

การเรียนรู้บนพื้นฐานหน้าต่างสำหรับการวิเคราะห์องค์ประกอบอิสระแบบโอเวอร์คอมพลีต



นางสาวเบญจมาศ ปัญญางาม

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต


สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

WINDOW-BASED LEARNING FOR OVERCOMPLETE INDEPENDENT COMPONENT ANALYSIS



Miss Benjamas Panyangam

สภามหาวิทยาลัย
จุฬาลงกรณ์มหาวิทยาลัย

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Science
Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic year 2006

Copyright of Chulalongkorn University

4573857123 : MAJOR COMPUTER SCIENCE

KEY WORD: BLIND SOURCE SEPARATION / UNSUPERVISED LEARNING.

BENJAMAS PANYANGAM: WINDOW-BASED LEARNING FOR OVERCOMPLETE INDEPENDENT COMPONENT ANALYSIS. THESIS ADVISOR: PROF. CHIDCHANOK LURSINSAP, Ph.D., THESIS CO-ADVISOR: ASST. PROF. KRISANA CHINNASARN, Ph.D., 66 pp.

The blind source separation (BSS) or independent component analysis (ICA) is a statistical technique for the separation of hidden source signals from observed signals in an unknown mixing system. This dissertation concerns the separation problem, where the number of sources (n) is greater than the number of observed signals (m). This situation is called overcomplete. Many existing algorithms are designed to identify the column vectors of the mixing matrix, basis components, which point toward the directions of independent components. However, most approaches assume that the number of sources mixed in the observed signals is known. This dissertation presents a new method to identify the mixing matrix without prior assumption on the number of sources. The proposed algorithm uses a window search length algorithm to identify the information index for preliminary filtering all relevant points clustered along the basic independent components. Then, the perturbed mean shift algorithm with entropy measure is applied to enhance the actual basic independent components. Finally, source signals are recovered by the minimum ℓ_1 -norm method. From the experimental results on the speech signals from TIMIT database, the proposed algorithm is able to estimate the mixing matrix and the estimated source number is also given. The difference between the actual mixing matrix and the estimated mixing matrix using AMDI value of the proposed algorithm is less than that of standard k-mean method, AICA method, Yuanqing Li et al. 's algorithm, and Qv Lv et al. 's algorithm. Moreover, the algorithm can perform in noisy environments.

Department :**Mathematics**..... Student's Signature : *Benjamas Panyangam*
 Field of Study :**Computer Science**..... Advisor's Signature : *[Signature]*
 Academic Year : ...**2006**..... Co-advisor's Signature : *[Signature]*

Acknowledgements

During my years as a Ph.D. student, I have received a lot of tuition, care and friendship from several people, some of which I wish to thank here.

- First of all I would like to thank National Science and Technology Development Agency (NSTDA) of Thailand who sponsors the research scholarships.
- During my time as a Ph.D.'s student, I am grateful to my advisor, Prof.Chidchanok Lursinsap, to whom with his advice, guidance and care, help me to overcome the necessary difficulties of the process of research and make this dissertation possible.
- I would like to thank my co-advisor, Assist. Prof. Krisana Chinnasarn at Burapha University, who gives me wonderful suggestions in Ph.D. research methodologies.
- I wish to express my special thanks to the dissertation committee for their valuable advice and guidance.
- My thanks go to Ms.Ureerat Wattanachon, Ms.Supaporn Bunrit, Mr.Yuttana Lila, Mr.Maytee Bamrungrajhirun, and all my colleagues at the Advanced Virtual Intelligent Computing (AVIC) Center, Department of Mathematics, Chulalongkorn University, who give me a number of useful suggestions. I also would like to thank my best friends, Ms.Matinee Kiewkanya and Ms.Benchaporn Jantarakongkul, for their supports throughout the research.
- My thanks also go to all my past and present lectures for their valuable lectures and instructions.
- Finally, my deepest gratitude goes to Panyangam's family, for their everlasting love, sponsor, and heart encouragements throughout my life.

Table of Contents

	Page
Thai Abstract	iv
English Abstract	v
Acknowledgements	vi
Table of Contents	vii
List of Tables	ix
List of Figures	xi
 CHAPTER	
1 INTRODUCTION	1
1.1 Introduction and Problem Review	1
1.2 Statement of the Problem	4
1.3 Research Objectives	4
1.4 Scopes of the Study	4
1.5 Research Advantages	5
2 THEORIES AND LITERATURE REVIEWS	6
2.1 Independence of Signals	6
2.2 Overcomplete Independent Component Analysis	7
2.3 The Sparse Representation of the Signals	8
2.4 Estimation of the Independent Components	9
2.5 Estimation of the Mixing Matrix	11
2.6 Mean Shift Procedure	12
2.7 Entropy	14
3 PROPOSED METHOD	17

CHAPTER	Page
3.1 Estimating the Mixing Matrix	18
3.1.1 Inner Point Removal	19
3.1.2 High Density Direction Identification	25
3.1.3 Enhancing Basic Components by Entropy Estimation	26
3.2 An Example of Mixing Matrix Detection	29
4 EXPERIMENTAL RESULTS	31
4.1 Performance Measures	31
4.2 Experimental Design	32
4.2.1 Synthetic Data Sets	33
4.2.2 Real Speech Data Sets	35
4.3 Experiment 1	38
4.4 Experiment 2	44
4.5 Experiment 3	48
4.6 Experiment 4	49
4.7 Experiment 5	49
4.8 Complexity Analysis	53
5 CONCLUSION	58
References	62
Biography	66

List of Tables

Table	Page
4.1 The parameters used in the experiments.	32
4.2 Data sets.	34
4.3 List of algorithms with parameters for comparison of overcomplete ICA algorithms in unknown source number case.	39
4.4 The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , for synthetic data set 1 to data set 4.	41
4.5 The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , for real speech data set 5 to data set 8.	43
4.6 The difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual mixing matrix \mathbf{A} on data set 5 and data set 6, AMDI value.	46
4.7 The difference between the estimated mixing matrix and the actual mixing matrix with AMDI value in time domain and Time Frequency domain, AMDI value.	48
4.8 Result of each test in noisy environment.	55
4.9 The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , using $\tau = 50, 100, \dots, 500$ in time domain.	56
4.10 The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , using $\tau = 50, 100, \dots, 500$ in time frequency domain.	57

Table

Page

5.1 The results of two closest directions of mixing matrix for 3 4 and 5 degrees. 60



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

List of Figures

Figure	Page
1.1	Three types of BSS/ICA problems. 2
2.1	(a) Two-dimensional data distribution. (b)-(c) The scaled basis vectors for the data point \mathbf{x} under a Gaussian and Laplacian prior. (d) The rank-order distribution of 128 entry values (coefficients) of a source vector \mathbf{s} (speech data) under a Gaussian prior (dashed) and a Laplacian prior (solid). 8
3.1	The overview of all our work. 17
3.2	A set of τ connected points around the data point \mathbf{x} and the adjacent points $AP_{\mathbf{x}}$ 20
3.3	An example of the adjacent point being the same direction of \mathbf{x} 21
3.4	Result on the experiment 1 on data set 1. (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process. 23
3.5	A new defined point $\tilde{\mathbf{x}}$ of a point \mathbf{x} with $\mathbf{AP}_{\mathbf{x}} = \mathbf{y}_1, \dots, \mathbf{y}_5$ 23
3.6	An example of two regions which have more overlapped data. 25
3.7	Data set 1 (a) Five simulated source signals (b) Three observed signals. . . 29
4.1	Data set 5 (a) Three speech signals from [7]. (b) Two observed signals. . . . 36
4.2	Data set 6 (a) Six speech signals from [7]. (b) Three observed signals. . . . 36
4.3	Data set 7 (a) Three speech signals from [3]. (b) Two observed signals. . . . 37
4.4	Data set 8 (a) Six speech signals from TIMIT database. (b) Three observed signals. 37

Figure	Page
4.5 Results on data set 5 (top) and data set 6 (bottom). (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process.	45
4.6 Result on data set 7. (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process.	47
4.7 The relationship between the percent of sparse source vectors, the size of parameter τ , and the average value of the information index (I'_{avg})	50
4.8 The average value of information index before and after the inner point removal process.	51
4.9 By using the different choices of τ for synthetic data set 4. (a) The number of estimated sources. (b) The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value	52

CHAPTER I

INTRODUCTION

1.1 Introduction and Problem Review

Independent Component Analysis (ICA) or Blind Source Separation (BSS) is a statistical technique for the separation of n hidden sources from m observed signals (mixtures or sensors) in an unknown mixing system. The original source signals are assumed to be mutually independent, called independent components. Let $\mathbf{s} \in R^n$ represents an n -dimensional vector of unknown sources. Figure 1.1 shows a simple case of BSS problem where these sources are linearly mixed by an unknown mixing matrix $\mathbf{A} \in R^{m \times n}$ in noise-free environment. Given only the mixtures $\mathbf{X} \in R^m$. The aim of BSS is to estimate the mixing matrix, $\hat{\mathbf{A}}$, and then to recover the original sources, $\hat{\mathbf{S}}$. Since the dimensions of source, mixture and estimated source are blind in the separation problem.

This results in three types of BSS problems as follows:

- For $m = n$, the problem is referred to as quadratic, complete, or simply ICA. In this case, when the mixing matrix \mathbf{A} is known, the source signals can be recovered by $\mathbf{S}=\mathbf{A}^{-1}\mathbf{X}$.
- For $m > n$, the problem is said to be undercomplete or overdetermined due to higher dimensionality of the mixture space. This problem is easily reduced to a quadratic BSS by discarding some mixtures or by applying preprocessing methods such as PCA.

- For $m < n$, the problem is called overcomplete or underdetermined due to more sources mixed into less signals.

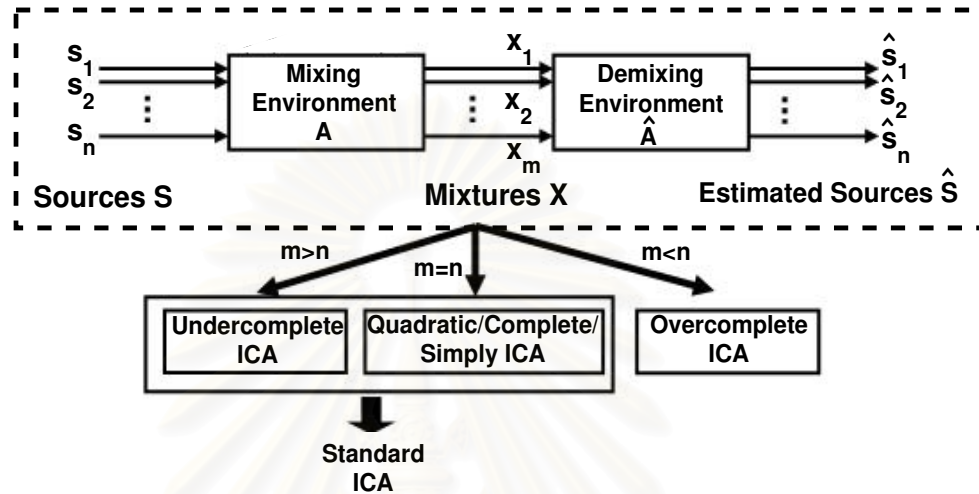


Figure 1.1: Three types of BSS/ICA problems.

Overcomplete ICA is a more powerful method which is able to find the underlying sources from the overcomplete mixtures while the standard ICA techniques fail to model these data [1]. Therefore, overcomplete ICA is an extension of standard ICA which generally assumes that the number of sources is less than or equal to the number of sensors. The main idea in handling the separation problem for overcomplete observed signals is the use of the sparse representation or sparse coding introduced by Olshausen and Field [2]. In this representation, it uses as few basis elements as possible. So this gives an advantage to ICA in case of having more basis vectors than the dimension of mixtures.

Lewicki and Sejnowski [1] firstly applied the sparse representation to overcomplete ICA. Lee et al. [3] proposed an algorithm based on the maximum likelihood (ML) principle for learning the overcomplete representation of two mixtures. Bofill and Zibulevsky [4] induced a two-step approach which separately estimated the mixing matrix and the source signals. In the first step, a potential function based method was used for estimating

the mixing matrix. The estimated sources were, then, obtained by using shortest path algorithm in the second step. This method was very effective for any two-dimensional observable data space. Theis et al.[6] presented a method for reconstructing more sources from fewer mixtures. The algorithm was based on an estimated histogram of the sensor data, and so called histogram-based algorithm. Waheed and Salem [7] used algebraic independent component analysis (AICA) to find the mixing matrix. In the last two methods, the source distributions were assumed to be Laplacian distributions. Thus, the minimum ℓ_1 -norm technique was used to reconstruct the estimated sources. Moreover, the DUET-type methods in [8, 9] and the TIFROM-type methods in [10, 11] were proposed to the separation problem by using the ratio of the observed signals in the time frequency domain. They were based on the connected region of sparse source vectors and consider only two-dimensional spaces of observed signals. In [12], a generalized Hough transform was applied to identify hyperplanes on observed data for estimating the mixing matrix. Then, the sources were recovered using the source recovery algorithm from [4]. The algorithm required to detect the precise intersections of all hyperplanes in which observed data located. However, in the two-step approach, the step for identifying the precise mixing matrix is the most difficult and challenging task [5, 12]. In additions, the case of unknown number of sources was not discussed in those two-step methods.

Recently, only a few of the methods for the underdetermined blind source separation problem were proposed for the unknown source number case. In [13], an extension of the DUET and the TIFROM methods was proposed to estimate the unknown mixing matrix. The algorithm divides data into many sub-matrices and chooses the one with lowest row variance. Several loops were performed to obtain all possible columns of the mixing matrix with small row variances. In [14], an unsupervised robust C prototypes algorithm was used to estimate the mixing matrix and the source number. However, the algorithm was considered by the condition that the source signals must be sufficiently

sparse and the situation was in noise-free environment.

In this dissertation, we study the separation of independent components from their overcomplete mixtures with assumption of sparse signal representations.

1.2 Statement of the Problem

The following issues of overcomplete ICA are investigated.

1. What efficient technique do we use to estimate the mixing matrix for the sparse mixtures?
2. How can we determine the number of sources mixed into the observed signals?

1.3 Research Objectives

In this dissertation, the objectives of this work are as follows:

1. To propose a new algorithm to estimate the mixing matrix for the separation of the overcomplete mixtures.
2. To obtain a technique to estimate the number of sources in unknown environment.

1.4 Scopes of the Study

The scopes in this dissertation are as follows:

1. The observed signals are sparse and their distributions are symmetric.
2. The sources are independently distributed.
3. The number of sources is equal or greater than the number of observed signals ($n > m$) and ($m \geq 2$).

4. The source signals are mixed together by an unbiased mixing matrix in the stationary environment.

1.5 Research Advantages

It is expected that the designed approaches will be:

1. A new unsupervised method is able to separate of overcomplete mixtures.
2. This algorithm can be used for a preprocessing procedure of other signal applications such as signal recognition.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER II

THEORIES AND LITERATURE REVIEWS

In this chapter, the basic concepts of independent component analysis, overcomplete independent component analysis, the sparse representation of signals, estimation of the independent components, estimation of the mixing matrix, mean shift procedure, and the entropy are briefly revised.

