AN EFFICIENT K-SCA BASED UNERDETERMINED CHANNEL IDENTIFICATION ALGORITHM FOR ONLINE APPLICATIONS

Ehsan Eqlimi[1, 2] *and Bahador Makkiabadi*[*, 1, 2]*, Member, IEEE*

[1]Department of Medical Physics & Biomedical Engineering, School of Medicine, Tehran University of Medical Sciences (TUMS), Tehran, Iran
[2]Research Centre of Biomedical Technology and Robotics (RCBTR), Institute for Advanced Medical Technologies (IAMT), Tehran, Iran
E-mails: Eghlimi@razi.tums.ac.ir, *Corresponding Author: B-makkiabadi@tums.ac.ir

ABSTRACT

In a sparse component analysis problem, under some non-strict conditions on sparsity of the sources, called k-SCA, we are able to estimate both mixing system ($\boldsymbol{A}$) and sparse sources ($\boldsymbol{S}$) uniquely. Based on *k*-SCA assumptions, if each column of source matrix has at most $N_x - 1$ nonzero component, where N_x is the number of sensors, observed signal lies on a hyperplane spanned by active columns of the mixing matrix. Here, we propose an efficient algorithm to recover the mixing matrix under *k*-SCA assumptions. Compared to the current approaches, the proposed method has advantages in two aspects. It is able to reject the outliers within subspace estimation process also detect the number of existing subspaces automatically. Furthermore, to accelerate the process, we integrate the "subspaces clustering" and "channel clustering" stages in an online scenario to estimate the mixing matrix columns as the mixture vectors are received sequentially.

Index Terms**—** Underdetermined Blind Identification, Sparse Component Analysis (SCA), *k*-SCA and Subspace Clustering

1. INTRODUCTION

Blind Source Separation (BSS) on an instantaneous linear mixing system could be formulated as follows

$$\mathbf{X} = \mathbf{AS} \qquad (1).$$

Where $\mathbf{X} = [\mathbf{x}(0), \ldots, \mathbf{x}(T-1)] \in R^{N_x \times T}$ includes the mixture data at all-time instants ($t = 0, \ldots, T-1$), $\mathbf{A} = [\mathbf{a}_1, \ldots. \mathbf{a}_{N_s}] \in R^{N_x \times N_s}$ is unknown full-column rank mixing matrix and $\mathbf{S} = [\mathbf{s}(0), \ldots, \mathbf{s}(\mathrm{T}-1)] \in R^{N_s \times T}$ includes unknown underlying source signals in different time instants (N_s and N_x are the number of sources and sensors respectively). The problem of BSS consists of decomposing data set $\mathbf{X}$ into $\mathbf{A}$ and $\mathbf{S}$ such that we have a priori knowledge about $\mathbf{S}$ [1]. Basically, a BSS problem is a two - stage procedure. The first stage consists of identifying the mixing matrix and the second stage is recovering the source signals. In this paper, we focus on the identification step.

In 2005 *Georgiev et al* [2, 3] proposed *k*-sparse component analysis (*k*-SCA) algorithm to solve the Underdetermined BSS (UBSS) that assumes the source signals are ($N_x + 1$) sparse i.e. there are at most ($N_x - 1$) active sources in each time instant. Indeed, UBSS problems are solvable as long as the number of mixture is greater than the number of active sources in each time instant. During these years, many UBSS algorithms have been proposed based on *k*-SCA assumptions in the literature. Most of these methods take advantage of subspace clustering approach. Actually, the observed signals are clustered to M clusters, where M is the maximum number of possible subspaces, in order to estimate the mixing matrix.

Georgiev et al applied a two-layer clustering approach to estimate $\mathbf{A}$. As the first step, the columns of $\mathbf{X}$ are clustered in $C_{N_s}^{N_x-1}$ groups such that the span of the components of each cluster generates different $N_x - 1$ dimensional subspaces. At second stage, a similar manner is applied to the normal vectors of each subspace. Actually, they are clustered in N_s groups such that each normal vector to the subspaces lies on the subspaces of normal vectors. Finally, the normal vectors of different subspaces are estimations of the mixing matrix columns (up to permutation and scaling). They proposed an algorithm called "subspace clustering" to put into practice their theory based on a minimization framework [3].

