A Fast Algorithm for Undetermined Mixing Matrix Identification Based on Mixture of Guassian (MoG) Sources Model

Jiechang Wen
1 School of Applied Mathematics, Guangdong University of Technology, Guangzhou, China
Email: wjcpig@126.com

Suxian Zhang 2, Junjie Yang
2 School of Information Engineering, Guangdong University of Technology, Guangzhou, China
Email: zsxchn@yahoo.cn

3 School of Automation, Guangdong University of Technology, Guangzhou, China
Email: yangjunjet0807@163.com

Abstract—This paper proposes a new fast method for identifying the mixing matrix based on a binary state mixture of Gaussian (MoG) source model. First, a necessary discussion for solving the mixing matrix detection is offered under the multiple dominant circumstance. Second, a density detection method is presented to improve the identification performance. Simulations are given to demonstrate the effectiveness of our proposed approach.

Index Terms—blind source separation (BSS); sparse component analysis (SCA); density detection; mixture of Gaussian (MoG) model.

I. INTRODUCTION

As a result of the widely application in the area of speech recognition, wireless communication and biological medical signal processing, Blind Source Separation (BSS) [1-6] is becoming one of the hottest spots in the signal processing field. The linear model of BSS can be stated as follows:

\[ x(t) = As(t) + e(t) \] (1)

where \( t = 1, 2, \ldots T \) and \( A = [a_1 \ldots a_n] \in R^{m \times n} \) is the mixing matrix. The sample of sources \( s(t) \in R^{n \times 1} \), the observed signals sample \( x(t) \in R^{m \times 1} \) and the white Gaussian noise sample \( e(t) \in R^{m \times 1} \). If \( m < n \), which means that the number of sources is greater than the number of observed signals, then the separation problem is degenerated to the Underdetermined Blind Source Separation (UBSS) [7] problem. In this case, traditional independent component analysis (ICA) cannot be directly applied again. However, since signals are sparse in the real environment or in the frequency domain through Fourier or Wavelet transform, therefore we can solve the UBSS problem using sparse component analysis (SCA) [8]. According to the sparsity assumption of sources, model (1) can rewrite as follows:

\[ x(t) = \sum_{j=1}^{k} \alpha_{i_j}(t) s_{i_j}(t) + e(t), i_j(t) \in \{1, 2, \ldots n\} \] (2)

where \( t = 1, 2, \ldots T \) and \( k \) is the number of active source components at each instant.

Typically, a two-stage "clustering-then-\( l_1 \)-optimization" approach is often used in SCA, which is included by the mixing matrix estimation stage and sources recovery stage. According to the different of active components number of sources, the problem of mixing matrix estimation can be categorized into two types: \( k = 1 \) and \( k > 1 \). To the one single dominant SCA problem [7] \((k = 1)\), several linear orientation-based algorithms [7-9] are addressed to solve this single dominant SCA problem; To the second case \( k > 1 \), this problem, which is called as multiple dominant SCA problem [10-12], can be solved by two steps: concentration hyperplanes clustering and mixing vectors identification. Although these hyperplane methods can effectively improve the precision of mixing matrix identification, they may not be applied in practice because of high computational costs. For overcoming this problem, we propose a novel method to reduce the time cost in the identification of \( A \). First, we discuss the geometrical distribution feature of the observed sample. Second, we give a simple density detection method to reduce the complexity of algorithm by avoiding traditional concentration hyperplane clustering step.

This paper is organized as follows: we make an intuitive and heuristic analysis of new algorithms in section 2. The binary state mixture of Gaussians (MoG) mode [13] is introduced in section 3. The complete algorithm is given in section 4. Section 5 provides some numerical simulations to demonstrate the effectiveness of our new algorithm. Then we discuss and conclude in section 6.

II. THEORETICAL ANALYSIS OF THE ALGORITHM

A. Evolutionary Algorithm

From the k-sparse mixture model of (2) , there are \( c(e = C^k_n) \) k-dimensional hyperplanes to the observed data samples. And each column of lies in the hyperplanes
of \( q(q = C^k_{n-1}) \) hyperplanes. By the normalization process for each \( x(t) \), data samples are projected onto a unit \( n \)-dimensional sphere (as is shown in Fig.1). In other words, the directions of vectors in mixing matrix \( A \) can be detected by the directions of data sample intersections. Therefore, we can estimate the columns of mixing matrix \( A \) by detecting these intersections instead of hyperplane clustering.