2.1 Independence of Signals

The separation of any random signals y_1, y_2, \dots, y_n can be performed with the assumption that the values of $y_i, i = 1 \dots n$, must be statistically independent at all times. The two random variables y_i and y_j are said to be statistically independent if knowing the value of y_i does not give any information on the value of y_j , for $i \neq j$ [15]. Mathematically, the independence of sources can be defined by the probability densities. The random variables y_i and y_j are independent if and only if

$$p(y_i, y_j) = p_i(y_i)p_j(y_j) \quad (2.1)$$

This means that the joint probability density function (pdf) of y_i and y_j , $p(y_i, y_j)$, can be factorized as the product of their marginal probability density functions, $p_i(y_i)$ and $p_j(y_j)$.

2.2 Overcomplete Independent Component Analysis

Given an observed vector $\mathbf{x} = (x_1, x_2, \dots, x_m)$, this variable is generated as a linear mixture of independent components $\mathbf{s} = (s_1, s_2, \dots, s_n)$ by an $m \times n$ unknown mixing matrix \mathbf{A} . Thus, the standard ICA model in a noise free case can be defined as follows:

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^n \mathbf{a}_i s_i \quad (2.2)$$

The above ICA model tries to estimate both the basis vectors, \mathbf{a}_i , and the source vector, \mathbf{s} , from the observed vectors. The standard ICA generally assumes that n is less than or equal to m .

A difficult problem in ICA model appears in the situation that there are more sources than observed signals ($n > m$). This means that the mixing system is not invertible. As a result, we could not directly recover the independent components even if we knew the mixing matrix. This ICA problem is much more complicated and cannot be solved by ordinary ICA methods because the number of “basis vectors”, \mathbf{a}_i , is larger than the dimension of the mixture space of \mathbf{x} . The overcomplete ICA can be divided into two problems: (1) how to estimate the mixing matrix and (2) how to recover the independent components. When the basis is overcomplete, the formulation of the likelihood is difficult. Methods based on Maximum Likelihood (ML) estimation are, therefore, rather computationally inefficient [15]. Most recent works on overcomplete ICA are usually assumed a priori that the input distribution is sparse. (we will discuss this in more detail in the next section). Furthermore, a Laplacian prior is usually used for data distribution. Thus, the problem can be mapped to a standard linear program and the sparse sources can be estimated by using a linear programming (LP) with ℓ_1 -norm [1, 15, ?].

2.3 The Sparse Representation of the Signals

Sparse representation or sparse coding of signals, which can be modeled using matrix factorization, has recently received a great attention. Most recent works on overcomplete ICA are usually assumed a priori that the input distribution is sparse which means that only a few of source data coefficients differs significantly from zero. For a given data point $\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^n \mathbf{a}_i s_i$, if one of the sources, s_i , is significantly far from zero, the remaining ones are likely to be close to zero. The density of data in the mixture space forms a set of cluster along the direction of the basis vector, \mathbf{a}_i . So, we can estimate the mixing matrix by finding the direction of maximum distribution using clustering method.

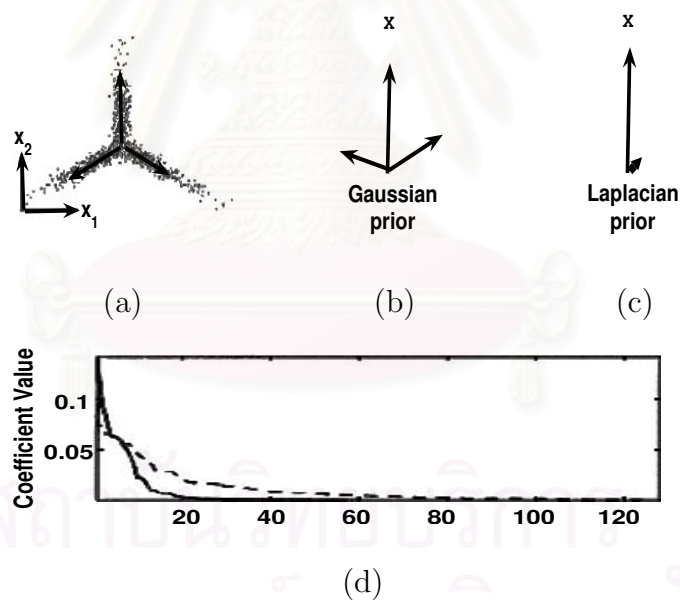


Figure 2.1: (a) Two-dimensional data distribution. (b)-(c) The scaled basis vectors for the data point \mathbf{x} under a Gaussian and Laplacian prior. (d) The rank-order distribution of 128 entry values (coefficients) of a source vector \mathbf{s} (speech data) under a Gaussian prior (dashed) and a Laplacian prior (solid).

The sparse data distributions are often modeled by a priori that has high Kurtosis

such as a Laplacian ($P(s_i) \propto \exp(-\alpha |s_i|)$) [1, 15]. Figure 2.1 (a)-(d) from [1] shows how different source distributions induce different representations of the data. A scatter plot of two-dimensional data with three main arms is shown in Figure 2.1 (a). Figure 2.1 (b) and (c) show two different basis vectors for the data point \mathbf{x} under Gaussian and Laplacian source distributions, respectively. In Figure 2.1 (d), nearly all entries of speech signals are non-zero for Gaussian distribution (dash), while a few of them is non-zero when all source entries have Laplacian distribution (solid).

The great advantage of sparse representation is that since it uses as few “basis” elements as possible, it is capable of reconstructing the original sources even if the number of observed signals is smaller than the number of sources under certain weak conditions. The use of sparse sources provides high quality of the separation results [1, ?]. Overcomplete ICA performance restrictively depends on the sparseness of sources. However, many natural signals are sparse. When the original sources are not sufficiently sparse, a suitable linear transform such as Fourier transform or wavelet transform should be applied prior for improving the sparsity of signals [1, 4, 17, 18, 19]. Typically, the observed data in time-frequency domain are represented by using wavelets package [19]. After performing the separation in the transformed domain, the recovered sources are then transformed back into the original domain.

2.4 Estimation of the Independent Components

Assuming we know the mixing matrix \mathbf{A} , a simple method for reconstructing the independent components can be done by the inverse, $\hat{\mathbf{s}} = \mathbf{A}^{-1}\mathbf{x}$, or pseudoinverse of the mixing matrix, $\hat{\mathbf{s}} = \mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-1}\mathbf{x}$. However, in overcomplete ICA, the mixing matrix is not invertible. The values of the statistically independent components cannot be directly recovered by simply inverting the mixing matrix. A very popular probabilistic approach

applied to estimate the independent components is the Maximum Likelihood approach. It maximizes the posterior probability densities of \mathbf{s} , $P(\mathbf{s}|\mathbf{x},\mathbf{A})$, which is the probability of event \mathbf{s} after knowing \mathbf{x} . For noise-free independent components, $P(\mathbf{s})$ describes the prior probability densities of the independent components, and $P(\mathbf{x}|\mathbf{s},\mathbf{A})$ denotes the probability of observed signal \mathbf{x} , given \mathbf{A} and \mathbf{s} . Thus, we can obtain an estimation of the unknown sources as follows:

$$\begin{aligned}\hat{\mathbf{s}} &= \operatorname{argmax}_{x=As} P(\mathbf{s}|\mathbf{x},\mathbf{A}) \\ &= \operatorname{argmax}_{x=As} P(\mathbf{x}|\mathbf{s},\mathbf{A})P(\mathbf{s})\end{aligned}\quad (2.3)$$

Since \mathbf{x} is fully determined by \mathbf{s} and \mathbf{A} and the sources are mutually independent of each other, the joint probability distribution has the form $P(\mathbf{s}) = \prod_{i=1}^n P(s_i)$. Then, we can get the maximum likelihood estimator of \mathbf{s} as follows:

$$\begin{aligned}\hat{\mathbf{s}} &= \operatorname{argmax}_{x=As} P(\mathbf{s}) \\ &= \operatorname{argmax}_{x=As} \prod_i P(s_i)\end{aligned}\quad (2.4)$$

After taking the logarithm, the product in the equation can be written as the sum of logarithms as follows:

$$\hat{\mathbf{s}} = \operatorname{argmax}_{x=As} \sum_i \log P(s_i)\quad (2.5)$$

If the source distributions were fixed a priori and assumed to be Laplacian distributed, that is $P(s_i) = \exp(-\alpha|s_i|)$, then the maximum log-likelihood estimation is equivalent to the minimum ℓ_1 -norm as follows:

$$\begin{aligned}\hat{\mathbf{s}} &= \operatorname{argmax}_{x=As} \sum_{i=1}^n \log(e^{-\alpha|s_i|}) \\ &= \operatorname{argmax}_{x=As} (-|s_1| - |s_2| - \cdots - |s_n|) \\ &= \operatorname{argmin}_{x=As} |s_1| + |s_2| + \cdots + |s_n| \\ &= \operatorname{argmin}_{x=As} \|\mathbf{s}\|_1\end{aligned}\quad (2.6)$$

where $\|\mathbf{s}\|_1 = \sum_i |s_i|$ defines the ℓ_1 -norm. Using this expression, \mathbf{s} can be uniquely determined [1, 20]. Choosing the priors to be Laplacian the problem can be achieved by using a standard linear program as follows:

$$\min_{\mathbf{x}=\mathbf{A}\mathbf{s}} \sum_i |s_i| = \min[1, 1]^T |\mathbf{s}| \quad \text{subject to } \mathbf{A}\mathbf{s} = \mathbf{x} \quad (2.7)$$

Recently, Li, Cichocki and Amari [18] discussed that it is possible to correctly estimate the source vector if the number of nonzero entries in the source vector is less than $\frac{(m+1)}{2}$. Furthermore, in their later work [19], they showed that if the number of non-zero entries is less m then the minimum ℓ_1 -norm approach can give the unique solution with a probability of one.

2.5 Estimation of the Mixing Matrix

For the Maximum Likelihood approach, the objective for learning the basis vectors of the mixing matrix \mathbf{A} is to maximize the probability of the data, $P(\mathbf{x}|\mathbf{A}) = \prod_{i=1}^N P(\mathbf{x}_i|\mathbf{A})$. The marginal probability over the unknown source values is defined as $P(\mathbf{x}|\mathbf{A}) = \int_{-\infty}^{\infty} P(\mathbf{x}|\mathbf{A}, \mathbf{s})P(\mathbf{s})d\mathbf{s}$. Thus, the basis vectors were learned by performing gradient ascent on the log of this probability using the approximation of the integral. The learning rule can be obtained as follows:

$$\Delta \mathbf{A} \propto \mathbf{A} \mathbf{A}^T \frac{\partial \log P(\mathbf{x}|\mathbf{A})}{\partial \mathbf{A}} \approx -\mathbf{A}(\Phi(\hat{\mathbf{s}})\hat{\mathbf{s}}^T + \mathbf{I}) \quad (2.8)$$

where $\Phi(\hat{s}_i) = \partial \log P(\hat{s}_i) / \partial \hat{s}_i$ is called the score function and \mathbf{I} is the identity matrix. Since \mathbf{A} is not restricted to be square, this method can work on overcomplete ICA but each gradient step requires the computation of $\hat{\mathbf{s}}$ beforehand by using the minimizing ℓ_1 -norm. The derivation of this learning rule was described in [1].

Recently, some methods based on the two-step approach have been proposed for estimation of the mixing matrix for overcomplete case [4, 17, 18, 20, 6, 7]. Unlike a gradient

type algorithm, in these methods, the estimation of mixing matrix is performed separately from the source recovery step. Some algorithms use the geometric separation with the assumption that sources are zero mean. So, the samples of sources will be transformed by the mixing matrix \mathbf{A} into data clusters along transformed coordinate axes through the origin [21]. Geometric algorithms have recently received some attention because of their ease of implementation. However, the algorithms using two-step approach require the sparse representation of data. As described in sparse representation section, if the signal representation is sparse enough, the density of data in the mixture space shows a clear tendency to cluster along the directions of basis vectors of the mixing matrix. Thus, the estimation of the mixing matrix can be performed by finding this direction using clustering method.

2.6 Mean Shift Procedure

Several non-parametric methods are available for probability density estimation such as the histogram method, the nearest neighbor method, and kernel estimation. The kernel estimation method is one of the most popular techniques used for estimating the density. The mean shift is an unsupervised and non-parametric estimator of density gradient. The mean shift method has been introduced by Fukunaga and Hosteltler in [23]. Many interesting and useful properties of the generalized mean shift algorithm were discussed in [24]. A probabilistic mean shift type algorithm is mentioned in [25]. Comaniciu and Meer also proved that the mean shift procedure applied to discrete data was guaranteed to converge [24, 25]. With the efficient and easy implementation, the mean shift method was applied to many fields such as image filtering [25], clustering [24, 25, 26], image segmentation [27], and microarray analysis [28].

Given N data points \mathbf{x}_i , $1 \leq i \leq N$ in a d -dimensional space, the multivariate mean

shift vector computed with kernel K in any point \mathbf{x} is given by [25].

$$m_h(\mathbf{x}) = \frac{1}{n_x} \sum_{\mathbf{x}_i \in S_h(\mathbf{x})} (\mathbf{x}_i - \mathbf{x}) \quad (2.9)$$

where the region $S_h(\mathbf{x})$ is a hypersphere of radius h containing n_x data points and the kernel is defined as follows:

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1 - \mathbf{x}^T\mathbf{x}) & \text{if } \mathbf{x}^T\mathbf{x} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

where c_d is the volume of the unit- d -dimensional sphere. It can be shown that the mean shift vector $m_h(\mathbf{x})$ at location \mathbf{x} is proportional to the normalized density gradient estimate computed with kernel K .

$$m_h(\mathbf{x}) = \frac{h^2}{d+2} \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_K(\mathbf{x})} \quad (2.11)$$

where $\hat{f}(\mathbf{x})$ is the density estimate of a point \mathbf{x} with a window radius h and kernel K defined as follows:

$$\hat{f}_K(\mathbf{x}) = \frac{1}{Nh^d} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (2.12)$$

The normalization is computed by the density estimate in \mathbf{x} obtained with kernel K . The relation captured in Eq. (2.11) is intuitive. The mean shift vector is aligned with the gradient estimate of the density and the window of computation is always moved toward regions of high density.

The Mean Shift algorithm can be described as follows:

1. Choose the radius h of the search window.
2. Initialize the location of the window.
3. Compute the mean shift vector $m_h(\mathbf{x})$.
4. Translate the search window by $m_h(\mathbf{x})$.
5. Repeat step 3 and step 4 until convergence.

One characteristic of the mean shift vector is that it always points toward the direction of the maximum density. The converged centers then correspond to modes or the centers of the regions of high concentration of data. The mean shift procedure consists of two steps: the estimation of the gradient of the density function, and the utilization of the results to form clusters. The details of an implementation of the mean-shift algorithm based on clustering method are provided in [25]. The mean shift based robust clustering for the number of clusters is automatically obtained by finding the centers of the densest regions in the space (the modes) (See [26]).

