ON-LINE K-PLANE CLUSTERING LEARNING ALGORITHM FOR SPARSE COMOPNENT ANALYSIS

Yoshikazu Washizawa and Andrzej Cichocki

Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan.
{washizawa, cia}@brain.riken.jp

ABSTRACT

In this paper we propose a new algorithm for identifying mixing (basis) matrix $\boldsymbol{A}$ knowing only sensor (data) matrix $\boldsymbol{X}$ for linear model $\boldsymbol{X} = \boldsymbol{AS} + \boldsymbol{E}$, under some weak or relaxed conditions, expressed in terms of sparsity of latent (hidden) components represented by the matrix $\boldsymbol{S}$. We present a simple and efficient on-line algorithm for such identification and illustrate its performance by estimation of unknown matrix $\boldsymbol{A}$ and source signals $\boldsymbol{S}$. The main feature of the proposed algorithm is its adaptivity to changing environment and robustness in respect to noise and outliers that do not satisfy sparseness conditions.

1. INTRODUCTION

Sparse Component Analysis (SCA) and Sparse Signals Representations (SSR) arise in many scientific problems, especially, when one wishes to represent signals of interest by using a small (or sparse) number of basis signals from a much larger set of signals. Such problems arise also in many applications including electro-magnetic and biomagnetic inverse problems (EEG/MEG), feature extraction, filtering, wavelet denoising, time-frequency representation, neural and speech coding, spectral estimation, direction of arrival estimation, failure diagnosis and speed-up processing.

We consider the following linear model :

$$\boldsymbol{X} = \boldsymbol{AS} + \boldsymbol{E}, \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{m\times T}$ is a given data (observation) matrix , $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is unknown full column rank mixing (basis) matrix (not necessary sparse) and and $\boldsymbol{S} \in \mathbb{R}^{m\times T}$ is also unknown matrix representing sparse sources or hidden components, T is the number of available samples, m is the number of observations and n is the number of sources. The objective is to estimate both the mixing matrix $\boldsymbol{A}$ and sources $\boldsymbol{S}$. Most of the existing algorithms for SCA assume that source signals are very sparse in the sense that for most samples in each time instant only one source signal is active. Very recently, these conditions have been considerably relaxed, but developed algorithms are quite sophisticated and complex and not easy to implement for on-line, close to real time applications [1], [2], [3]. Most of these algorithms exploit data structure or very specific clustering of observed data $\boldsymbol{X}$ for sparse representation. The most challenging problem in the SCA is precise estimation of mixing matrix. In this paper, we present a new adaptive on-line algorithm for such clustering, which is robust to outliers. Computer simulation examples are presented showing the validity and robustness of our algorithm. We use two stage procedures. In first step we estimate the mixing matrix then in second stage we estimate source signals using pseudoinverse for $m > n$, and FOCUSS algorithm or the algorithm in [4] for $m < n$.

2. IDENTIFICATION OF MIXING MATRIX

Estimation of source signals by SCA requires two steps. The first step is the estimation of mixing matrix $\boldsymbol{A}$, and the second step is recovering original source signals.

If a column of the matrix $\boldsymbol{S}$ has more than $(n - m + 1)$ zero components in most of instants, observed vector lies on a hyperplane generated from columns of $\boldsymbol{A}$. Let us consider at first, an easiest case $n = m = 2$, and assume that source signals are sufficiently sparse, then many columns of observed signals focus on two 1-dimensional subspaces spanned by each column of $\boldsymbol{A}$. In the case of $n, m > 2$, many columns of observed signals focus on $\begin{pmatrix} n \\ m-1 \end{pmatrix}$ $(m - 1)$-dimensional subspaces (hyperplanes) and 1-dimensional subspaces which are generated by intersections of the subspaces correspond columns of $\boldsymbol{A}$. Some methods to obtain hyperplanes are proposed [4], [5]. However they are not on-line algorithms. The main objective of this paper is to propose adaptive on-line algorithm.