In 2009 *Zhaoshui He et al* [4, 5] proposed a novel approach to implement *Georgiev*'s theory called "*k*-hyperplane learning clustering (*k*-HLC)" based on a new complex space distance combined with the *k*-Eigenvalue decomposition (*k*-EVD) process. Moreover, *k*-HLC is able to detect the number of hyperplanes.

In 2010, *Wen Yang et al* [6] suggested a novel anti-noise approach by definition of hyperplane membership functions to estimate more accurate mixing matrix. Recently, *Jiechang Wen et al* [7] have proposed an extension of the normal vector clustering prototype that is combined with a new fuzzy *k*-EVD based algorithm. The authors of

This study was part of a PhD. thesis supported by Tehran University of Medical Sciences (TUMS); Grant No. : 94-01-30-28327.

the work presented in [8], in order to find the normal vectors of the subspaces, suggested a new simple algorithm which tries to find all $N_x - 1$ linearly independent column vectors of **X**. Then, they cluster the normal vectors of achieved column vectors which span different subspaces. *Yoshikazu Washizawa et al* [9] introduced an online adaptive clustering that is capable of learning individual hyperplane.

In 2007, *Theis et al* [10, 11] implemented a robust SCA algorithm based on generalized Hough transform. Although their approach takes advantage of the different concept compared to other works in the literature but the angular resolution limits the accuracy of the mixing matrix.

In this paper, we propose a new approach to implement *Georgiev*'s theory and detect the number of hidden subspaces using a subspace selective search based algorithm. Most of the subspace clustering based methods might not converge to accurate estimations due to outliers. Here, outliers are defined as the measurement vectors with more than $N_x - 1$ active columns or the vectors that are not belonged to the span of **A**.

The proposed method is able to reject the outliers by applying a novel selective algorithm in order to achieve accurate estimation of mixing matrix. Furthermore, here an online approach is considered in contrast to the two–fold offline methods in the literature. Here, as the first step, we discuss the basic k-SCA concept in Section 2. Our online selective approach to identify mixing matrix **A** and its clustering prototype is presented in Section 3. Simulation examples are described in Section 4. The conclusions and discussion are provided in Section 5.

2. *K*- SCA PROBLEM FORMULATION

Based on Eq.1, the adopted version of instantaneous mixing system under k-SCA assumption, in a vector-wise scheme, can be formulated as follows

$$\mathbf{x}(t) = \sum_{j=1}^{k} \mathbf{a}_j \, \mathbf{s}_j(t) \quad (2).$$

Based on new formulation, observed signals in moment t is built by combination of k weighted version columns of **A**, where k is the number of active sources in each time instant.

The k-SCA assumptions are listed as follows[2].

***A*1.** Each square $N_x \times N_x$ submatrix of **A** is nonsingular.

***A*2.** Source matrix **S** has at most $N_x - 1$ active (nonzero) source at each column.

***A*3.** Source matrix is able to excite the all possible subspaces sufficiently. Actually, it has at least N_x columns such that each of them has inactive sources in same places and $N_x - 1$ of them are linearly independent.

Based on *Georgiev*'s proof, as long as $k < N_x$, Eq. 1 has a unique solution. Let's assume k=N_x-1, according to Eq. 2 and based on k-SCA assumptions two concept are elicited as follows. First, each subspace $\mathbf{H}_i$ is spanned by k columns of **A** and every k columns of **X** are linear independent i.e.:

$$\mathbf{H}_i = \boldsymbol{Span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} \quad (3).$$

Consequently, all number of possible subspaces is known as $C_{N_s}^{k}$. Second, jth column of **A** (i.e. $\mathbf{a}_j$) lies in the intersection of $C_{N_s-1}^{k-1}$ subspaces, produced by those columns of **A,** which involve $\mathbf{a}_j$.

3. METHOD

The standard Underdetermined Blind Identification (UBI) problem under k-SCA assumptions could be tackled by two below steps [2].