2.7 Entropy

In information theory, information entropy is a measure of the average information content associated with the outcome of a random variable. Entropy is defined as in the context of a probabilistic model, and entropy is taken as information content. Consider a random variable $X = x_k, k = -K \leq k \leq K$, where x_k is a discrete time number and $(2K+1)$ is the total number of discrete levels.

Let the event $X = x_k$ occur with probability

$$p_k = P(X = x_k) \quad (2.13)$$

with the requirement that,

$$0 \leq p_k \text{ and } \sum_{k=-K}^K p_k = 1 \quad (2.14)$$

Suppose that the event $X = x_k$ occurs with probability $p_k = 1$, which therefore requires that $p_i = 0$ for all $i \neq k$. In such a situation there is no “surprise” and, therefore, no “information” conveyed by the occurrence of the event $X = x_k$, since we know that the message must be. If, on the other hand, the various discrete levels were to occur with different probabilities and, in particular, the probability p_k is low, then there is more

“surprise” and, therefore, “information” when X takes the value x_k rather than another value x_i with higher probability p_i , $i \neq k$. Thus, the words “uncertainly”, “surprise” and “information” are related. Before the occurrence of the event $X = x_k$ occurs, there are a quantity of surprise. After the occurrence of the event $X = x_k$, there is an increase in the amount of information. These three amounts are obviously the same. Moreover, the quantity of information is related to the inverse of the probability of occurrence. The amount of information gained after observing the even $X = x_k$ with probability p_k is defined as the logarithmic function

$$I(x_k) = \log\left(\frac{1}{p_k}\right) = -\log(p_k) \quad (2.15)$$

where the base of the logarithm is arbitrary. When the natural logarithm is used the units for information are *nats*, and when the base 2 logarithm is used the units are *bits*. In any case, the definition of information given in 2.15 exhibits the following properties:

1. $I(x_k) = 0$ for $p_k = 1$. Obviously, if we are absolutely certain of the outcome of an event, there are no information gained by the occurrence.
2. $I(x_k) \geq 0$ for $0 \leq p_k \leq 1$. That is the occurrence of an event $X = x_k$ either provides some or no information, but it never results in a loss of information.
3. $I(x_k) > I(x_i)$ for $p_k < p_i$. That is, less probable an event is, the more information are gained through its occurrence. The amount of information $I(x_k)$ is a discrete random variable with probability p_k . The mean value of $I(x_k)$ over the complete range of $2K + 1$ discrete values is given by

$$\begin{aligned} H(X) &= E[I(x_k)] \\ &= \sum_{k=-K}^K p_k I(x_k) \\ &= - \sum_{k=-K}^K p_k \log(p_k) \end{aligned} \quad (2.16)$$

The quantify $H(X)$ is called the entropy of a random variable X permitted to take a

finite set of discrete values; it is so called in recognition of the analogy between the definition given in Eq. (2.16) and that of entropy in statistical thermodynamics. The entropy $H(X)$ is a measure of the average amount of information conveyed per message. Note, however, that the X in $H(X)$ is not an argument of a function but rather a label for a random variable (see the details in [22]).



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER III

PROPOSED METHOD

The goal of this work is to consider the underdetermined blind source separation (BSS) problem of separating n source signals, \mathbf{S} , from m observed signals \mathbf{X} , when the observations are a linear combination of the sources using an unknown mixing matrix $\hat{\mathbf{A}}$, $\mathbf{X}=\mathbf{A}\mathbf{S}$. The underdetermined condition means that the number of source signals is greater than the number of output signals, $n > m$. To solve the problem, this dissertation proposes an inner point removal based on information index and perturbed mean shift algorithm to estimate the columns in the mixing matrix for underdetermined BSS with the unknown number of sources. The proposed algorithm also gives the estimated number of sources.

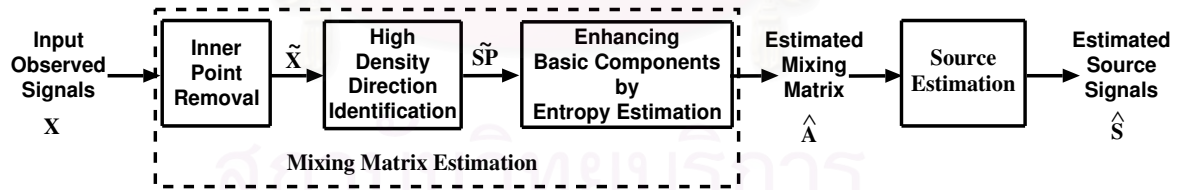


Figure 3.1: The overview of all our work.

In this dissertation, a two-step approach for blind source separation is presented. In the first step, the mixing matrix is estimated by the proposed algorithm. Then, in the second step, the estimated sources are recovered by using minimum ℓ_1 -norm method from the observed signals, \mathbf{X} , and the estimated mixing matrix, $\hat{\mathbf{A}}$ obtained from the first step. Therefore, this dissertation mainly concentrates on the first step.

The proposed method consists of three phases: Inner Point Removal, High Density Direction Identification, and Enhancing Basic Components by Entropy Estimation. In the first phase, the irrelevant inner points are removed from the input observed signals, \mathbf{X} , and each point of the remaining data is redefined by its corresponding adjacent points which lay in the same direction. The remaining redefined data from phase 1 is denoted as $\tilde{\mathbf{X}}$. In the next phase, all possible directions of high density regions are identified by using the perturbed mean shift procedure. This is the extension of the classical mean shift concept based on clustering technique [25]. The converged mean shift vectors or sample points are denoted as $\tilde{\mathbf{SP}}$. Finally, some points in the same direction are removed and the high density directions are obtained by using entropy theory. The remaining high density directions are taken as the directions of the columns of the estimated mixing matrix, $\hat{\mathbf{A}}$, and the number of obtained columns is taken as the estimated number of sources. The overview of all our works is shown in Figure 3.1.

3.1 Estimating the Mixing Matrix

In this section, the proposed method for identifying the mixing matrix based on information index removal and perturbed mean shift algorithm is discussed.

For sparse source signals, at any particular time step, there is only one source signal appears [1]. These signals after being mixed by a mixing matrix can be viewed as a distribution of points clearly clustered along the basic independent components or the columns of the mixing matrix. Obviously, this situation is easy to estimate the number of columns in the mixing matrix. In this dissertation, the sparseness of the signals does not require. The actual basic independent components must be identified from the directions of maximum data density. Many methods based on clustering technique have been presented to estimate the mixing matrix such as a standard k-mean method,

Fuzzy-C clustering algorithm and AICA method [7]. However, it is well known that when the sources are insufficiently sparse, a precise estimate of the mixing matrix is difficult to handle with the exist clustering based algorithms. This is because they need a precondition that the sources must be very sparse in the analyzed domain. Moreover, several methods for estimating the mixing matrix are very sensitive to the irrelevant inner points. Therefore, in the proposed algorithm, the first task is to identify which data points contain enough information about the correct directions of the mixing matrix. All the points for the m -dimensional vector space of the mixtures are observed which points lie in the sparse regions, and affect the correctness of estimating process. They may be considered as irrelevant points and must be eliminated. The aim of this part is to reduce the computational burden of the remaining estimation process, and also to remove those columns that do not contain much information for estimating the mixing matrix. After discarding some irrelevant points, in the second task, the remaining points are used to estimate the number of sources and also the directions of the mixing matrix by using the perturbed mean shift technique with entropy measure.

3.1.1 Inner Point Removal

The purpose of this section is to observe which data points are irrelevant inner points and affect the correctness of mixing matrix identification. Since, the directions of the columns in the mixing matrix correspond to the directions of maximum of data density. Thus, all the data points are examined. If they contain less information or lie in the sparse region, then such points are removed to enhance the correctness of basic component identification. Note that the word “information” in this section has different meaning from that in the information theory as described in Section 2.7. Since the main process is to find the filtering threshold for eliminating the points containing less information,

the first task is to compute the information index of each data point. Before finding the information index of each point, the adjacent region is firstly sought where all points lie along the same direction of such point.

To determine which point \mathbf{y} lies along the same direction of a considered point \mathbf{x} , the closest distance between such two points is computed by the following equation

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{|\langle \mathbf{y}, \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle^2|} \quad (3.1)$$

where $\langle *, * \rangle$ denotes the inner product or dot product and a point \mathbf{x} is normalized (i.e it has a unit length). A set of the adjacent points, \mathbf{AP}_x , with respect to the point \mathbf{x} can be defined by the following equation

$$\mathbf{AP}_x = \{\mathbf{y}_j | D(\mathbf{x}, \mathbf{y}_j) < \alpha, j = 1, \dots, \tau\} \quad (3.2)$$

where α is a predefined distant threshold, τ is a predefined window search length, and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_\tau\}$ is a set of neighboring points of the point \mathbf{x} in the time sequence of input data set.

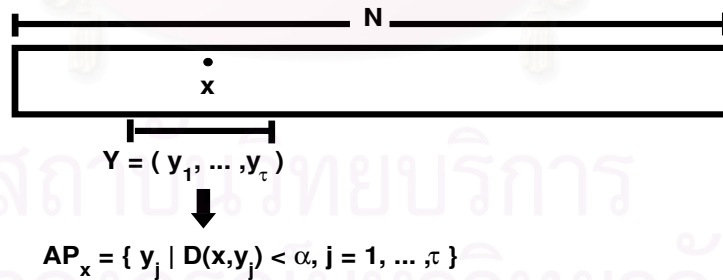


Figure 3.2: A set of τ connected points around the data point \mathbf{x} and the adjacent points \mathbf{AP}_x .

In Figure 3.2 to find the adjacent points that lie along the same direction of the considering point \mathbf{x} , a set of τ connected points around the point \mathbf{x} is selected to \mathbf{Y} . Then, the distance between point \mathbf{x} and each selected point \mathbf{y}_j is computed by using Eq. (3.1).

If the distance value $D(\mathbf{x}, \mathbf{y}_j)$ is less than a predefined distant threshold α , then such data point \mathbf{y}_j is assigned to a set of adjacent points $\mathbf{AP}_{\mathbf{x}}$. Consequently, the column indices in the obtained set of adjacent points $\mathbf{AP}_{\mathbf{x}}$ can be arbitrarily disconnected. Note that the value of τ is much smaller than the number of input data N ($\tau \ll N$). Figure 3.3 shows an example of point \mathbf{y}_j which does not lie along the direction of the considering point \mathbf{x} , because the distance between \mathbf{x} and \mathbf{y}_j is greater than the threshold α , $D(\mathbf{x}, \mathbf{y}_j) > \alpha$.

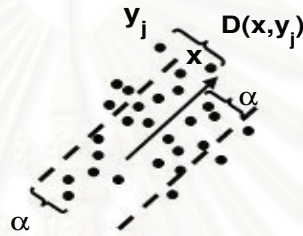


Figure 3.3: An example of the adjacent point being the same direction of \mathbf{x} .

To find the filtering threshold for discarding some irrelevant points, the information index of each point is quantified by using the dot product between two data vectors. If the two vectors are co-incident along the same direction and have a unit length, the absolute value of the dot product is 1. If they are orthogonal to each other, that absolute value becomes zeros. The absolute dot product value depends on the length and the different direction of the two vectors. Consequently, the information index of a considered point is defined by collecting the absolute dot product values between such point and any points in the set of its adjacent points in the same direction. Note that if such point lies in the sparse region, then it contains less information for estimating the directions of mixing matrix.

The information index of any set of adjacent points $I'(\mathbf{AP}_{\mathbf{x}_i})$ is defined as

$$I'(\mathbf{AP}_{\mathbf{x}_i}) = \sum_{j=1}^k |\langle \mathbf{x}_i, \mathbf{y}_j \rangle| \quad (3.3)$$

where k is the size of $\mathbf{AP}_{\mathbf{x}_i}$. The value of $I'(\mathbf{AP}_{\mathbf{x}_i})$ is large if there are more adjacent points aligned along the direction of \mathbf{x}_i , and it is small if \mathbf{x}_i lies in a region with less adjacent points aligned along the considered direction. The data points with small $I'(\mathbf{AP}_{\mathbf{x}_i})$ are discarded because they are located in the direction of sparse region and are less significant for estimating the directions of the columns of the mixing matrix. Furthermore, with the use of dot product technique, the points at the region around zero will have small density and will be removed. The result will not affect the next estimating process, because, in noisy settings, these points contain less information about the correct directions of the mixing matrix.

To determine which points will be removed, the average value of all information indices, I'_{avg} , and the standard deviation value, I'_{std} , of all the information indices $I'(\mathbf{AP}_{\mathbf{x}_i})$ are computed. Both values are used as a threshold, ϑ , for discarding the irrelevant inner points as follows:

$$\vartheta = I'_{avg} + \frac{I'_{std}}{2} \quad (3.4)$$

Any point \mathbf{x}_i having $I(\mathbf{AP}_{\mathbf{x}_i})$ less than ϑ , will be discarded. $\sigma/2$ is used as a threshold to filter only the points locating within the half variance. Thus, most remaining points will be clustered along the directions of maximum data density. This makes it is easy to identify the actual basic independent components in the next steps.

Before the mixing matrix identification, a new defined point $\tilde{\mathbf{x}}_i$ corresponding \mathbf{x}_i is constructed from the summation of all adjacent points in $\mathbf{AP}_{\mathbf{x}_i}$. Note that, all points in $\mathbf{AP}_{\mathbf{x}_i}$ are the points aligning along the direction of the considered point \mathbf{x}_i , thus, the value of the new defined point will be large, if there are many corresponding adjacent points. After the points containing less information are removed, each remaining point $\tilde{\mathbf{x}}_i$ is computed by the following equation

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^k (\mathbf{x}_i + \mathbf{y}_j * \text{sign}(\langle \mathbf{x}_i, \mathbf{y}_j \rangle)) \quad (3.5)$$

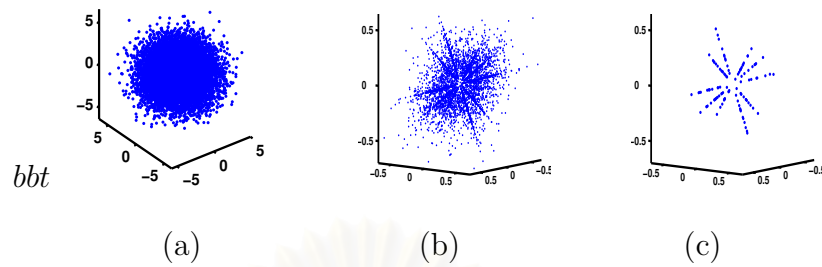


Figure 3.4: Result on the experiment 1 on data set 1. (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process.

Note that, only point \mathbf{x}_i whose information index $I'(\mathbf{AP}_{\mathbf{x}_i})$ is larger than or equal to the filtering threshold, ϑ , is redefined. This process produces the new data points, $\tilde{\mathbf{X}}$, in which each data point has a high density distribution of the adjacent points in the corresponding direction.