Since the amplitude of sources is limited, then each hyperplane is bounded in a fixed region in which the area is considered as \( s \). Suppose the probability of data points located in a hyperplane is \( f \) and one hyperplane can be equally devided into several hyperplanes; Without loss of generality, the probability of data points locating in a hyperplane, which is denoted as \( \varphi \), can be calculated as \( \frac{1}{f} \int f_s dx_1 dx_2...dx_m \). For simplicity, we assume that each hyperplane has the same area. If the number of hyperplanes is \( l \) and the number hyperplane
denotations are denoted as \( N_1,...,N_c \), the probability of points in the same hyperplane (which is denoted as \( \phi \)) is approximately as \( f \times \frac{1}{s^2} \). As is stated in section 2.1, there are \( n \) intersections on the hypersphere. The probability of the hyperplane containing intersection point is denoted as \( p_i(i \in \{1...n\}) \), which is intersected by \( q \) hyperplanes \( B_{i1},...,B_{iq} \). Therefore, the number of points in this hyperplane can be calculated by the following equation:

\[
N_{p_i} = (NB_{i1} \times f_{B_{i1}}(P_i) + ...NB_{iq} \times f_{B_{iq}}(P_i)) \times s/l \quad (3)
\]

We assume that all the hyperplanes have the same number of points, and this total number equals to \( N \). Then (3) can be changed as

\[
N_{p_i} = (f_{B_{i1}}(P_i) + ...f_{B_{iq}}(P_i)) \times N \times s/l \quad (4)
\]

But the other hyperplanes which do not contain intersection points would contain less than \( N \times f_{B_{i1}}(\bar{P}_i) \times s/l \) numbers of point, \( i \in \{1...c\} \), where \( f_{B_{i1}}(\bar{P}_i) \) refers to probability density functions of other points in an arbitrary hyperplane \( B_{i} \), \( i \in \{1...c\} \). In other words, the number of points in the two kinds of hyperplane (one contains the intersection point and the other does not) is greatly different. So the ratio between them is given as:

\[
\frac{N_{p_i}}{N_{\bar{p}_i}} = \frac{f_{B_{i1}}(P_i) + ...f_{B_{iq}}(P_i)}{f_{B_{i1}}(\bar{P}_i)} \quad (5)
\]

Note that if the distribution of points in every hyperplane is previously known, there may be some methods to distinguish the difference between intersection regions and other regions. For example, consider points in all hyperplanes are identically distributed with a uniform distribution. Then the ratio value of these two kind regions is

\[
\frac{N_{p_i}}{N_{\bar{p}_i}} = \frac{f_{B_{i1}}(P_i) + ...f_{B_{iq}}(P_i)}{f_{B_{i1}}(\bar{P}_i)} = q \quad (6)
\]

As is shown in (6), the number of data points in the intersection regions is \( q \) times larger than other regions which do not contain intersection point intersection detect the intedens points from the density regions.

III. SYSTEM MODEL AND THE DISTRIBUTION FEATURE OF DATA SAMPLES

From the analysis above, we found that the distribution of observed data points in \( n \)-dimensional space is decided by the distribution of observed signal points in one hyperplane. In this section, our major job is to study the distribution of observed signals in a hyperplane.

A. The mathematical model of one hyperplane

Still review the model of (2), suppose that there are \( N \) obshyperplanes in the same hyperplane. Then the system model is:

\[
x_t = \sum_{j}^{k} \alpha_{ij} s_j(t_i) + e(t_i), l \in \{1...N\}, i_j \in \{1,2,...n\} \quad (7)
\]

As is stated in (7), it is very important to study the source model. In order to depict the distribution of source, we will introduce the following source models.

B. Binary state MoG source model

The mixture of Gaussian model [13] is one of the non-Gaussian signal model. A \( p \)-th order of MoG is given as:

\[
p(s_i) = \sum_{k=1}^{p} \pi_{i,k} N_{s_i}(0, \delta^2_{i,k}) \quad (8)
\]

where \( \sum_{k=1}^{p} \pi_{i,k} = 1 \). The MoG model is often used for depicting non-Gaussian signals like speech/audio signals. To the binary state of MoG model, which is the simplest MoG model, is widely applied in image processing and modeling sparsity in wavelet decomposition [13-16]. This model is provided as follows:

\[
p(s_i) = \pi_{i,1} N_{s_i}(0, \delta^2_{i,1}) + \pi_{i,2} N_{s_i}(0, \delta^2_{i,2}) \quad (9)
\]

where \( \delta_{i,1} \approx 0, \delta_{i,2} \gg \delta_{i,1}, \pi_{i,1} \in [0, 1] \) and \( \pi_{i,1} + \pi_{i,2} = 1 \). In other words, we can rewrite model (9) as follows:

\[
p_{s_i} = \begin{cases} N_{s_i}(0, \delta^2_{i,1}), & s_i \text{ is inactive with probability of } \pi_{i,1} \\ N_{s_i}(0, \delta^2_{i,2}), & s_i \text{ is inactive with probability of } \pi_{i,2} \end{cases} \quad (10)
\]
matrix identification:
of intersected region.