1. Find the normal vectors of hidden subspaces $\boldsymbol{w}_i \in R^{N_x}$ when the active source in each moment is at most $k = N_x - 1$.
2. Estimate the normal vectors $\hat{\mathbf{a}}_b$ ($b = 1, \dots, N_s$) of each subspace, where $\hat{\mathbf{A}}$ with columns of $\hat{\mathbf{a}}_b$ is an estimation of **A**.

Different clustering methods with different performances are developed to be employed in Step 2 [2-7, 9-11]. One of the most challenging problems in this step is due to existing outliers data involved in different clusters. Here, in order to promote the performance of the process listed on Step 2, we have developed a selective algorithm which will be explored in the next subsections.

3.1 Subspace Selective Search

The performance of many algorithms mentioned in Section 1 might not be desirable due to the outliers. Hence, here a selective algorithm has adopted to reject the outliers. One of the ways to find the subspaces is finding and clustering the normal vectors of them. In this regard, it is proper to perform Eigen-value decomposition (EVD), in analogy with k-EVD [5] and k-HLC [4] frameworks, to find the normal vectors as eigenvector corresponding the smallest Eigen-value. The key difference between our algorithm and other methods in the literature is that the proposed method performs a selective search to choose the proper subspaces.

In addition, the number of sources could be unknown in some practical applications. Hence, it will be impossible to determine the number of the hidden subspaces. Most existing methods assume that the number of subspaces is computable. Few of them, such as the method proposed by *Zhaoshui He et* al [4] assume that this number is unknown. They tackle this problem by overestimating the number of the subspaces which makes their performance dependent to overestimating ratio. In contrast, our proposed framework is capable of measuring the number of existing subspaces with no overestimating scheme.

Subspace selective search (so called S^3) tries to compute altered versions of normal vectors of each subspace then compute the ultimate normal vector in a selective scenario summarized in Algorithm 1. Here, we describe the S^3 procedure for current selected N_x vectors that are chosen randomly. Note that if the selection criterion (in Step 2 of Algorithm 1) is satisfied, S^3 finds a proper subspace and its normal vector (as its representative). In the cases with some vectors of different subspaces the output is null (showing that the input vectors do not lie on one subspace). We need to operate S^3 several times on the randomly selected vectors of $\mathbf{X}$ which will be explained in section 3.4.

Algorithm 1 Subspace Selective Search(S^3)

Task: Selective searching for identifying the proper subspaces.
Initialization: Th_1, a threshold for selection.
Input: $\tilde{\mathbf{X}}$, a $N_x \times N_x$ submatrix from $\mathbf{X}$.
Output: w: The candidate for the normal vectors of subspaces or null.
Step 1: Build $\boldsymbol{R} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T$ and obtain its Eigenvalues i.e. λ_i.
Step 2: $if\ \frac{|\lambda_{1\min}|}{|\lambda_{\max}|} < Th_1$ then $\leftarrow \mathbf{v}_1$, where λ_1 and λ_{N_x} are the smallest and largest Eigenvalues respectively and $\mathbf{v}_1$ is the eigenvector that correspond to λ_1.
Step 3: else $\mathbf{w} \leftarrow null$.

3.2 Bidding - based Clustering Algorithm

In the noise free cases, selective method to find the subspaces (Algorithm 1) detects all ($C^k_{N_s}$) subspaces without any prior information about the number of subspaces. With this scenario, each subspace and its normal vector will be turned up many times. Therefore, an online clustering algorithm (with unknown number of clusters) is needed to cluster the different achieved normal vectors related to different subspaces. By using this strategy, both subspace selection and clustering could be performed simultaneously. To this aim, we adopt a bidding process that tries to decide in assigning any input normal vector to the one of existing centroids or producing a new cluster from it. In this process, we consider the compactness of each cluster members combined with most discrimination among the different clusters using the absolute cosine distance (ACD). This distance is defined between two vectors (e.g. for $\mathbf{x}_i$ and $\mathbf{x}_j$) as follows

$$ACD_{ij} = 1 - |cos\theta| \quad where\ cos\theta = \frac{\mathbf{x}_i{}^T\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|} \qquad (4).$$