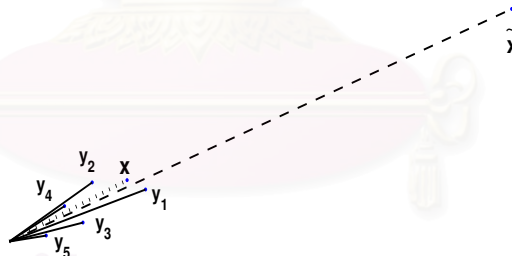


Figure 3.5: A new defined point $\tilde{\mathbf{x}}$ of a point \mathbf{x} with $\mathbf{AP}_{\mathbf{x}} = \mathbf{y}_1, \dots, \mathbf{y}_5$.

Fig 3.5 shows an example of a new defined point $\tilde{\mathbf{x}}$ of a point \mathbf{x} and its corresponding adjacent points $\mathbf{AP}_{\mathbf{x}}$. Let $\mathbf{AP}_{\mathbf{x}} = \{\mathbf{y}_1, \dots, \mathbf{y}_5\}$ (i.e. $\mathbf{D}(\mathbf{x}, \mathbf{y}_i) < \alpha, \forall \mathbf{y}_i \in \mathbf{AP}_{\mathbf{x}}$). Therefore, a new defined point $\tilde{\mathbf{x}}$ of the point \mathbf{x} can be represented as

$$\tilde{\mathbf{x}}_i = (\mathbf{x}_i + \mathbf{y}_1) + (\mathbf{x}_i + \mathbf{y}_2) + (\mathbf{x}_i + \mathbf{y}_3) + (\mathbf{x}_i + \mathbf{y}_4) + (\mathbf{x}_i + \mathbf{y}_5) \quad (3.6)$$

Note that, if \mathbf{y}_i has an opposite direction of the point \mathbf{x} , then \mathbf{y}_i is changed to be $-\mathbf{y}_i$. As depicted with a dash line in the figure, the redefined point $\tilde{\mathbf{x}}$ has a few changes in

the direction to the left side which more the number of adjacent points of the point \mathbf{x} (depicted with a dot line). Since the length of the redefined point $\tilde{\mathbf{x}}$ depends on the number of adjacent points in $\mathbf{AP}_{\mathbf{x}_i}$ and the length of each point $y_i \in AP_{x_i}$, any direction that has more the number of points will dominate other directions with smaller adjacent points. Compared with the original data space (Figure 3.4(a)), the increased sparseness can be seen in the scatter plots of the new defined data as shown in Figure 3.4(b). After the points with small information indices are removed, most remaining points in the obtained space are clustered along the directions of maximum of data density. The overall algorithm in this phase is described in **Algorithm InnerPointRemoval**.

Algorithm Inner Point Removal

Input: the set of analyzed data \mathbf{X} .

Output: the new set of the redefined data $\tilde{\mathbf{X}}$.

Begin

1. **For** each point $\mathbf{x}_i \in \mathbf{X}$ **Do**
2. Let $\mathbf{AP}_{\mathbf{x}_i}$ be a set of k adjacent points aligning in the direction of \mathbf{x}_i using Eq. (3.2).
3. Find the information index $I'(\mathbf{AP}_{\mathbf{x}_i})$ of the current point \mathbf{x}_i by using Eq. (3.3).
4. **End.**
5. Compute the average value I'_{avg} and the standard deviation I'_{std} from all information indices $I'(\mathbf{AP}_{\mathbf{x}_i})$
6. Find the filtering threshold, $\vartheta = I'_{avg} + \frac{I'_{std}}{2}$.
7. Remove all points \mathbf{x}_i in which the corresponding information index $I'(\mathbf{AP}_{\mathbf{x}_i})$ is less than ϑ .
8. Compute a new defined value of $\tilde{\mathbf{x}}_i$ of \mathbf{x}_i and the set of its corresponding adjacent points $\mathbf{AP}_{\mathbf{x}_i}$ (in step 2) by using Eq. (3.5), for all remaining points \mathbf{x}_i .

End

3.1.2 High Density Direction Identification

As shown in Figure 3.4(b), the representation of the remaining data attained from the previous phase is sparser. Based on this sparseness of the obtained data space, each column of the mixing matrix can be estimated by finding the directions of maximum data density. This can be achieved by using the property of mean shift technique. As described in the previous section, the mean shift vector always points toward the direction of the maximum increase in the density and the algorithm converges to the closest high density region. However, because the distribution of some analyzed data has a high density in the region around zero and the density in an closet region is very high, the original mean shift procedure may shift some mean shift vectors to the other closest regions and makes some directions disappeared.

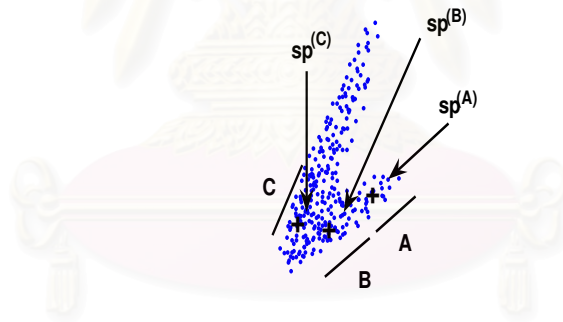


Figure 3.6: An example of two regions which have more overlapped data.

Figure 3.6 shows two directions of maximum density regions. Assume that, in the right direction, the density of data in area B is higher than that in area A , i.e., the density is increased from area A to area B . Let $\mathbf{sp}^{(A)}$ be a considered sample point or mean shift vector, and lies in the right direction of area A . The mean shift vector $\mathbf{sp}^{(A)}$ starts at the location in the area A . After performing k iterations of the original mean shift procedure, $\mathbf{sp}^{(A)}$ is directly replaced with a next computed mean value for each iteration. $\mathbf{sp}^{(A)}$ will be moved toward the area B , denoted by $\mathbf{sp}^{(B)}$, because of the

increase of the density. Moreover, if the area C has the higher density than that of area B and more data are overlapped between both areas, then, for some next iterations of mean shift procedure, $\mathbf{sp}^{(B)}$ may be moved to area C , denoted by $\mathbf{sp}^{(C)}$. Now, $\mathbf{sp}^{(C)}$ changed its direction. As a result, if all the sample mean shift vectors laying in the right direction have the same behavior as $\mathbf{sp}^{(A)}$, then no mean shift vector represents the high density in the left direction. To avoid this event, in the proposed algorithm, the length of all mean shift vectors will be preserved. This is the first difference between the proposed algorithm and other exist clustering methods based on mean shift technique in [24],[25] and [26]. Moreover, step 10 in **Algorithm DirectionIdentifying** is used to perturb each sample point for the convergence as follow

$$\mathbf{sp}_j = \frac{E(\psi_j)}{\|E(\psi_j)\|} * \|\mathbf{sp}_j\|, \forall \mathbf{sp}_j \in \mathbf{SP} \quad (3.7)$$

where ψ_j is a set of $\tilde{\mathbf{x}}_i \in \tilde{\mathbf{X}}$ that lies in the same direction of \mathbf{sp}_j and $E(\psi_j)$ is the average of all points in ψ_j . Note that, the length of each sample point is no changed. It is found that the use of the mean of all points in the same direction makes us to get more smoothing lines than use of a small real value to perturb the points.

It is obvious that each mean shift vector is converged to the closest region of high concentration of data. As shown in Figure 3.4(c), the converged mean shift vectors, denoted by $\tilde{\mathbf{SP}}$, form the lines in the direction of the maximum density. The detailed algorithm of the direction estimation is described in the **Algorithm DirectionIdentifying**.

3.1.3 Enhancing Basic Components by Entropy Estimation

In this phase, the converged sample points obtained from the perturbed mean shift algorithm in high density direction identification process are separated into different direction groups, denoted by \mathbf{G} . There are the points in same group locating in the same direction. For each group, the maximum eigenvalue λ_k is computed and represented as

Algorithm DirectionIdentifying

Input: the redefined data $\tilde{\mathbf{X}}$.

Output:the converged sample points $\tilde{\mathbf{SP}}$.

Begin

1. Initial a set of sample points \mathbf{SP} from the input data $\tilde{\mathbf{X}}$.
2. **Repeat**
3. **Repeat**
4. **For** each sample point \mathbf{sp}_j **Do**
5. Let \mathbf{NBS} be a set neighboring points of \mathbf{sp}_j using a spherical window of radius h .
6. Compute the mean NBS_{avg} of all points in \mathbf{NBS} .
7. Update the current sample points using the equation $\mathbf{sp}_j = \frac{NBS_{avg}}{\|NBS_{avg}\|} * \|\mathbf{sp}_j\|$.
8. **End.**
9. **Until** The number of sample points changed is less than a small threshold value.
10. Perturb each sample point \mathbf{sp}_j using Eq.(3.7)
11. **Until** No sample point is changed.

End

the area of the corresponding group g_k . Any group k whose λ_k is less than a preset threshold β_1 is removed. Then, the density of each remaining group is calculated by using the following equation

$$z_k = \frac{n_{g_k}}{\lambda_k}, k = 1 \dots K \quad (3.8)$$

here n_{g_k} is the number of member points in group k and K be the number of groups in different directions.

To determine that any group is suitable to represent the direction of maximum density (i.e. direction of columns of mixing matrix), the entropy of each group is computed

as follows

$$p_k = \frac{z_k}{\sum_{j=1}^K z_j}, j = 1 \dots K \text{ and } k = 1 \dots K \quad (3.9)$$

where p_k is the probability of group k , and $\sum_{k=1}^K p_k = 1$. Any group whose entropy is less than a predefined entropy threshold β_2 is removed. This enhancing process is summarized in **Algorithm PointRefining**. The eigenvectors with maximum variance of remaining groups are obtained and denoted by $\hat{\mathbf{A}}$, and the number of columns in $\hat{\mathbf{A}}$ is then taken as the estimated number of sources.

Algorithm PointRefining

Input: the converged sample points $\hat{\mathbf{S}}\mathbf{P}$.

Output: the estimated mixing matrix $\hat{\mathbf{A}}$.

Begin

1. Cluster the converged sample points $\hat{\mathbf{S}}\mathbf{P}$ into different direction groups denoted by \mathbf{G} , where the member points in each group are the sample points that lie in the same direction (locating in the same line).
2. Compute the maximum eigenvalue λ_k from the member points in each group $g_k, \forall g_k \in \mathbf{G}$
3. Remove group g_k whose λ_k is smaller than β_1 .
4. Find the density z_k of each group g_k by using Eq. (3.8).
5. Compute the information entropy and remove group g_k where the information $p_k \log p_k < \beta_2$
6. Find eigenvector e_k with highest variance for each remaining group g_k , and store in $\hat{\mathbf{A}}$.

End

3.2 An Example of Mixing Matrix Detection

This section shows an example of how the estimated mixing matrix is identified on data set 1 (Figure 3.7). The input data set \mathbf{X} consists of three observed signals linearly mixed from five simulated source signals \mathbf{S} with the following normalized mixing matrix $\mathbf{A} \in R^{3 \times 5}$.

$$\mathbf{A} = \begin{bmatrix} 0.8412 & -0.0298 & -0.0750 & -0.1735 & -0.7240 \\ -0.5025 & 0.8305 & -0.8294 & 0.3621 & 0.4088 \\ 0.1997 & 0.5563 & 0.5536 & 0.9158 & -0.5556 \end{bmatrix} \quad (3.10)$$

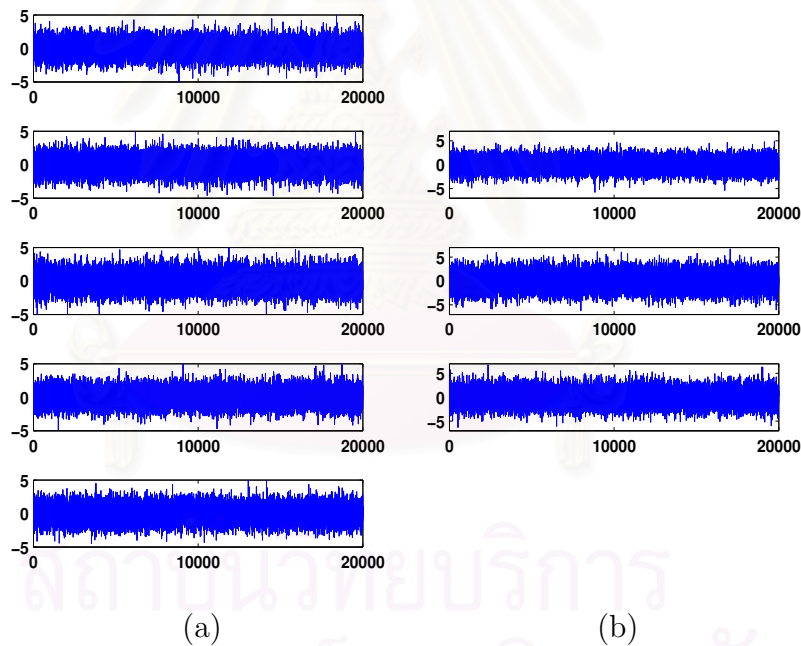


Figure 3.7: Data set 1 (a) Five simulated source signals (b) Three observed signals.

The five source signals are generated from five normal distributions, $N(0, 1)$, as shown in the first column of Figure 3.7(a). Each source signal contains 20,000 points with the value range $[-5, 5]$ and only 10% of sparse source vectors. That is, the entries of 18,000 columns of source matrix \mathbf{S} have a uniform distribution and other 2,000 columns have only one nonzero entry and four zeros entries and also have the uniform distribution.

Furthermore, the column indices of the 2,000 columns in the matrix \mathbf{S} are disconnected. Thus, the 2,000 columns are divided into five sets containing 400 columns each and the i^{th} entries of the i^{th} set have nonzero values. The column indices of the 400 columns of the i^{th} set are as follows: $(i-1) \times 10 + 1, (i-1) \times 10 + 51, \dots, (i-1) \times 10 + 1, 951, i = 1 \dots 5$. Figure 3.4(a) is the scatter plot of three observed signals \mathbf{X} .

The algorithm starts with finding the information indices of all points and, then, the filtering threshold ϑ is computed to discard irrelevant inner points. Figure 3.4(b) shows the results after inner point removal process with the window search length $\tau = 200$. The sparseness in scatter plot makes the mixing directions clearly identifiable because the points containing less information are discarded and the points around the zero are also removed. In the next process, with the property of mean shift method all the sample points is, then, shift to the directions of the maximum increase in the density. The result shows in Figure 3.4(c). Finally, the points being the same or opposite in direction are removed by using **Algorithm PointRefining**. Thus, the estimated mixing matrix $\hat{\mathbf{A}}$ is obtained as shown below

$$\hat{\mathbf{A}} = \begin{bmatrix} 0.0321 & -0.1727 & -0.8396 & -0.0748 & 0.7234 \\ -0.8315 & 0.3617 & 0.5049 & -0.8287 & -0.4129 \\ -0.5546 & 0.9161 & -0.2011 & 0.5547 & 0.5533 \end{bmatrix} \quad (3.11)$$

The number of columns in the recovered mixing matrix is equal to that of the actual mixing matrix. But the estimated mixing matrix, $\hat{\mathbf{A}}$, exhibits permutation and also sign reversal for some sources.