estimate the columns of mixing matrix \( A \) by the detection
intersections is larger than others. As a result, we can
state sparse Gaussian model, the density of points of the
If the source signal satisfies the distribution of binary-

\[
\frac{1}{\eta^2} \sum_{i=1}^{k} \alpha_i \alpha_i^T \delta \eta^2 
\]

and calculate the number of each hyperplane. As is shown
we divide the data sample space into 350 hyperplanes
produce 10000 observed points in a 3-dimensional space.
Assuming that 5 sources are generated by the model of
binary state sparse MoG and the parameter \( \delta_2 = 1 \), then
produce 10000 observed points in a 3-dimensional space.
We divide the data sample space into 350 hyperplanes
and calculate the number of each hyperplane. As is shown
in Fig.2, it is obvious that there are 5 indices are more
significant than others.

With the discussion above, we can make a conclusion.
If the source signal satisfies the distribution of binary-
state sparse Gaussian model, the density of points of the
intersections is larger than others. As a result, we can
estimate the columns of mixing matrix \( A \) by the detection
of intersected region.

C. Distribution of data points in one hyperplane

Suppose the components of source signals are independent
with the same distribution, then the distribution of
observed signals is:

\[
P(x|A, s) = \sum_{i=1}^{k} N_k(0, \alpha_i \alpha_i^T \delta^2) = N_k(0, \sum_{i=1}^{k} \alpha_i \alpha_i^T \delta^2)
\]

Therefore, we can know that the distribution of the
observed signals also satisfies the Gaussian distribution.
The Gaussian signals, the values of probability density
function in the interval \([-\delta, \delta]\) are close to each other.
When the variance is large enough, especially when \( \delta \to \infty \),
the probability of hypersphere points in a hypersphere
can be viewed as a constant in the interval \([-\delta, \delta]\). Therefore,
the value of (5) is close to \( q \) which is large enough to
distinguish intersection regions from others. For example,
assuming that 5 sources are generated by the model of
binary state sparse MoG and the parameter \( \delta_2 = 1 \), then
produce 10000 observed points in a 3-dimensional space.
We divide the data sample space into 350 hyperplanes
and calculate the number of each hyperplane. As is shown
in Fig.2, it is obvious that there are 5 indices are more
significant than others.

With the discussion above, we can make a conclusion.
If the source signal satisfies the distribution of binary-
state sparse Gaussian model, the density of points of the
intersections is larger than others. As a result, we can
estimate the columns of mixing matrix \( A \) by the detection
of intersected region.

IV. COMPLETE MIXING MATRIX
IDENTIFICATION ALGORITHM

Here we summarize the complete algorithm of mixing
matrix identification:

1) Remove the sample that are close to origin.
2) Normalize and symmetric the sample \( x(t) \) by the following process:

\[
x(t) = \begin{cases}
\frac{x(t)}{||x(t)||}, & x(t) > 0 \\
\frac{-x(t)}{||x(t)||}, & x(t) < 0
\end{cases}
\]

3) Divide the \( m \)-dimensional Euclidean space into hyperplanes, where \( l_i = \frac{\max(x_{it}) - \min(x_{it})}{\eta} \), \( \eta \) is an interval length.
4) Assign sample \( x(t) \) into different spaces by the following method. Define a partition matrix \( U \in R^{L \times T} \),
\( u_{ij} \in [0, 1], i \in 0...L, j \in 0...T. \)
For \( j = 1 : T \)
\( \text{For } q = 1 : m \)

\[
\text{Loc} = \ceil{\frac{\sum_{k=1}^{q} x_{j}(t)}{\eta}}
\]

\[
Set = \sum_{i=1}^{m} l_{iq} + \text{Loc}, j
\]

End

End

5) Calculate the number of points in each hyperplane. Choose the first \( n \) largest hyperplanes and estimate the
center of each hyperplane by the following equations:

\[
\tilde{\alpha}_1 = \frac{\sum_{i=1}^{n} x_{1}(i)}{K_1},
\]

\[
\tilde{\alpha}_m = \frac{\sum_{i=1}^{m} x_{m}(i)}{K_m}.
\]

Where \( K_i \) is the number of the data points in the \( i \)-th
data hyperplane.

6) Finally, construct vectors \([\tilde{\alpha}_1,...\tilde{\alpha}_m]\) as the estimated matrix \( \tilde{A} \).