Algorithm 2 Bidding based- Clustering Algorithm (BBC)

Task: Online Generation Clustering.
Initialization: Th_2, a threshold for compactness.
Input: $\mathbf{w}_i$: the candidates.
Output: C: the updated centroids.
Step 1: $\forall\ \mathbf{c}_j \in C(1,\dots,J)$ find D_i=min ACD_{ij} where $ACD_{ij} = 1 - \left|\frac{\mathbf{w}_i^T\mathbf{c}_j}{\|\mathbf{w}_i\|\|\mathbf{c}_j\|}\right|$ is the bid distance and $\mathbf{w}_i \leftarrow sign(\frac{\mathbf{w}_i^T\mathbf{c}_j}{\|\mathbf{w}_i\|\|\mathbf{c}_j\|})\mathbf{w}_i$.
Step 2: $if\ \exists\ \mathbf{c}_l \in \mathbf{C}$ such that $D_i < Th_2$ then allocate $\mathbf{w}_i$ to $\mathbf{c}_j$ as closest
Step 3: Else allocate $\mathbf{w}_i$ as a new cluster (i.e. $\mathbf{c}_{J+1} \leftarrow \mathbf{w}_i$).

3.3 Channel Selective Search

In order to detect the subspaces and channels (the columns of mixing matrix) simultaneously, we need a channel estimation process after detecting the subspaces. As mentioned before, the most subspace-based methods find the normal vectors of the known number ($C^k_{N_s}$) of subspaces in their first step. Then, the channels are identified based on a similar process separately. In fact, ($C^{k-1}_{N_s-1}$) subspaces are intersected by a column of $\mathbf{A}$ where k is the number of active source in each moment. Let us assume $N_x = 3$ and $N_s = 5$, implying there shall be $C^2_5 = 10$ subspaces ($k = N_x - 1 = 2$) that could be represented by their normal vectors. Therefore, we need all ($C^{2-1}_{5-1} = 4$) subspaces to find each channel that it is orthogonal to all 4 normal vectors.

Algorithm 3, named channel selective search (so called CS^2) illustrates our scheme for online channel estimation. CS^2 is in fact equivalent to the third step in the subspace based k- SCA algorithm that was mentioned in the beginning of section 3. However, we propose a novel method of channel identification whose structure is different from the existing subspace-based algorithms.

We shall consider that we could proceed to estimate a channel with as little as N_x detected subspaces. We shall locate these N_x vectors to make a matrix $\tilde{\mathbf{C}}_j \in R^{N_x \times N_x}$. In fact, CS^2 implements, in analogy with Algorithm 1, a selective algorithm to disentangle the desired orthogonal vectors from undesired ones. Here, the major difference with Algorithm1 is that the proposed algorithm rejects the few degenerative solutions which might be achieved by finding the eigenvector corresponding to the smallest Eigen-value of the covariance matrix of $\tilde{\mathbf{C}}_j$only. Generally, degenerative solutions are the results of having more than one Eigenvalues very close to zero. To mitigate this problem we could reject cases with the second smallest Eigenvalues very close to zero. Nevertheless, still some spurious vectors might be detected. These spurious solutions could be encountered when we have one or more sub-matrices $\hat{\mathbf{C}}_i \in R^{N_x \times N_x - 1}$ in $\tilde{\mathbf{C}}_j$ with more than one zero (or very close to zero) Eigenvalues. We address these issues by finding the applying the selective procedure to all sub-matrices $\hat{\mathbf{C}}_i \in R^{N_x \times N_x - 1}$ in $\tilde{\mathbf{C}}_j$. More details in this regards are given in Algorithm 3.