CHAPTER IV

EXPERIMENTAL RESULTS

The proposed algorithm has been implemented and tested for the validity and performance on a 1.33 GHz Intel Pentium IV based computer with 512 MB of RAM. In the proposed algorithm, there exist five parameters which should be preset. Eigenvalue threshold, β_1 , and entropy threshold, β_2 , are related to the density of data. Let $\lambda_{max} = \max(\lambda_k)$, where λ_k represents the variance of data in direction k . β_1 can be set to a fraction of λ_{max} to remove some directions with small length and number of points. β_2 is a small value to remove some directions which less effect to the entropy. Window search length, τ can be set to be any positive integer much less the number of data points N . Distant threshold, α , and window radius, h , are related to the precise direction of the mixing matrix estimation. They should be set to make sure that all directions can be identified by the proposed algorithm. In all experiments, all parameter values used in all experiments are shown in the Table 4.1.

4.1 Performance Measures

To validate the performance of the algorithm, the difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual mixing matrix \mathbf{A} is measured by using **Algebraic Matrix-Distance Index (AMDI)** proposed in [7].

Assume that both the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$ are column normalized (i.e. the norm of the columns in the matrices is one). Then,

Table 4.1: The parameters used in the experiments.

Parameter	Selected Value
Window search length, τ	$0.01N$
Distant threshold, α	$\pi/180 \approx 0.0175$
Window radius, h	$\pi/180 \approx 0.0175$
Eigenvalue threshold, β_1	$0.1\lambda_{max}$
Entropy threshold, β_2	0.05

Algebraic Matrix-Distance Index (AMDI) between \mathbf{A} and $\hat{\mathbf{A}}$ is defined as follows:

$$\text{AMDI}(\mathbf{A}, \hat{\mathbf{A}}) = \frac{n - \sum \max_{rows}(|\hat{\mathbf{A}}^T * \mathbf{A}|)}{n} + \frac{n - \sum \max_{cols}(|\hat{\mathbf{A}}^T * \mathbf{A}|)}{n} \quad (4.1)$$

where n is the number of sources. With the property of dot product, if two unit vectors are in the same direction, the value becomes one. Therefore, if the columns of the estimated mixing matrix $\hat{\mathbf{A}}$ are very close to the columns of the actual mixing matrix \mathbf{A} , the AMDI index approaches zero.

The AMDI has the following properties:

- $0 \leq \text{AMDI}(\mathbf{A}, \hat{\mathbf{A}}) \leq 2$; $\text{AMDI}(\mathbf{A}, \hat{\mathbf{A}}) = 2$ only if both matrices are orthogonal to each other in the n -dimensional space.
- $\text{AMDI}(\mathbf{A}, \hat{\mathbf{A}}) = \text{AMDI}(\hat{\mathbf{A}}, \mathbf{A})$.

4.2 Experimental Design

The experiments in this dissertation consist of five parts. Experiment 1 was performed on all data sets to compare the proposed method and the literatures [13] and [14], for the unknown source number case of the underdetermined blind source separation prob-

lem. Moreover, the experiment was also used to show the performance of the proposed algorithm on the data sets with the small fraction of sparse source vectors and the disconnected region of sparse source vectors. Experiment 2 was performed on three sets of speech signals to illustrate the detail results each phase of the proposed algorithm. Moreover, the obtained results were compared to the results from a standard k-mean method and AICA method. In experiment 3, the proposed algorithm was performed on six speech signals of data set 8. The comparison of the proposed method and k-mean method was discussed again in both time domain and frequency domain. Experiment 4 was used to measure the performance of the proposed algorithm on the data set in noisy environment. The last experiment showed the relationship between the percent of sparse vectors in source matrix S , the window search length parameter τ and the average value of information indices, I'_{avg} , of all the information indices $I'(\mathbf{AP}_{\mathbf{x}_i})$. The details of each experiment are explained in the next sections.

The data sets used in all experiments are given details in Table 4.2. They consist of two groups: four synthetic data sets in the first four rows and four real speech data sets in the remaining rows. The second column displays the number of sample points contained in each data set. The number of sources and the number of observed signals of each data set are demonstrated in the second and third columns. The fraction of sparse source vectors of each data set is illustrated in the last column, except the speech data sets on the last four rows.

4.2.1 Synthetic Data Sets

Each synthetic data set was in 3-dimensional data space and contained 20,000 (N) points with the value range $[-5,5]$. The four synthetic data sets were generated with the different percents of sparse source vectors, a vector with only one nonzero entry, as depicted in

Table 4.2: Data sets.

	Number of Sample Points	Number of Sources	Number of Observed Signals	Percent of Sparse Vectors
Data set 1	20,000	5	3	10%
Data set 2	20,000	5	3	20%
Data set 3	20,000	5	3	50%
Data set 4	20,000	10	3	10%
Data set 5	62,500	3	2	-
Data set 6	62,500	6	3	-
Data set 7	23,000	3	2	-
Data set 8	30,000	6	3	-

the last column of the first four rows in Table 4.2. Each source signal was generated from a normal distribution, $N(0, 1)$. In additions, the indices of all the sparse source columns of each data set were not disjointed as described in Section 3.2.

- **Data set 1** was of three observed signals \mathbf{X} linearly mixed from five synthetic source signals \mathbf{S} with 10% of the sparse sources vectors and a random and normalized mixing matrix $\mathbf{A} \in R^{3 \times 5}$ as follows

$$\mathbf{A} = \begin{bmatrix} 0.8412 & -0.0298 & -0.0750 & -0.1735 & -0.7240 \\ -0.5025 & 0.8305 & -0.8294 & 0.3621 & 0.4088 \\ 0.1997 & 0.5563 & 0.5536 & 0.9158 & -0.5556 \end{bmatrix} \quad (4.2)$$

The five source signals and three observed signals of this data set are illustrated in Figure 3.7.

- **Data set 2** and **data set 3** were of three observed signals. They were linearly mixed from two different sets of five synthetic source signals with 20%, and 50% of sparse source vectors, respectively. Both data sets were mixed with the same mixing matrix \mathbf{A} in Eq.(4.2).
- **Data set 4** was of three observed signals linearly mixed from ten synthetic sources signals with 10% of the sparse sources vectors and a randomly mixing matrix $\mathbf{A} \in R^{3 \times 10}$ as shown below

$$\mathbf{A} = \begin{bmatrix} 0.6206 & 0.0765 & 0.4253 & -0.8558 & 0.8412 & -0.0298 & -0.0750 & -0.1735 & -0.7240 & 0.1594 \\ 0.7619 & -0.5920 & -0.0675 & 0.4717 & -0.5025 & 0.8305 & -0.8294 & 0.3621 & 0.4088 & -0.9029 \\ -0.1856 & 0.8023 & 0.9025 & 0.2126 & 0.1997 & 0.5563 & 0.5536 & 0.9158 & -0.5556 & -0.3991 \end{bmatrix}$$

4.2.2 Real Speech Data Sets

Four speech data sets including data sets 5, 6, 7 and 8 are presented in the last four rows of Table 4.2).

- **Data Set 5** was obtained from <http://www.egr.msu.edu/results/ijcnn2003> [7]. Each of three speech source signals contained 62,500 sample points. These speech source signals, \mathbf{S} were mixed into two observed signals \mathbf{X} with a randomly mixing matrix $\mathbf{A} \in R^{2 \times 3}$ as follows

$$\mathbf{A} = \begin{bmatrix} 0.7071 & -0.4472 & -0.9487 \\ 0.7071 & 0.8944 & 0.3162 \end{bmatrix}$$

- **Data set 6** also is speech signals from [7]. But this data set is of two observed signals \mathbf{X} mixed from six speech signals \mathbf{S} with a randomly mixing matrix $\mathbf{A} \in R^{3 \times 6}$ as follows

$$\mathbf{A} = \begin{bmatrix} 0.6206 & 0.0765 & 0.4253 & -0.8558 & 0.9965 & 0.1594 \\ 0.7619 & -0.592 & -0.0675 & 0.4717 & -0.0646 & -0.9029 \\ -0.1856 & 0.8023 & 0.9025 & 0.2126 & -0.0523 & -0.3991 \end{bmatrix}$$

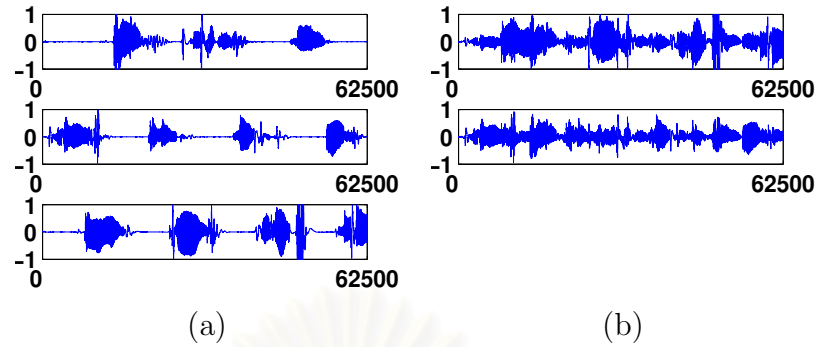


Figure 4.1: Data set 5 (a) Three speech signals from [7]. (b) Two observed signals.

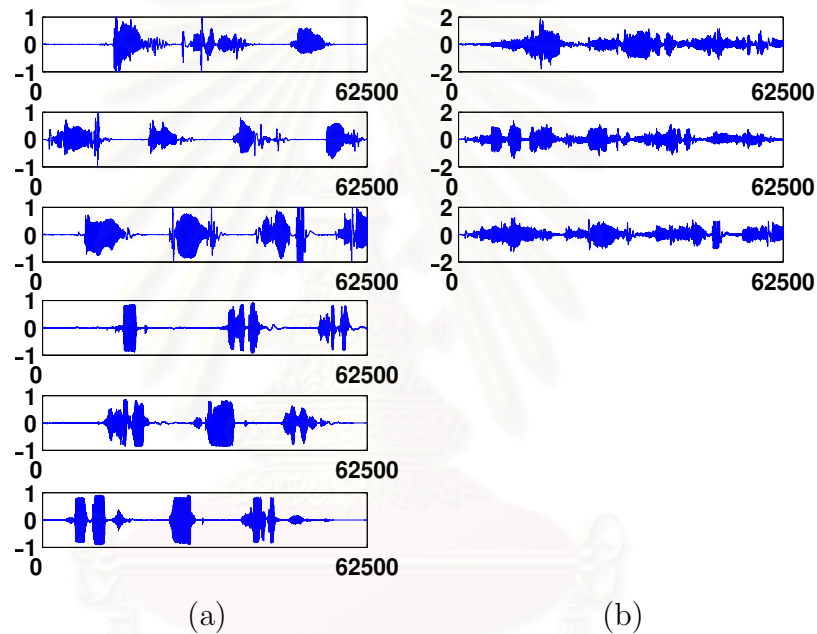


Figure 4.2: Data set 6 (a) Six speech signals from [7]. (b) Three observed signals.

- **Data set 7** from <http://www.cnl.salk.edu/~tewon/Over> [3] was of two observed signals \mathbf{X} . Each signal contained 23,000 sample points as shown in Figure. 4.3. They were mixed from three speech source signals \mathbf{S} by using the following mixing matrix $\mathbf{A} \in R^{2 \times 3}$.

$$\mathbf{A} = \begin{bmatrix} 0 & 0.7071 & 0.7071 \\ 1 & 0.7071 & -0.7071 \end{bmatrix}$$

- **Data set 8** was of six speech source signals \mathbf{S} from the TIMIT database as demonstrated in Figure 4.4(a). All signals contained 30,000 sample points each and were

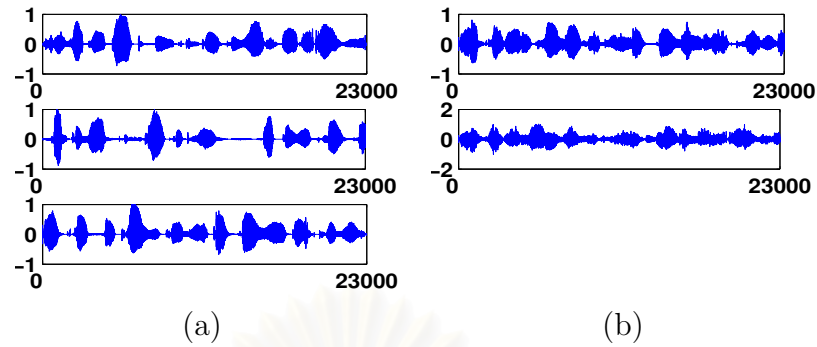


Figure 4.3: Data set 7 (a) Three speech signals from [3]. (b) Two observed signals.

mixed in three observed signals the \mathbf{X} (Figure 4.4(b)) by using a 3×6 dimensional mixing matrix $\mathbf{A} \in R^{3 \times 6}$ which was selected randomly and, then, normalized to unit length as follows.

$$\mathbf{A} = \begin{bmatrix} -0.7481 & -0.5355 & -0.3513 & -0.5701 & -0.2204 & -0.8431 \\ -0.6431 & 0.6187 & 0.9346 & -0.6691 & 0.1883 & -0.0494 \\ 0.1638 & -0.5748 & -0.0560 & -0.4769 & -0.9571 & 0.5354 \end{bmatrix}$$

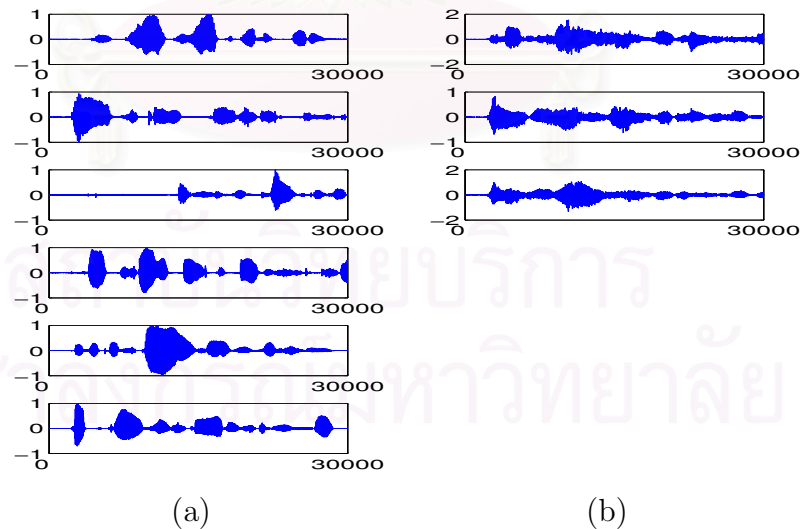


Figure 4.4: Data set 8 (a) Six speech signals from TIMIT database. (b) Three observed signals.