V. SIMULATION EXAMPLES

In all experiences, source samples are generated inde-
pendently and satisfy the distribution of the binary state
MoG model which is also used in paper [13]. All the
simulations were performed in MTALAB7 environment
using Intel Pentium 42.4GHz processor with 512M RAM
under Microsoft Window XP operating system.

A. Experiment 1

Set \( n = 5, m = 4, k = 3 \), the mixing matrix \( A \) are randomly generated and normalized as follows:

\[
A = \begin{pmatrix}
0.7930 & 0.7428 & 0.1410 & 0.9021 & 0.3281 \\
0.1480 & 0.5901 & 0.7010 & -0.3691 & -0.4419 \\
-0.5910 & 0.3161 & -0.6992 & -0.2235 & -0.8349
\end{pmatrix}
\]

The Procedures of our algorithm are shown in Fig.3
and we obtained the estimated mixing matrix as follows:

\[
\tilde{A} = \begin{pmatrix}
0.7890 & 0.9030 & 0.7441 & 0.3244 & 0.1434 \\
0.1546 & -0.3665 & 0.5888 & -0.4454 & 0.7000 \\
-0.5941 & -0.2231 & 0.3152 & -0.8343 & -0.6995
\end{pmatrix}
\]

For demonstrating the validity of our algorithm, the
 criterion which is presented in paper [11] and paper [12]
is used:

\[
\xi = \min_{P \in \Omega} ||A - \tilde{A} P||_2
\]

Where \( \rho \) is the set of all permutation matrices. We calculate
estimation error is 0.0089, and the result is 0.0066
using the algorithm in paper [13] and 0.2018 with paper
[17]. The process took about 160s when the source sample

© 2014 ACADEMY PUBLISHER
number is 1000 and 25s when the source sample number is 400, while the algorithm in this paper only took less than 1s to finish the computing process.

B. Experiment 2

In order to demonstrate our fast detection algorithm is capable of solving large scale problems, we compared our algorithm to the existing method proposed in paper [13] by using two simulations. In the first simulation, the parameters are set as $n=15$, $m=7$, $k=3$, $T=9000$; In the second experiment, parameters are set as $n=30$, $m=15$, $k=2$, $T=8500$. Our method took less than 6 seconds in this two simulation while it took about 40 minutes for the first case and two hours for the second case by using the algorithm in paper [13].

![Figure 5. Regions that contain first and largest points were detected.](image)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>our algorithm</th>
<th>algorithm in paper[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n=15,m=7,k=3,T=9000$</td>
<td>6 Sec</td>
<td>About 40 minutes</td>
</tr>
<tr>
<td>$n=30,m=15,k=2,T=8500$</td>
<td>4 Sec</td>
<td>About 2 hours</td>
</tr>
</tbody>
</table>

Table 1 The performance comparison between our algorithm and the algorithm in paper [13]. To measure the precision of identification, the angle between each estimated vector and its corresponding actual mixing vector (inverse cosine of their dot product) is calculated, the result is shown in Fig.4, the accuracy is close to that presented in paper [17], which shows the proposed method can also estimate the mixing matrix successfully.

As is seen in this experiment, the proposed algorithm can estimate mixing matrix very fast with higher accuracy and do not change much when the parameters get larger. It means that the proposed algorithm can be used for dealing with middle scale problems.

VI. DISCUSSION AND CONCLUSION

In this paper, we propose a fast algorithm to estimate the mixing matrix $A$ in multi-dominant SCA based on a binary state sparse MoG model. There are some aspects we need to discuss to our algorithm.
bases for images. Fortunately, it will be possible to solve all kinds of BSS problem of different kinds of source models if the probability density function of observed data points is given in advance. These issues are currently under study. Meanwhile, the construction of the sparse signal model is still an open problem that remained to be solved.

ACKNOWLEDGMENT

This work was jointly supported by the Natural Science Foundation of Guangdong Province (S2011010005075) and Guangzhou technology projects (11C42110781).

REFERENCES


© 2014 ACADEMY PUBLISHER


Jiechang Wen Female, was born in 1964, Master, Professor. The major research covers Optimization Method and its Application, Intelligent Computation.

Suxian Zhang received his Master degree in Chongqing University of Posts and Telecommunications. He is a M.S.candidate at the school of applied mathematics, Guangdong University of Technology, Guangzhou, China.

Jun-Jie Yang received his Master degree in Applied Mathematics from Guangdong University of Technology, China, in 2011. He is now pursuing his Ph.D degree in Automation of Guangdong University of Technology. His current research interests include Sparse Component Analysis, Physical Layer Security and Smart Grid Security.