The clustering algorithm to obtain hyperplanes is often called k-plane clustering which clusters vectors to k hyperplanes [6]. Let $\mathbf{x}_i$ and $\boldsymbol{w}_j$ be i-th normalized column of $\boldsymbol{X}$ and normal vector of jth hyperplane respectively. To fit a hyperplane to samples, the hyperplane is modified as follows.

$$\begin{aligned} \boldsymbol{w}_j^{(k+1)} &= [I - \eta \mathbf{x}_i \mathbf{x}_i^\top] \boldsymbol{w}_j^{(k)} \qquad (2) \\ &= \boldsymbol{w}_j^{(k)} - \eta \langle \boldsymbol{w}_j^{(k)}, \mathbf{x}_i \rangle \mathbf{x}_i, \qquad (3) \end{aligned}$$

where I is the identity matrix and $0 < \eta \leq 1$ is a learning note which controls a strength of learning. Symbols $\langle \cdot, \cdot \rangle$ and $^\top$ denote an inner product and transpose respectively. If $\eta = 0$, $\boldsymbol{w}_j$ is not modified. If $\eta = 1$, $\boldsymbol{w}_j$ is projected to the orthogonal complement of the space spanned by $\mathbf{x}_i$. Then the $\boldsymbol{w}_j$ and $\mathbf{x}_i$ are perpendicular. In other words, $\mathbf{x}_i$ lies on j-th hyperplane. If $0 < \eta < 1$, $\boldsymbol{w}_j$ is modified to be close to $\mathbf{x}_j$. For the larger η, learning speed is higher. η can change with respect to learning time. In this case, it decreases monotonically with respect to learning time. Fig. 1 shows a scheme of this algorithm.

However there are $\begin{pmatrix} n \\ m-1 \end{pmatrix}$ hyperplanes lain $\mathbf{x}_i$ and some samples may not have zero component in an instant. Thus we introduce the threshold $\theta(t)$ that decide learning area. If $|\langle \boldsymbol{w}_j, \mathbf{x}_i \rangle| \leq \theta(t)$, j-th hyperplane is learned. We found that the best results are obtained if $\theta(t)$ also decreases monotonically with respect to learning time.

Here we summarize our new learning algorithm.

Matrix identification algorithm

```
1. Remove columns from the matrix $X$ that are close to origin. They are generated from source signal whose components are all zero.

2. Normalize and whiten the matrix $X$. Whitening is not always necessary, but it improves condition number of $A$ if the problem is ill-conditioned and makes estimation be easy.

3. for $j \leftarrow 1$ to $\begin{pmatrix} n \\ m-1 \end{pmatrix}$
    (a) Initialize normal vector $w_j$ randomly and normalize it to unit vector.
        for $t = 1, 2, \ldots,$ learning time
            i. for $i = 1, 2, \ldots, T$
                • If $|\langle w_j, x_i \rangle| \leq \theta(t)$, update $w_j$
                  $w_j \leftarrow w_j - \eta(t) \langle w_j, x_i \rangle x_i$,
                  $w_j \leftarrow w_j / ||w_j||$
    (b) Remove samples such that $d_i \leq \theta(t)$

4. Obtain normal vectors of hyperplanes spanned by every $(m-1)$ vectors which are subset of a set of $w_j$.

5. Cluster normal vectors to $n$ cluster, then they are row vectors of $A$ up to permutation and scaling.
```

After all $\boldsymbol{w}_j$ are obtained, it is not necessary to remove samples in step 3-(b). For sequential $\mathbf{x}_i$, hyperplanes such that $\langle \mathbf{x}_i, \boldsymbol{w}_i \rangle \leq \theta(t)$ should be learned.

3. RECOVERING ORIGINAL SOURCE MATRIX

In the case of $n \leq m$, estimated original source signals $\hat{\boldsymbol{S}}$ are obtained by pseudo inverse matrix of estimated mixing matrix $\hat{\boldsymbol{A}}$,

$$\hat{\boldsymbol{S}} = \hat{\boldsymbol{A}}^\dagger \boldsymbol{X} = (\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^\top \boldsymbol{X}. \qquad (4)$$

If $m < n$, recovering original source signals is more difficult. Some methods to recover source signals are proposed. We use the method proposed in [4]. This method can ve used in on-line learning.