4.3 Experiment 1

The result of the proposed algorithm will be compared with an extension of the DUET and TIFROM algorithms [13] and a unified method [14] based on an unsupervised robust C prototypes (URCP) algorithm [29]. In order to perform the comparison, parameters for both algorithms, [13] and [14], are set in different values as illustrated in Table 4.3.

For the extension of the DUET and TIFROM algorithms [13], the second row is the given parameter values which were proposed in [13]. For implementing division operation in the algorithm, Li et al. suggested that ξ_2 should be set as a fraction of Q_2 (e.g., $\xi_2 = 0.1Q_2$), where Q_2 was the maximum of the amplitude of the entries of the data matrix. M_0 was the numbers of bins, which depended on users to make sure that the direction of mixing matrix can be detected (as seen the details of parameter sets in [13]). But from this experimental result shown in Table 4.5 using another real speech signals, it is found that the parameter given in the [13] cannot produce the satisfactory results for all real speech signals. Then, the next three rows are the other parameter values which used in this experiment in order to improve the result for the extension of the DUET and TIFROM algorithms.

For a unified method [14] based on an unsupervised robust C prototypes (URCP) algorithm [29], although the URCP algorithm required some parameters which should be set in advance: α , ϵ and C_{max} [29], Frigui et al. concluded that varying α between 3 and 10 had a small affect on points with small weights. The value of ϵ was used to check the similarity of two clusters and any value between 0.2 and 0.5 for ϵ yield similar results. In order to perform the comparison, the parameter values suggested in [29] were assigned, i.e. $\alpha = 3.5$ and $\epsilon = 0.25$. In additions, C_{max} could be any number that was larger than the expected number of clusters containing in the data set. A very large number of C_{max} only made the URCP algorithm slower. From this experiment, if the

initial cluster number, C_{max} , was set equal to the actual number of sources (n), the estimated mixing matrices did not give any satisfactory results even the several times using the unified method were performed. Therefore, the three cases with the different choices of $C_{max} = 2n, 3n$, and $4n$ were set to perform the unified method as shown in the last three rows of Table 4.3. Similar to the k-mean method, the convergent result of the unified method depended on the initialization. The unified method was performed in several times for each case of C_{max} . The result with minimum AMDI value was selected to show in Table 4.4 and Table 4.5.

Table 4.3: List of algorithms with parameters for comparison of overcomplete ICA algorithms in unknown source number case.

	Algorithm Name	Parameter Value
algo. 1	Proposed algorithm	see Table 4.1
algo. 2.1	An extension of the DUET and TIFROM algorithms	$M_0 = 400, \xi_2 = 0.1Q_2$
algo. 2.2	An extension of the DUET and TIFROM algorithms	$M_0 = 400, \xi_2 = 0.05Q_2$
algo. 2.3	An extension of the DUET and TIFROM algorithms	$M_0 = 200, \xi_2 = 0.1Q_2$
algo. 2.4	An extension of the DUET and TIFROM algorithms	$M_0 = 200, \xi_2 = 0.05Q_2$
algo. 3.1	A unified method	$C_{max} = 2n$
algo. 3.2	A unified method	$C_{max} = 3n$
algo. 3.3	A unified method	$C_{max} = 4n$

Table 4.4 shows the comparison results for the synthetic data sets which include data sets 1, 2, 3 and 4, having number of sources (n) equals to 5, 5, 5 and 10, respectively. Four data sets are generated with 10%, 20%, 50%, and 10% of sparse source vectors as described in Section 4.2.1. For each data set (or each column), the table provides the estimated number of sources \hat{n} and the AMDI value between \mathbf{A} and $\hat{\mathbf{A}}$. The results from

the proposed algorithm is demonstrated in the first row and the remaining rows shows the results obtained from the literatures [13] and [14]. First, consider the estimated number of sources \hat{n} , it can be seen that the estimated number of sources \hat{n} from the proposed method equals to the actual number of sources n for all data sets. Second, for the result of the extension of the DUET and TIFROM algorithms (algo. 2.1 to algo. 2.4), the estimated number of sources \hat{n} equals to the actual number of sources n for all data sets when the value of parameter ξ_2 is reduced to $0.05Q_2$, as shown in rows 3 and 5. However, with over decreasing the value of parameter ξ_2 , it is found that the estimated number of sources will be increased. For an example, when ξ_2 is set to $0.001Q_2$ and $M_0 = 200$, the estimated number of sources, \hat{n} , for the four synthetic data sets are 22, 21, 12, and 32, respectively. Hence, all estimated numbers of sources are higher than the actual number of sources n for all data sets. Third, for the result of the unified method (algo. 3.1 to algo. 3.3), the estimated number of sources \hat{n} is higher than the actual number of sources n for all data sets.

Consider the AMDI value representing the difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$. First, the proposed algorithm can successfully detect the mixing matrices for all data sets, although the percent of sparse source vectors in data sets 1 and 4 is very small. The AMDI values from the proposed algorithm are very small for all data sets. The results show that all estimated mixing matrices are very close to the actual mixing matrices. Second, for the results of the extension of the DUET and TIFROM algorithms (algo. 2.1 to algo. 2.4), all AMDI values are also very small for all data sets when ξ_2 is reduced to $0.05Q_2$. Third, the results of the unified method (algo. 3.1 to algo. 3.3) are shown in the last three rows. The algorithm gives the good estimated mixing matrix $\hat{\mathbf{A}}$ with small AMDI values only on data set 3 which has highest percent of sparse sources vectors, 50% of source column vectors having only one none-zero entry.

Consider the C_{max} value for the unified method [14]. Higher C_{max} values give the smaller AMDI values as seen in row 6 and row 8. However, the AMDI values of case $C_{max} = 3n$ are smaller than ones of the case $C_{max} = 4n$ because the good initialization is very close to the columns of the actual mixing matrix in the $C_{max} = 3n$. In additions, with over specifying of initial value of C_{max} , the estimated number of sources is higher respect to higher initial value C_{max} . Therefore, the unified method fails to provide the good estimated number of sources when C_{max} value is increased.

Table 4.4: The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , for synthetic data set 1 to data set 4.

	Data Set 1 (n=5)		Data Set 2 (n=5)		Data Set 3 (n=5)		Data Set 4 (n=10)	
	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}
algo. 1	6.38E-06	5	3.11E-06	5	9.63E-06	5	1.20E-04	10
algo. 2.1	2.20E-10	5	3.09E-11	5	1.64E-11	5	4.50E-01	6
algo. 2.2	6.47E-09	5	1.09E-09	5	7.88E-11	5	4.18E-08	10
algo. 2.3	2.55E-08	5	5.57E-09	5	1.92E-09	5	4.50E-01	6
algo. 2.4	2.57E-07	5	1.76E-08	5	4.85E-10	5	1.45E-06	10
algo. 3.1	3.04E-02	10	1.07E-02	9	1.89E-03	7	3.34E-02	18
algo. 3.2	8.14E-04	14	0.00	11	0.00	7	9.22E-03	29
algo. 3.3	1.89E-02	20	3.81E-03	15	5.65E-06	11	9.55E-03	38

Concluding remarks for the comparison results on Table 4.4, the unified method gives a smaller AMDI value when the percent of the sparsity increase as seen from the results on all data sets in the last three rows. If the fraction of sparse source column

vectors of data is very small, the unified method fails to provide the precise estimated mixing matrix $\hat{\mathbf{A}}$ and the estimated number of sources as seen from the result on data set 1 and data set 4 with only 10% of sparse source vectors. However, when the unified method is tested on another new data set with 80% of sparse source vectors, it is found that the unified method gives very small AMDI values with 2.92E-04, 1.92E-04, and 1.52E-04 for the cases of C_{max} with $2n$, $3n$, and $4n$, respectively. Moreover, the unified method also gives the correct source number for all cases on the new data set with 80% of sparse source vectors. Therefore, it can be concluded that the performance of the unified method depends on the sparseness of sources.

On the other hand, the results from the extension of the DUET and TIFROM algorithms are more efficient than ones of the unified method for all synthetic data sets. In other words, the algorithm relaxes the condition of sparseness in the unified method.

In order to confirm the performance of the proposed method to the real speech data sets, this experiment is evaluated again with data sets 5, 6, 7 and 8 as described in Section 4.2.2. Table 4.5 illustrates the comparison of the results from the proposed algorithm and the literatures [13] and [14]. Consider the estimated number of sources \hat{n} , the proposed method produces the correct numbers of sources for all data sets (as shown in the first row). On the other hand, the results from other methods fail to estimate the number of sources for all real speech data sets. Moreover, the proposed method gives all good estimated mixing matrices $\hat{\mathbf{A}}$ with small AMDI values.

The extension of the DUET and TIFROM algorithms gives the good results for the estimated mixing matrix only on data set 5 and data 7. For data set 8, all cases of the extension of the DUET and TIFROM algorithms grant the unfortunate results both the AMDI value and \hat{n} . However, for data set 8 when the value of ξ_2 is vary from $0.001Q_2$ to $0.1Q_2$ and $M_0 = 200$, the smallest AMDI value with 0.002 can be found in the case of $\xi_2=0.001Q_2$. However, the value is still high and the obtained source number ($\hat{n} =$

Table 4.5: The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , for real speech data set 5 to data set 8.

	Data Set 5 (n=3)		Data Set 6 (n=6)		Data Set 7 (n=3)		Data Set 8 (n=6)	
	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}
algo. 1	7.40E-06	3	1.78E-03	6	1.92E-05	3	1.93E-04	6
algo. 2.1	1.29E-07	125	2.00	0	6.78E-02	34	2.00	0
algo. 2.2	2.43E-05	85	2.06E-01	7	1.06E-04	81	2.00	0
algo. 2.3	3.77E-06	79	8.00E-02	9	4.72E-05	57	2.00	0
algo. 2.4	7.59E-06	67	9.59E-04	41	2.57E-04	56	2.00	0
algo. 3.1	7.12E-04	6	5.32E-02	10	1.19E-03	6	4.21E-02	12
algo. 3.2	1.45E-04	6	2.65E-03	18	4.18E-04	8	6.36E-03	18
algo. 3.3	1.19E-04	10	2.01E-02	24	4.03E-04	11	1.87E-02	24

36) is higher the actual source number ($n = 6$). For data set 6, although the AMDI value is smallest, compared to other tests, but the estimated number of sources is very high. Third, the unified method (algo. 3.1 to algo. 3.3) gives the good estimated mixing matrix $\hat{\mathbf{A}}$ with the small values of AMDI only for two dimensional data set 5 and data set 7, especially when the initial cluster number, C_{max} , is increased. Similar to the result on synthetic data in the previous table, the estimated source number is increased for the high value of C_{max} .

Like the extension of the DUET and TIFROM algorithms, the unified method cannot estimate the number of sources for all speech data sets. Moreover, the unified method fails to estimate the precise mixing matrix for data set 6 and data set 8.

Note that, for computing the AMDI value, if the estimated number of sources or the column numbers of the estimated mixing matrix $\hat{\mathbf{A}}$ is higher than the column number of the actual mixing matrix \mathbf{A} , then the sub-matrix $\hat{\mathbf{A}}$ which is close to the mixing matrix \mathbf{A} is selected to compute the AMDI value between \mathbf{A} and $\hat{\mathbf{A}}$. Therefore, even the estimated number of sources is very high, if there exists the actual mixing matrix \mathbf{A} as the subset of $\hat{\mathbf{A}}$, then the AMDI value between \mathbf{A} and $\hat{\mathbf{A}}$ is small.

For the results in row 5 (algo. 2.4) of the extension of the DUET and TIFROM algorithms on data sets 5, 6 and 7, the column numbers of $\hat{\mathbf{A}}$, \hat{n} , are 67, 41, and 56, while the column numbers of the actual mixing matrices \mathbf{A} , n , are 3, 6, and 3, respectively. Since all results give higher column number of mixing matrices $\hat{\mathbf{A}}$, only 3, 6 and 3 columns of $\hat{\mathbf{A}}$ which are close to the corresponding actual mixing matrices \mathbf{A} are selected from $\hat{\mathbf{A}}$ to compute the AMDI values for data sets 5, 6 and 7, respectively. Note that, for data set 8, the extension of the DUET and TIFROM algorithms cannot find any column of the estimated mixing matrix $\hat{\mathbf{A}}$, i.e. $\hat{n} = 0$. As a result, the AMDI value of data set 8 is 2.00 which is the maximum error value of AMDI.

In the similar way, for the results in last row (algo. 3.3) of the unified method on data sets 5, 6, 7 and 8, the column numbers of $\hat{\mathbf{A}}$, \hat{n} , are 10, 24, 11, and 24, while the column numbers of the actual mixing matrix \mathbf{A} , n , are 3, 6, 3, and 6, respectively. Therefore, only 3, 6, 3, and 6 columns of $\hat{\mathbf{A}}$ which are close to the corresponding actual mixing matrices \mathbf{A} are selected from $\hat{\mathbf{A}}$ to compute the AMDI values.

4.4 Experiment 2

The purpose of this experiment is to demonstrate some detail results from the proposed algorithm on real speech data sets. The obtained results also were compared to the results of the standard k-mean method and AICA method. In order to perform the

comparison, the proposed algorithm was tested on data set 5 and data set 6 used in [7] with AICA method (see the details of these two data sets in Section 4.2.2). In both experiments of these two data sets, the parameter τ to find the filtering threshold for inner point removal process was set to 625 that equals to $0.01 * N$, where $N = 62,500$ being the number of sample points of the analyzed data set.

Figure 4.5 shows the scatter plot of two observed signals (a), the remaining data obtained from inner point removal process (b), and, the result from high density direction identification process of data set 5 on the first row and data set 6 on the bottom. As shown in the figures, all directions with high concentrate of data can be detected. All columns of the mixing matrix can be estimated by refining the points which lay in the same direction.

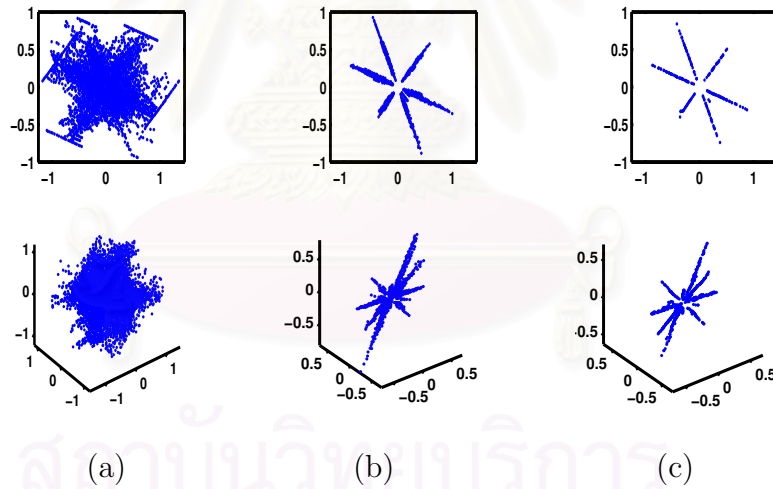


Figure 4.5: Results on data set 5 (top) and data set 6 (bottom). (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process.