Algorithm 3 Channel Selective Search (CS^2)

Task: Channel Identification.
Initiations: $Th_3 \leftarrow$ selection threshold.
Input: C $\leftarrow$ the updated centroids for clustered normal vectors of subspaces.
Output: $\hat{\mathbf{w}} \leftarrow$ the channel candidate.
Step 1: $\forall\ \tilde{\mathbf{C}}_j \in R^{N_x \times N_x}$ from sub-matrices of $\mathbf{C} \in R^{N_x \times N}$ where $j \leftarrow 1\ to\ C^{N_x}_N$.
Step 2: Build $\boldsymbol{\rho}_j = \tilde{\mathbf{C}}_j\tilde{\mathbf{C}}_j{}^T$ if $\frac{|\lambda_1|}{|\lambda_{N_x}|} < Th_3$ and $\frac{|\lambda_2|}{|\lambda_{N_x}|} > 10^8 \times Th_3$ then $\mathbf{v} \leftarrow \mathbf{v}_1$ go to step 3 else go to step 1, where λ_1 and λ_{N_x} are the smallest and largest Eigenvalues of $\boldsymbol{\rho}_j$ respectively and $\mathbf{v}_1$ is the Eigenvector that correspond to λ_1.
Step 3: Calculate $d_1 = max(max(|\mathbf{v}^T\tilde{\mathbf{C}}_j|)$.
Step 4: $\forall\ \hat{\mathbf{C}}_i \in R^{N_x \times N_x - 1}$ from submatrices of $\tilde{\mathbf{C}}_j \in R^{N_x \times N_x}$ where $i = 1\ to\ C^{N_x - 1}_{N_x}$.
Step 5: Build $\hat{\boldsymbol{\rho}}_i$=$\hat{\mathbf{C}}_i\hat{\mathbf{C}}_i{}^T$ and calculate $\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}}_1$ where $\hat{\mathbf{v}}_1$the Eigenvector that correspond to the smallest Eigen-value of is $\hat{\boldsymbol{\rho}}_i$.
Step 6: Calculate $d_{i+1} = max(max(|\hat{\mathbf{v}}^T\hat{\mathbf{C}}_i|)$.
Step 7: If max(d)< Th_3 then $\hat{\mathbf{w}} \leftarrow \mathbf{v}_1$ else go to step 1.

3.4 Online Continuous Selective Channel Identification

In this subsection, we describe our proposed algorithm named online continuous selective channel identification (OCS-CI). By continuous term, we meant developing an algorithm that is capable of performing all k-SCA stages simultaneously. Moreover, our online framework could proceed to identify the columns of $\mathbf{A}$ gradually. The OCS-CI benefits from selective algorithms (S^3 and CS^2) in order to enhance the accuracy of channel estimation. In this regard, we repeatedly choose the $N_x \times N_x$ submatrices from the mixed data, in a random way, and accept the generated subspaces based on criteria introduced in S^3 algorithm.

To further accelerate the convergence of OCS-CI, we cluster the received mixture vectors using an online and unsupervised clustering methods such as fuzzy ART[12, 13]. Moreover, in order to avoid having local search problem we try to choose subspace candidate vectors from more than one clusters, generated by online-clustering process, randomly. In Algorithm 4, these randomly selected vectors are stacked up in a matrix called ($\mathbf{X}_{sel}$).

A summary of our proposed algorithm is shown by Algorithm 4 (as OCS-CI). As a short description, in the first step we suppress all columns that are close to the origin. Then each column is normalized in order to scaling data to a known range. We next select N_x vectors randomly and apply Algorithm 1 (S^3). The output of previous step would be the normal vector candidate for the detected subspace if there exists .Due to our observation that these normal vectors are detected many times, we then apply a strict online clustering layout based on a bidding distance (Algorithm 2). The output would be updated centroids corresponding to the normal vectors. Finally, we perform our channel estimation algorithm (CS^2) using only N_x subspaces instead of all ($C_{N_s-1}^{k-1}$) subspaces. To cluster the channel candidates the BBC procedure is developed similar to the subspaces clustering algorithm.