Theorem 1 (Source recovery [4]). *Let $\mathcal{H}$ be the set of all $\mathbf{x} \in \mathbb{R}^m$ such that the linear system $\boldsymbol{A}\boldsymbol{s} = \mathbf{x}$ has a solution with at least $n - m + k$ components. If $\boldsymbol{A}$ fulfills A1), then there exists a subset $\mathcal{H}_0 \supset \mathcal{H}$ with measure zero with respect to $\mathcal{H}$, such that for every $\mathbf{x} \in \mathcal{H} \backslash \mathcal{H}_0$ this system has no other solution with this property.*

From Theorem 1 it follows that the sources are identifiable generically (or with probability one), i.e. up to a set with measure zero, if they have level of sparseness grater than or equal to $n - m + 1$, and the mixing matrix is known. Below is the algorithm, based on the observation in Theorem 1.

Source recovery algorithm

```
1. Identify the set of $k$-conditional subvectorspaces $\mathcal{H}$ produced by taking the linear hull of every subsets of the columns of $A$ with $m - k$ elements.

2. for $i \leftarrow 1$ to $N$
    (a) Identify the space $H \in \mathcal{H}$ containing $x_i$, or in practical situation with presence of noise, identify the one to which the distance from $x_i$ is minimal and project $x_i$ onto $H$ to $\tilde{x}_i$.
    (b) If $H$ is spanned by columns $a_{i_1}, \ldots, a_{i_{m-k}}$ of $A$, then find coefficients $\lambda_{i,j}$ such that $\tilde{x}_i = \sum_{j=1}^{m-k} \lambda_{i,j} a_{i_j}$. These coefficients are uniquely determined if $\tilde{x}_i$ does not belong to the set $\mathcal{H}_0$ with measure zero with respect to $\mathcal{H}$.
    (c) Construct solution $s_i$; $i$th column of $S$: it contains $\lambda_{i,j}$ in the place $i_j$ for $j = 1, \ldots, m - k$, the other component are zero.
```

4. COMPUTER SIMULATION

To validate our algorithm, we show two experimental results. One is a complete ($n = m$) and nonstationary matrix $\boldsymbol{A}$ case and the other is an overcomplete ($m < n$) and stationary case, i.e., for fixed matrix $\boldsymbol{A}$.

4.1. Complete ($n = m = 5$) and nonstationary case

In this experiment, we show the ability of pursuit with variations of $\boldsymbol{A}$. We made five artificial source signals using *randn* command of Matlab, which provides standard normal distribution. Then one element chosen randomly substitute zero for each column. $\boldsymbol{A}$ is also decided randomly as follows,

$$\boldsymbol{A}(t) = \begin{bmatrix} 0.5130 & 0.5841 & 0.8188 & 0.5449 & 0.3984 \\ 0.1718 & 0.5608 & 0.1926 & 0.5200 & 0.4375 \\ 0.5325 & 0.4870 & 0.4443 & 0.2074 & 0.4702 \\ 0.6507 & 0.3224 & 0.0129 & 0.5482 & 0.4380 \\ 0.0164 & 0.0577 & 0.3080 & 0.2986 & 0.4867 \end{bmatrix} \quad (\text{for } \ t \leq 500).$$

Each column of $\boldsymbol{A}$ is normalized to unit length vector. After 500 samples, we modified $\boldsymbol{A}_{11}$ as $\boldsymbol{A}_{11} + 0.1\lceil t/1000 \rceil$, where t is sample number after 500 samples and $\lceil \cdot \rceil$ denotes a ceil function. Then we normalized each column of $\boldsymbol{A}$.

At first, we estimated a mixing matrix $\hat{\boldsymbol{A}}$ using first 500 samples. Then the error between $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ which is measured by $\min_{P \in \mathcal{P}} \|\hat{\boldsymbol{A}}P - \boldsymbol{A}\|_2$ is 3.13×10^{-3}, where $\mathcal{P}$ is a set of permutation matrices, and $\| \cdot \|_2$ is Frobenius norm. We set maximum number of iterations to 2,000 and while learning, learning coefficient $\eta(t)$ and $\theta(t)$ are set as follows,

$$\begin{aligned} \eta(t) &= 0.5 - (2.5 \times 10^{-4})t \\ \theta(t) &= \cos(\pi/4 + (4 \times 10^{-4})t). \end{aligned}$$

Fig. 2 shows the error after 500 samples. The dotted line shows the error between an initial mixing matrix $\boldsymbol{A}_{\text{ini}}$ and modified $\boldsymbol{A}$.