After applying the proposed algorithm, the obtained mixing matrices were compared with the results presented in [7] with the AICA method as shown in the first and the second rows of Table 4.6. Moreover, the standard k-mean method was performed to

estimate the mixing matrix \mathbf{A} . From k cluster centers, the sub-matrix $\hat{\mathbf{A}}$ which was close to the mixing matrix \mathbf{A} was given. The difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual matrix \mathbf{A} was, then, represented by AMDI value. Since the convergence of the k-mean method depends on the initialization, the k-mean method was performed in several times for each case. The minimum AMDI value of each case was selected to show in Table 4.6. As seen in the table, the AMDI values obtained by the proposed algorithm are smaller than the results of both the AICA method and the k-mean method.

Table 4.6: The difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual mixing matrix \mathbf{A} on data set 5 and data set 6, AMDI value.

	(Data Set 5) (n=3)	(Data Set 6) (n=6)
The proposed method	7.40E-06	0.0018
The AICA method	< 0.001	0.0103
k-mean method, $k=n$	0.0340	0.0892
k-mean method, $k=2n$	0.0008	0.0149
k-mean method, $k=3n$	0.0015	0.0091

To estimate source signals, the minimum ℓ_1 -norm solution is applied to observed signals \mathbf{X} and the estimated mixing matrix $\hat{\mathbf{A}}$. The signal-to-noise ratio (SNR) reconstruction index in [30] is, then, used to evaluate the estimated sources

$$SNR_i = 10 \log \frac{\|s_i(t)\|^2}{\|\hat{s}_i(t) - s_i(t)\|^2}, i = 1 \dots n. \quad (4.3)$$

where $\|*\|^2$ is the sum of squares over discrete time t and n is the number of sources mixed in the observed signals.

From data set 5, the SNR reconstruction indices of three estimated signals are 13.03, 15.07, and 25.21 dB. For six estimated sources signals of data set 6, they are 10.80, 12.68, 12.47, 9.66, 10.48 and 12.16 dB.

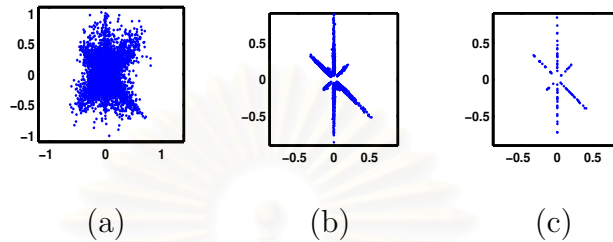


Figure 4.6: Result on data set 7. (a) Observed signals. (b) The remaining data after inner point removal process. (c) The result from high density direction identification process.

Moreover, another test is to compare with the single step approach for overcomplete source separation using Maximum Likelihood (ML) approach [3]. The proposed algorithm was tested on data set 7. The original speech source signals are depicted in Figure 4.6 (as given detail in Section 4.2.2). After estimating the mixing matrix by the proposed algorithm, the results are illustrated in Figure 4.6. The AMDI value representing the difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual matrix \mathbf{A} is $1.92\text{E-}05$. Since it is very small, the estimated mixing matrix is very close to the original mixing matrix \mathbf{A} . Moreover, the number of columns in the estimated mixing matrix is equal to the number of sources, S , mixed in the observed signals, \mathbf{X} . Then, the ℓ_1 -norm method is used to recover the three source signals. The signal-to-noise ratio (SNR) for the separation is 13.09, 20.83, and 22.52 dB, respectively. These results are similar to the results obtained from the ML approach [3]: 15.71, 19.61, and 20.42 dB.

4.5 Experiment 3

This experiment was performed for estimating mixing matrix process in both time domain and time frequency domain on speech data set 8, as demonstrated in Figure 4.4. The results of proposed algorithm were compared to the results gained from the standard k-mean algorithm. After applying the proposed algorithm, the six columns of the estimated mixing matrix was obtained. As shown in Table 4.7, the AMDI value is very small and the estimated number of sources is equal to the actual number of sources. For comparison, after normalizing all column vectors to the unit length, the k-mean clustering algorithm is used to estimate the mixing matrix \mathbf{A} . Again the k-mean method is performed several times and the minimum error case is selected. When the number of clusters is set to six, the high value of AMDI result shows that the standard k-mean method cannot find the satisfactory results. Moreover, the difference choices of $k = 12, 18$ and 24 are set to the number of clusters to perform the k-mean method. The difference between \mathbf{A} and $\hat{\mathbf{A}}$ of each case is presented in the second column of Table 4.7.

Table 4.7: The difference between the estimated mixing matrix and the actual mixing matrix with AMDI value in time domain and Time Frequency domain, AMDI value.

	Time Domain	Time Frequency Domain
The proposed method	2.15E-04	1.42E-04
K-mean method, k=6	0.0954	0.0988
K-mean method, k=12	0.0146	0.0127
K-mean method, k=18	0.0037	0.0074
K-mean method, k=24	0.0019	0.0017

Furthermore, both methods are performed in the time-frequency domain by using

the single-level Daubechies wavelet packets transform. The result is presented in the third column of Table 4.7. In the transformed domain, the error value of the estimated mixing matrix from both methods is also very small. However, the AMDI value is less than 0.001 and also smaller than all the AMDI values obtained by the standard k-mean clustering algorithm.

4.6 Experiment 4

This experiment was tested on speech data sets 5, 6, 7, and 8 (as described in section 4.2.2). The aim of this experiment was to test the performance in the noisy environment. Each data set was tested under 20-dB SNR additive Gaussian noise and was added with five different noise sets. The result of each test is shown in Table 4.8. The second and third columns show the AMDI value and the estimated number of sources in time domain. The last two columns show the results in time-frequency domain by using the single-level Daubechies wavelet packets transform.

From the results in Table 4.8, the difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual matrix \mathbf{A} with AMDI value of each test is very small, except the results on data set 6. However, all AMDI values time-frequency domain are very small. That is the estimated mixing matrix $\hat{\mathbf{A}}$ is close to the actual mixing matrix \mathbf{A} . Moreover, the estimated number of sources, \hat{n} , for each test is equal to the actual number of sources, n .

4.7 Experiment 5

This experiment was tested on data sets 1, 2, 3, and 4 (as described in Section 4.2.1). Figure 4.7 shows the relationship between the percent of sparse vectors in source matrix S , the parameter τ and the average value, \bar{I} , of all the information indices $I'(\mathbf{A}\mathbf{P}_{\mathbf{x}_i})$.

Note that the average value I'_{avg} (see in Section 3.1.1) and the parameter τ is used in the inner point removal process. The four data sets are tested by using the different choices of $\tau = 50, 100, \dots, 500$. The result shows that when using the same size of τ , the value of \bar{I} of data set 3 is the highest because this synthetic data set has the highest fraction of sparse source vectors compared to other data sets. Moreover, the average value \bar{I} corresponds to the size of τ . When the size of τ increases, the number of data points being in the same direction is also increased.

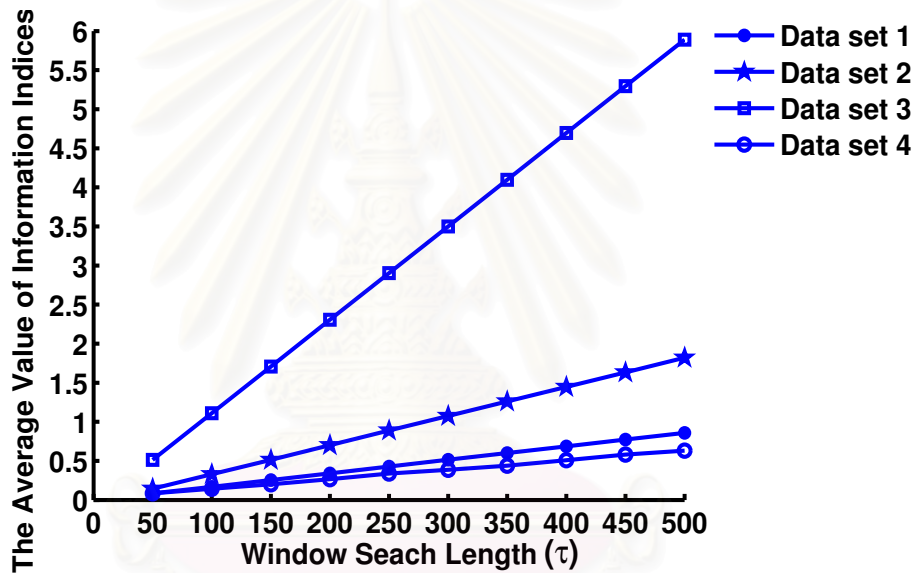


Figure 4.7: The relationship between the percent of sparse source vectors, the size of parameter τ , and the average value of the information index (I'_{avg}).

Consider the data set 4 which has the same percentage of the sparse source vectors as data set 1. Since the number of source signals mixed in the data set 4 is higher, by using the same τ , the value of \bar{I} of data set 4 is very small and less than that of data set 1. This result is given because the data set 4 has the smaller number of sparse vectors. As mentioned in the previous section, if the sources are not sparse enough, the precise mixing matrix is difficult to estimate. However, after the points with small information indices are removed the average value of all information indices, it is found

that I'_{avg} of the remaining points is increased as shown in Figure 4.8. From this result, the average value of information indices, I'_{avg} , depends on the fraction of sparse source vectors. Since the average value of informative, I'_{avg} , increases after applying the inner point removal process, this means that the fraction of the sparse source vectors should be also increased. Then, the mixing matrix can be easily identified.

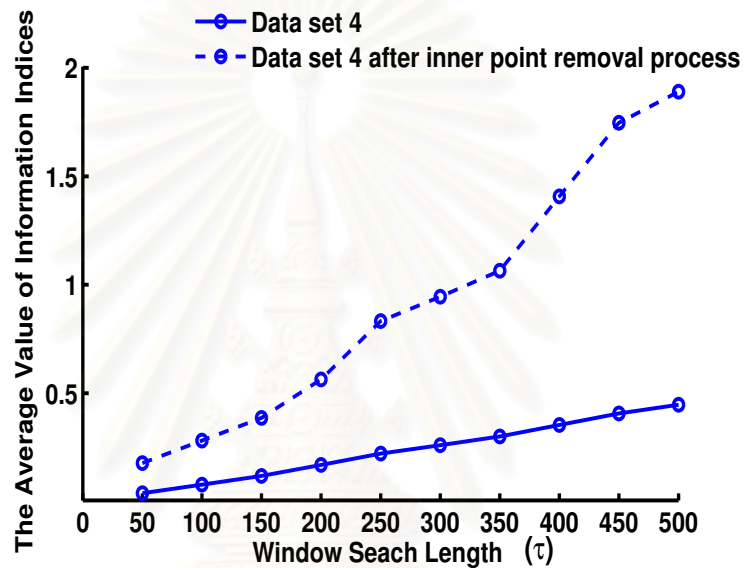


Figure 4.8: The average value of information index before and after the inner point removal process.

After applying the algorithm on data set 4 using $\tau = 50, 100, \dots, 500$ to find the information indices in the inner point removal process. Then, all points containing less information are removed and the remaining points are used to identify the mixing matrix by using our perturbed mean shift method. The column number of the estimated mixing matrix is shown in Figure 4.9(a) for each choice of τ . The algorithm is able to select the result by looking for a longer range of values of τ that yield the same number of columns in the estimated mixing matrix. As seen in Figure 4.9(b), the difference between the estimated mixing matrix $\hat{\mathbf{A}}$ and the actual matrix \mathbf{A} also approaches zeros when the selected choice of τ is greater than or equal 200.

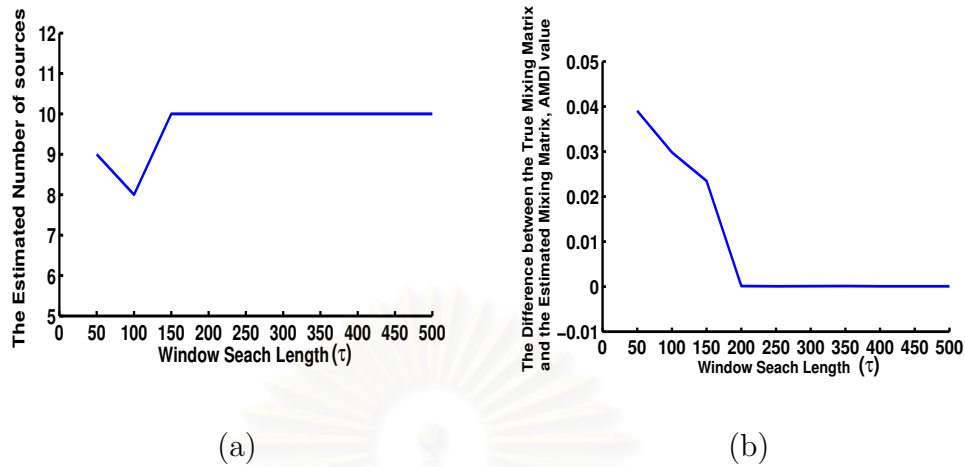


Figure 4.9: By using the different choices of τ for synthetic data set 4. (a) The number of estimated sources. (b) The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value

Moreover, the proposed algorithm was tested on data set 5 to data set 8 by using $\tau = 50, 100, \dots, 500$. Table 4.9 and Table 4.10 show the difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , in time domain and time frequency domain, respectively. From the results on data set 7 and data set 8 in Table 4.9, the proposed algorithm gives the good estimated mixing matrix, $\hat{\mathbf{A}}$ with small AMDI value and the same estimated source number \hat{n} , when the size of window search length τ is increase. However, in time frequency domain, the estimated source number, \hat{n} , of each test is not far from the actual source number n . Moreover, the difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$ with AMDI value of each test is very small. That is the estimated mixing matrix $\hat{\mathbf{A}}$ is very close to the actual mixing matrix \mathbf{A} with small size of τ as shown in Table 4.10.

4.8 Complexity Analysis

Since, this dissertation mainly focuses on the estimating mixing matrix part. The computational complexity is computed only the sum of three phases of the proposed algorithm of estimating the mixing matrix. The computational complexity is described in the following.

1. *Complexity of Phase 1: Inner Point Removal.*

In Phase 1, the algorithm consists of three steps. In the first step, the algorithm uses window search length algorithm to find the set of adjacent points (\mathbf{AP}_x) and also compute the information index of each data point. Assume that the input data contains N points and the window search length (τ) = M . Consequently, the amount of computational load is MN , where $M \ll N$. In the second step, the filtering threshold is computed and then removes the data points containing the information indices less than the threshold. The amount of computational load of this step is N . In the last step, each remaining data point is redefined with its corresponding adjacent points (\mathbf{AP}_x). This step consumes the computational of MN_1 , where N_1 is the number of remaining points and $N_1 < N$. Then, the computational complexity of Phase 1 is $O(MN + N + MN_1) = O(MN)$.

2. *Complexity of Phase 2: High Density Direction Identification.*

In Phase 2, the perturbed mean shift is applied to identify all possible directions of high concentrate of data. At first, a set of sample points or mean shift vectors, \mathbf{SP} , is defined to reduce the computational load. In additions, after the mean shift algorithm converges, each converged sample point is perturbed with data laying in its corresponding direction. The perturbed mean shift procedure is repeated until the convergence is stable. The amount of computational load of the process is $M_1N_1 + M_1N_1$, where M_1 is the number of sample points and $M_1 \ll N_1$.