Algorithm4 Online Continuous Selective Channel Identification (OCS-CI)

Task: Underdetermined Blind Identification under k-SCA assumptions.
Initializations: $Count \leftarrow 0$ to $preassigned$ N_{max}.
Input: $\mathbf{X}_{Re} \leftarrow$ the received mixed data.
Output: The channels (The columns of mixing matrix).
Step 1. While $Count < N_{max}$ perform Step 2 to 10 otherwise go to Step 11.
Step 2. Cluster the $\mathbf{X}_{Re}$ based on fuzzy ART method.
Step 3. Choose one or more groups of clusters randomly called $\mathbf{X}_{sel}$.
Step 4. Remove the columns from the matrix $\mathbf{X}_{sel}$ that are close to origin.
Step 5. Normalize each observed signal i.e. $\mathbf{x}_{sel}(t) = \mathbf{x}_{sel}(t)/\|\mathbf{x}_{sel}(t)\|$.
Step 6. Select N_x vectors of obtained $\mathbf{X}_{sel}$.
Step 7. Apply *Algorithm 1 (S^3)* on $\widetilde{\mathbf{X}}_{sel}$. If any subspace $\mathbf{w}$ is not detected, go to step 2 otherwise go to step 8.
Step 8. Perform bidding *Algorithm 2 (BBC) for* **w.**
Step 9. If at least N_x subspaces are clustered, apply *Algorithm 3 (CS^2).*
Step 10. If any channel candidate $\widehat{\mathbf{w}}$ is found, $Count \leftarrow Count + 1$ and apply *BBC Algorithm.*
Step 11. End while.

---

4. SIMULATION RESULTS

The accuracy of our proposed approach was evaluated by simulated data experiments. We compared our measured channel identification error for different number of sources (N_s) and sensors (N_x) with those of *k*-SVD [14] algorithm. The *k*-SVD algorithm is designed to learn an overcomplete dictionary matrix that contains k signal-atoms from measurements. Then the measurement matrix can be represented sparsely such that each measurement vector can be considered as linear combination of few estimated atoms. Similarly, in our problem the measured vectors are built by linear combination of overcomplete mixing matrix **A**.
Therefore, it could be used for channel identification when the sources are *k*-sparse. We employ biased angle sum (BAS) [5] distance to measure the channel estimation error as follows

$$BAS(\mathbf{A}, \widehat{\mathbf{A}}) = \sum_{b=1}^{N_s} arccos(<\mathbf{a}_b, \widehat{\mathbf{a}}_b>) \quad (5)$$

Where $\mathbf{a}_b$ and $\widehat{\mathbf{a}}_b$ are bth column of the original and estimated mixing matrix, with optimally re-ordered columns, and $<.,.>$ denotes the inner product [15]. In our experiment, we generate N_s sources signal containing 2000 samples such that there are at most $k = N_x - 1$ active sources in each moment. The sources are mixed using a randomly generated mixing matrix $\mathbf{A} \in R^{N_x \times N_s}$ to build the mixtures without adding noise. Both the source and mixing matrices are created randomly (uniform in$[-1.1]$). Moreover, we considered k -SCA assumptions (*A1*, *A2*, and *A3*) in the simulated data generation process.

We set $Th_1 = 10^{-15}$ (Algorithm 1), $Th_2 = 10^{-3}$ (Algorithm 2), $Th_3 = 10^{-14}$ (Algorithm 3), $N_{max} \geq 10^2$ (Algorithm 4). We performed each experiment 20 times with different $N_x$and N_s values and provided the averaged error of estimated mixing matrix using BAS distance.
The Orthogonal Matching Pursuit (OMP) was chosen for sparse source recovery phase of *k*-SVD and the maximum number of dictionary-learning iterations was set to 400. Table 1 and 2 show the results of the proposed method and those of the *k*-SVD algorithm, respectively. Obviously, Table 1 shows very low estimation error of the proposed method compared to those of the rival. It seems that the K-SVD algorithm was not efficient for $k = N_x - 1$. However, our side experiments showed that it was more efficient for the cases with very sparse sources, $k \ll N_x$ which was not matched with our mixing scenario in this paper.

The Matlab codes to obtain the results of the proposed method (Table 1) and *k*-SVD algorithm (Table 2) are available from:

(https://sites.google.com/site/ehsaneqlimi/codes).