4.2. Overcomplete and stationary case

In order to show the performance of this algorithm in the overcompleate case, we experimented under the condition $n = 5$, $m = 3$. $\boldsymbol{A}$ is decided randomly as follows,

$$\boldsymbol{A} = \begin{bmatrix} 0.5525 & 0.3919 & 0.5707 & 0.3934 & 0.6904 \\ 0.6863 & -0.6066 & 0.5166 & 0.8634 & -0.6007 \\ 0.4730 & 0.6917 & -0.6383 & -0.3158 & 0.4032 \end{bmatrix}. \quad (5)$$

We generate 2,000 samples as artificial source signals and substitute zero to 3 components chosen randomly. Fig. 3 shows the source signals we used. We set learntimes to 2,000 and $\eta(t)$ and $\theta(t)$ are set as follows,

$$\begin{aligned} \eta(t) &= 0.1 - (5 \times 10^{-5})t \\ \theta(t) &= \cos(\pi/4 + (4 \times 10^{-4})t). \end{aligned}$$

Using our method, we obtained estimated mixing matrix as follows,

$$\hat{\boldsymbol{A}} = \begin{bmatrix} 0.3865 & 0.3935 & 0.5525 & 0.3981 & 0.6907 \\ -0.6082 & 0.8634 & 0.6861 & 0.5431 & -0.6004 \\ 0.6933 & -0.3158 & 0.4733 & -0.7393 & 0.4031 \end{bmatrix}. \quad (6)$$

In this case, all columns of the matrix $\boldsymbol{A}$ are recovered almost perfectly, but only 3rd column is estimated with larger error. Then the error $\min_{P \in \mathcal{P}} \|\hat{\boldsymbol{A}}P - \boldsymbol{A}\|_2$ equals to 0.2018. The recovered signals are shown in Fig. 4.

5. DISCUSSION

Many existing algorithms for SCA require some strong conditions to estimate mixing matrix. Especially, they require the number of zeros in each time instant is sufficiently large, because they cluster observed signals before finding hyperplanes. For example, in [4], it is assumed that each column of $\boldsymbol{S}$ has $n - m + 1$ zero components, or it is assumed that each column of $\boldsymbol{S}$ has no more than $m/2$ nonzero components. However it is not guaranteed that environmental signals satisfy such conditions.

Our algorithm allows us to identify a mixing matrix when these conditions are considerably relaxed introducing the learning threshold $\theta(t)$ instead of such assumption. Moreover our algorithm can ignore samples which have many nonzero components if the number of them is relative small in comparison to "sparse" samples.

The performance of matrix identification is related to many factors. The condition number of mixing matrix, the number of samples which have sufficient zero elements, additive noise and the number of observations are all related with its performance. If the mixing matrix is ill-posed, i.e., its condition number is large, hyperplanes are close each other and it becomes very difficult to detect all hyperplane. If n is large and m is small, the number of hyperplanes, $\begin{pmatrix} n \\ m-1 \end{pmatrix}$ is very large and the number of their intersections increases exponentially. Therefore, many samples are needed to obtain all hyperplane. We also experimented such cases, however our method is valid at least for $n < 10$.

From an experiment in section 4.1, our algorithm can pursue slow variations of mixing matrix. The ability of pursuit is related with variations and learning threshold $\theta(t)$. For large variation, learning threshold should be large, however in the case that the noise is large or mixing matrix has large condition number, pursuit will be difficult.

From an experiment in section 4.2, our algorithm is still useful for overcompleate case. However, as mentioned above, if n increase, the number of hyperplane and their intersection increase exponentially and for large or even medium scale problem SCA is still a big challenge.

6. CONCLUSION

We developed new algorithm for SCA. It is simple and on-line learning method. The experimental results show its ability of pursuit and performance under the overcompleate condition.

Optimal parameters or stability of our methods is not still discussed. It has to be considered in the future.

