 301–312.

[7] R. Gribonval and S. Lesage, "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.

[8] P. G. Georgiev, F. J. Theis, and A. Cichocki, "Blind source separation and sparse component analysis for over-complete mixtures," in *Proceedings of ICASSP'04*, Montreal (Canada), 2004, pp. 493–496.

[9] D. L. Donoho and M. Elad, "Optimally sparse representation from overcomplete dictionaries via l^1 norm minimization.," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, March 2003.

[10] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," *Tech. Rep.*, 2004.

[11] Y. Li, A. Cichocki, and S. Amari, "Sparse component analysis for blind source separation with less sensors than sources," in *Proceedings of ICA2003*, Nara, Japan, April 2003, pp. 89–94.

[12] P. Bofill and M. Zibulevsky, "Underdetrmined blind source separation by sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, November 2001.

[13] F. Movahedi, G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "Estimating the mixing matrix in sparse component analysis (sca) based on partial k-dimensional subspace clustering," *to appear in Neurocomputing, Special Issue on Advances in Blind Signal Processing.*

[14] N. Noorshams, M. Babaie-Zadeh, and C. Jutten, "Estimating the mixing matrix in sparse component analysis based on converting a multiple dominant to a single dominant problem," in *Proceedings of ICA2007, LCNC 4666*, London, September 2007, pp. 397–405.

[15] A. A. Amini, M. Babaie-Zadeh, and C. Jutten, "A fast method for sparse component analysis based on iterative detection-projection," in *Proceeding of 26th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt)*, Paris, France, July 2006.

[16] Y. Washizawa and A. Cichocki, "On-line k-plane clustering learning algorithm for sparse component analysis," in *Proceedinds of ICASSP'06*, Toulouse (France), 2006, pp. 681–684.

[17] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. on Information Theory*, vol. 49, no. 12, pp. 3320–3325, December 2003.