Table 1.The error of proposed channel identification using BAS distance

N_x \ N_s	5	6	7	8	9
3	2.10e-08	3.31e-08	4.21e-08	5.26-08	6.32e-08
4	3.08e-08	4.98e-08	5.42e-08	6.39e-07	8.26e-07

Table 2.The error of *k*-SVD channel identification using BAS distance

N_x \ N_s	5	6	7	8	9
3	0.6535	0.68312	0.7632	0.8157	0.9069
4	0.6847	0.7036	0.7712	0.8231	0.9278

5. DISSCUSSION AND CONCLUSIONS

In this work, a novel *k*-SCA based underdetermined blind identification algorithm based on subspace selective search process is proposed. The proposed method differs from the most current subspace-based algorithms in the sense that, in order to reject outliers, it searches the subspaces selectively. In this way, the subspaces are detected with no exact information about the number of them. As a result, it could be employed to solve UBI/UBSS problems when the number of sources is unknown.

Compared to the conventional k-SCA algorithms, our proposed method is able to perform the channel identification in conjunction with subspace searching in an online scheme. In addition, with this scheme, in contrast to the other well-known k-SCA methods, we can estimate the mixing matrix columns individually even when a few of subspaces are emerged in the mixture data. The method is evaluated with introducing mixture vectors, built by mixing the synthetically generated k-sparse sources, for different cases with different number of mixtures and sources. As the preliminary results, the proposed method has shown good performance compared to state of the art k-SVD algorithm. In addition, few of k-SCA methods, such as the methods proposed in [2,5], were implemented to compare with the proposed one. However, their running trends were not straightforward and mostly were trapped in local minimums.

Consequently, they were not always successful to estimate all mixing columns. Therefore, the comprehensive comparison with the other k-SCA algorithms for accuracy, robustness to noise, and the speed of convergence terms was considered for the future publications.

REFERENCES

[1] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal processing,* vol. 24, pp. 1-10, 1991.

[2] P. Georgiev, F. Theis, and A. Cichocki, "Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures," *IEEE Transactions on Neural Networks,* vol. 16, p. 993, 2005.

[3] P. Georgiev, F. Theis, and A. Cichocki, "Blind source separation and sparse component analysis of overcomplete mixtures," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, 2004, pp. V-493-6 vol. 5.

[4] Z. He, A. Cichocki, Y. Li, S. Xie, and S. Sanei, "K-hyperline clustering learning for sparse component analysis," *Signal Processing,* vol. 89, pp. 1011-1022, 6// 2009.

[5] Z. He and A. Cichocki, "K-subspace clustering and its application in sparse component analysis," in *Proc. ESANN 2006*, 2006.

[6] W. Yang and H. Zhang, "Blind source separation based on K-SCA assumption," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 2010, pp. 116-121.

[7] J. Wen, H. Liu, S. Zhang, and M. Xiao, "A new fuzzy K-EVD orthogonal complement space clustering method," *Neural Computing and Applications,* vol. 24, pp. 147-154, 2014.

[8] C.-J. Yao, H.-L. Liu, and Z.-T. Cui, "Mixing matrix recovery of underdetermined source separation based on sparse representation," in *Computational Intelligence and Security, 2007 International Conference on*, 2007, pp. 1-5.

[9] Y. Washizawa and A. Cichocki, "On-Line K-PLANE Clustering Learning Algorithm for Sparse Comopnent Analysis," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2006, pp. V-V.

[10] F. J. Theis, P. Georgiev, and A. Cichocki, "Robust sparse component analysis based on a generalized Hough transform," *EURASIP Journal on Applied Signal Processing,* vol. 2007, pp. 86-86, 2007.

[11] F. J. Theis, P. G. Georgiev, and A. Cichocki, "Robust overcomplete matrix recovery for sparse sources using a generalized Hough transform," in *ESANN*, 2004, pp. 343-348.

[12] C.-T. Lin and C.-F. Juang, "An adaptive neural fuzzy filter and its applications," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 27, pp. 635-656, 1997.

[13] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural networks,* vol. 4, pp. 759-771, 1991.

[14] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation" *IEEE Trans. Signal Process.*, vol. 54, pp. 4311-4322, 2006.

[15] L. De Lathauwer and J. Castaing, "Second-order blind identification of underdetermined mixtures," in *Independent Component Analysis and Blind Signal Separation*, ed: Springer, 2006, pp. 40-47.