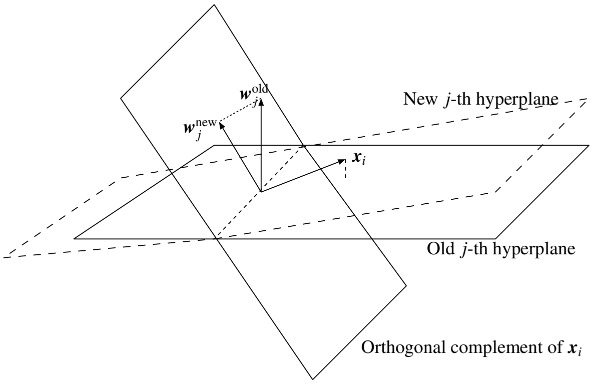

Fig. 1. Matrix identification algorithm

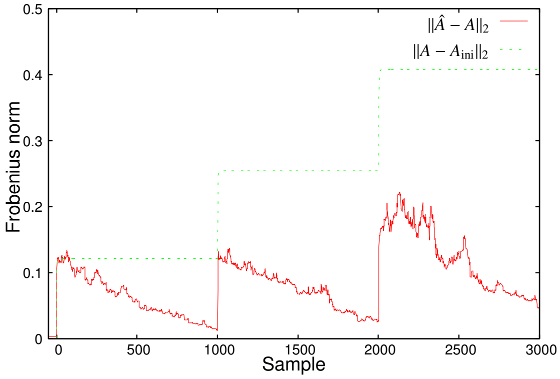

Fig. 2. Error between $\hat{A}$ and A

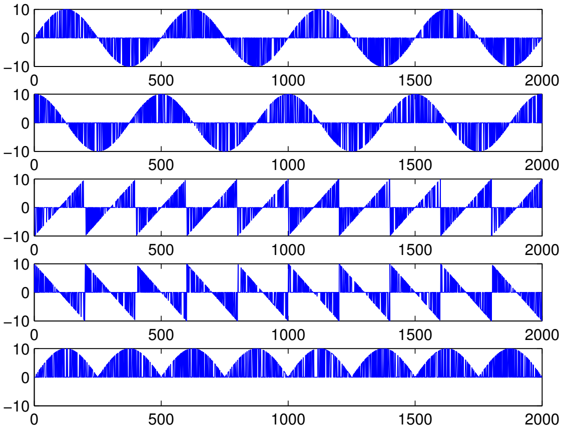

Fig. 3. Source signals (section 4.2)

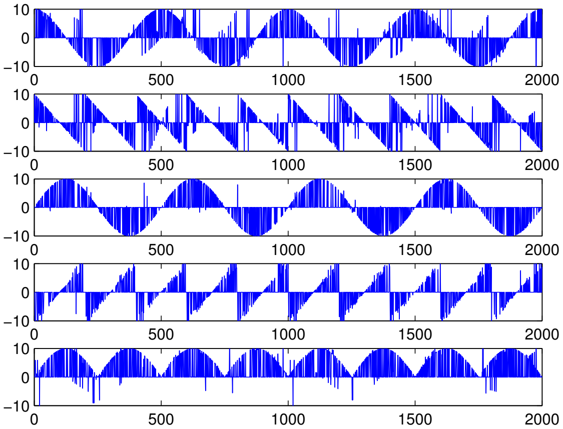

Fig. 4. Estimated Source signals from 3 mixed signals (section 4.2)

7. REFERENCES

[1] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *to appear in IEEE transactions On Signal Processing*.

[2] J. A. Tropp, "Greed is good : Algorithmic results for sparse approximation," *IEEE transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, October 2004.

[3] R. Gribonval and M. Nielsen, "Sparse decompositions in unions of bases," *IEEE transactions on Information theory*, vol. 49, no. 12, pp. 3320–3325, December 2003.

[4] P. G. Georgiev, F. J. Theis, and A. Cichocki, "Blind source separation and sparse component analysis of overcomplete mixtures," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2004)*, Montreal, Canada, May 2004, IEEE Signal Processing Society, vol. V, pp. 493–496, IEEE.

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

[6] P. S. Bradley and O. L. Mangasarian, "k-plane clustering," *Journal of Grobal Optimization*, vol. 16, no. 1, pp. 23–32, January 2000.