Therefore, the computational complexity of Phase 2 is $kO(M_1N_1)$, where k is the number of iterations to perform the perturbed mean shift algorithm for stable convergence.

3. Complexity of Phase 3: Information Entropy Estimation.

After the convergence of the proposed algorithm in phase 2, the converged sample points, $\hat{\mathbf{S}}\mathbf{P}$ are taken as the all possible directions of high data density. In this phase, the algorithm is to refine some points in $\hat{\mathbf{S}}\mathbf{P}$ which lie in the same direction. At first, the algorithm clusters the points into k different groups where all points in the same group lie along in the same direction. In this step, the amount of computation is M_1^2 . Because of $M_1 \ll N_1$, the amount of computation in this step is less than M_1N_1 . In the second step, the eigenvalue and information entropy are computed to refine some directions having small member points. This step also uses the amount of computation less than M_1N_1 . As a result, the computational complexity of Phase 3 becomes $O(M_1N_1)$.

Thus, the overall complexity of the proposed algorithm is $O(\text{Phase1} + \text{Phase2} + \text{Phase3}) = O(MN + M_1N_1 + M_1N_1) = O(MN)$.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Table 4.8: Result of each test in noisy environment.

	Time Domain		Time Frequency Domain	
	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	(\hat{n})	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	(\hat{n})
Data set 5	1.10E-05	3	1.71E-05	3
(n=3)	1.98E-05	3	9.81E-06	3
	5.06E-06	3	1.99E-05	3
	6.27E-06	3	3.01E-05	3
	6.66E-06	3	1.29E-05	3
Data set 6	1.12E-02	9	2.96E-04	6
(n=6)	2.49E-02	8	3.82E-04	6
	2.52E-02	6	2.64E-04	6
	2.36E-02	6	5.42E-04	6
	2.47E-02	5	2.35E-04	6
Data set 7	3.20E-06	3	3.98E-04	3
(n=3)	3.78E-06	3	2.39E-04	3
	1.30E-04	4	1.31E-05	3
	1.05E-05	3	2.97E-04	3
	2.69E-06	3	6.02E-04	3
Data set 8	3.75E-04	6	8.29E-04	6
(n=6)	8.26E-04	6	2.58E-04	6
	3.44E-04	7	5.06E-04	6
	3.24E-04	6	1.56E-04	6
	3.86E-04	6	4.26E-04	6

Table 4.9: The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , using $\tau = 50, 100, \dots, 500$ in time domain.

Size of τ	Data set 5 (n=3)		Data set 6 (n=6)		Data set 7 (n=3)		Data set 8 (n=6)	
	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}
50	7.42E-06	5	2.45E-03	8	4.56E-05	4	2.73E-04	6
100	2.44E-05	4	2.10E-02	5	7.57E-06	3	2.66E-04	6
150	7.88E-05	4	1.57E-03	9	4.62E-05	4	3.52E-04	6
200	3.83E-05	4	1.72E-03	8	6.47E-06	3	3.17E-04	6
250	2.48E-05	4	1.70E-03	8	7.68E-05	3	2.80E-04	6
300	5.66E-05	3	1.83E-03	6	8.48E-05	3	2.15E-04	6
350	3.77E-05	4	1.04E-03	7	7.77E-05	3	2.28E-04	6
400	1.60E-05	4	2.13E-03	8	6.91E-05	3	3.11E-04	6
450	1.38E-05	4	1.91E-03	8	6.61E-05	3	2.48E-04	6
500	1.70E-05	3	2.55E-02	6	5.83E-05	3	2.77E-04	6

Table 4.10: The difference between the actual mixing matrix \mathbf{A} and the estimated mixing matrix $\hat{\mathbf{A}}$, AMDI value, and the estimated source number, \hat{n} , using $\tau = 50, 100, \dots, 500$ in time frequency domain.

Size of τ	Data set 5 (n=3)		Data set 6 (n=6)		Data set 7 (n=3)		Data set 8 (n=6)	
	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)	\hat{n}
50	2.34E-06	4	2.23E-04	6	1.06E-04	4	8.87E-05	6
100	4.85E-05	4	1.12E-04	6	1.97E-04	4	1.14E-04	6
150	2.36E-05	4	2.37E-04	6	2.09E-04	3	1.30E-04	6
200	2.70E-05	4	3.78E-04	6	2.19E-04	3	1.64E-04	6
250	1.70E-05	3	1.44E-04	6	3.63E-04	3	1.30E-04	6
300	1.44E-05	3	2.72E-04	6	2.94E-04	3	1.44E-04	6
350	1.13E-05	3	2.17E-04	6	4.57E-04	3	1.96E-04	7
400	1.42E-05	3	1.35E-04	7	4.50E-04	3	1.07E-04	7
450	8.70E-06	3	2.03E-04	6	1.86E-04	3	1.72E-04	7
500	1.01E-05	3	1.77E-04	6	1.24E-04	3	1.83E-04	7

CHAPTER V

CONCLUSION

In this dissertation, an unsupervised learning algorithm for underdetermined blind source separation is proposed. The proposed algorithm for estimating the mixing matrix has been designed into three phases. In Phase 1, the inner points which locate in the sparse region is identified and removed by using window based learning. Each remaining point is also redefined by its corresponding adjacent points. It is found that the result obtained of this phase shows sparser than the original analyzed input data. So, it is easy to find the directions of the maximum data density by using a clustering method. Perturbed mean shift method based on clustering technique is applied to identify all possible directions of high concentrate data. Then, all possible directions are grouped into different groups in which the members of each group lie in the same direction. Finally, the entropy measure is used to obtain the optimal number of sources. The time complexity of the algorithm is $O(MN)$, where N is the number of data points and M is the size of window search length τ , as described in Chapter 4. After the mixing matrix is estimated, a standard linear programming algorithm is used to recover the source matrix.

From the experimental results, the proposed algorithm is able to perform on underdetermined blind source separation problem when the number of sources is unknown. The proposed algorithm can also work on the speech signals under 20-dB SNR additive Gaussian noise. The algorithm provides better results than a standard k-mean, AICA method, an extension of DUET and TIFROM method and a unified method. Moreover,

the estimated number of sources obtained from the proposed algorithm is very close to the actual number of sources. The main contributions of our proposed method can be summarized as follows:

- The proposed algorithm is able to estimate the mixing matrix when the number of sources is unknown. The accuracy of the estimated mixing matrix is very high. The proposed algorithm also gives a better result than a standard k-mean method and AICA method.
- The proposed algorithm also find the number of sources.
- The proposed algorithm can work well even if the percent of sparse source vectors is very small and the indices of the sparse source vectors are not connected This means that the algorithm can be used in a wide range of data or signals.
- The algorithm can estimate the mixing matrix from observed signals under 20-dB SNR additive Gaussian noise. As the results of all experiments, an estimating mixing matrix can be identified with high and the estimated number of sources is very close to the actual number of sources.

In order to check the limitation of the proposed algorithm for the closest direction detection of the mixing matrix, the experiment was performed by using the first two speech source signals of data set 8 (Figure 4.4 (a)). They were linearly mixed with five different mixing matrices $\mathbf{A} \in R^{2 \times 2}$. The two columns of the five mixing matrices are set to have the difference in direction equals to 1, 2, 3, 4, and 5 degrees. Table 5.1 shows the AMDI values for the mixing matrix estimation. Only the results of 3, 4 and 5 degrees are illustrated in the last three columns because the proposed algorithm totally fails to identify the two closest columns of the mixing matrix for 1 and 2 degrees. The first two columns show the parameter values of α used in the inner point removal

process, and α used in the mean shift procedure. The results show that the proposed algorithm can identify the two closest columns of the mixing matrix when the parameter window radius h is set to be $\frac{\pi}{180}$ and the difference in direction of the two columns of the mixing matrix is higher than 3 degrees. The results are displayed in the first and third row of Table 5.1. However, the algorithm can correctly estimate the two closest directions of the mixing matrix for 3 degrees when window radius h is set to $0.5 \frac{\pi}{180}$.

Table 5.1: The results of two closest directions of mixing matrix for 3 4 and 5 degrees.

Distant Threshold (α)	Window Radius (h)	AMDI($\mathbf{A}, \hat{\mathbf{A}}$)		
		3 degree	4 degree	5 degree
$\frac{\pi}{180}$	$\frac{\pi}{180}$	-	4.69E-05	2.76E-05
$\frac{\pi}{180}$	$0.5 \frac{\pi}{180}$	3.12E-06	3.86E-06	1.83E-06
$0.5 \frac{\pi}{180}$	$\frac{\pi}{180}$	-	2.22E-05	2.20E-05
$0.5 \frac{\pi}{180}$	$0.5 \frac{\pi}{180}$	9.38E-07	1.10E-05	4.24E-07

Finally, the future works should be stated as follows:

- Adaptive parameter values should be used in the future experiment to find proper values of each data set.
- The experiment should be repeated with more number of sources in order to increase reliability of experimental result.
- The detail on pre-processing for transforming data to TF domain should be further considered.
- For source recovery step, it is found that the ℓ_1 -norm method dose not give a good separation even a precise mixing matrix is estimated as shown in Experiment 2.

Thus, more researches on source estimation should be studied.

- Applications of ICA problem should be further considered.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

References

- [1] Lewicki, M. S. and Sejnowski, T. J. Learning Overcomplete Representations. Neural Computation. 12(2000):337-365.
- [2] Olshausen, B. A. and Field, D. J. Sparse Coding with an Overcomplete Basis Set: a Strategy Employed by V1?. Vision Research. 37(1997):3311-3325.
- [3] Lee, T., Lewicki, M. S., Girolami, M. and Sejnowski, T. J. Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations. IEEE Signal Processing Letters. 6(1999):87-90.
- [4] Bofill, P. and Zibulevsky, M. Blind Separation of More Sources than Mixtures Using the Sparsity of Their Short Term Fourier Transform. Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation. (2000):87-92.
- [5] Georgiev, P. G., Theis, F. J., and Cichocki, A. Blind Source Separation and Sparse Component Analysis of Overcomplete Mixtures. Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP). 5(2004):493-496.
- [6] Theis, F. J., Puntonet, C. and Lang, E. W. A Histogram-Based Overcomplete ICA Algorithm. Proceedings of the 4th International Conference on Independent Component Analysis and Blind Signal Separation. (2003):1071-1076.
- [7] Waheed, K. and Salem, F. M. Algebraic Independent Component Analysis: an Approach for Separation of Overcomplete Speech mixtures. IEEE International Conference on Neural Networks. 1(2003):775-780.
- [8] Jourjine, A., Rickard, S. and Yilmaz, O. Blind Separation of Disjoint Orthogonal

- Signals: Demixing N Sources from 2 Mixtures. IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP). 5(2000):2985-2988.
- [9] Yilmaz, O. and Rickard, S. Blind Separation of Speech Mixtures via Time-Frequency Masking. IEEE Transaction on Signal Processing. 52,7(2004):1830-1847.
- [10] Abrard, F., Deville, Y. and White, P. From Blind Source Separation to Blind Source Cancellation in the Underdetermined Case: A New Approach Based on Time-Frequency Analysis. Proceedings of the 3rd International Conference on Independent Component Analysis Signal Separation (ICA). (2001):734-739.
- [11] Abrard, F. and Deville, Y. Blind Separation of Dependent Sources using the Time-Frequency Ratio of Mixtures Approach. Proceedings of the 7th International Symposium on Signal Processing Applications (ISSPA). (2003):14.
- [12] Theis, F. J., Georgiev, P. G., and Cichocki, A. Robust Overcomplete Matrix Recovery for Sparse Source using a Generalized Hough Transform. Proceedings of the 12th European Symposium on Artificial Neural Networks (ESANN). (2004):223-232.
- [13] Li, Y., Amari, S., Cichocki, A., Ho, D. W. C. and Xie, S. Underdetermined Blind Source Separation Based on Sparse Representation. IEEE Transactions on Signal Processing. 54,2(2006):423-437.
- [14] Lv, Q. and Zhang X. A unified Method for Blind Separation of Sparse Sources With Unknown Source Number. IEEE Signal Processing Letters. 13,1(2006):49-51.
- [15] Hyvarinen, A., Karhunen, J. and E. Oja. Independent Component Analysis. A Wiley-Interscience Publication. 2002.
- [16] Chen, S. S., Donoho, D. L. and M. A. Saunders. Atomic Decomposition by Basis Pursuit. SIAM Journal of Scientific Computing. 20(1998):33-61.

- [17] Bofill, P. and Zibulevsky, M. Underdetermined Blind Source Separation using Sparse Representations. Signal Processing. 81(2001):2353-2362.
- [18] Li, Y., Cichocki, A. and Amari, S. Sparse Component Analysis for Blind Source Separation with Less Sensors Than Sources. Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation. (2003):89-94.
- [19] Li, Y., Cichocki, A. and Amari, S. Analysis of Sparse Representation and Blind Source Separation. Neural Computation. 16(2004):1-42.
- [20] Theis, F. J. and Lang, E. W. Formalization of Two-Step Approach to Overcomplete BSS. Proceedings of the International Conference on Signal and Image Processing. (2002):207-212.
- [21] Theis, F. J., Jung, A., Puntonet, C. G. and Lang, E. W. Linear geometric ICA: Fundamentals and Algorithms. Neural Computation. 15(2002):1-21.
- [22] Haykin, S. Neural Network a Comprehensive Foundation. 2nd ed. Prentice Hall, 1999.
- [23] Fukunaga, K. and Hostetler, L. D. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Transactions on Information Theory. 21(1975):32-40.
- [24] Cheng, Y. Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 17(1995):790-799.
- [25] Comaniciu, D. and Meer, P. Distribution Free Decomposition of Multivariate Data. Pattern Analysis and Applications. 2(1999):pp22-30.
- [26] Comaniciu, D. and Meer, P. Mean Shift: A Robust Approach Toward Feature

- Space Analysis. IEEE transactions on Pattern Analysis and Machine Intelligence. 24,5(2002):603-619.
- [27] Comaniciu, D. and Meer, P. Robust Analysis of Feature Spaces: Color Image Segmentation. IEEE International Conference on Computer Vision and Pattern Recognition. (1997):750-755.
- [28] Barash, D. and Comaniciu, D. Meanshift Clustering for DNA Microarray Analysis. IEEE International Conference on Computational Systems Bioinformatics (CSB). (2004):578-579.
- [29] Frigui H. and R. Krishnapuram, R. A Robust Algorithm for Automatic Extraction of an Unknown Number of Clusters from Noisy Data. Pattern Recognition Letter. 17(1996):1223-1232.
- [30] Bofill, P. Underdetermined Blind Separation of Delayed Sound Sources in the Frequency Domain. Neurocomputing. 55(2003):627-641.

Biography

Name: Miss Benjamas PANYANGAM.

Date of Birth: 1st October, 1973.

Educations:

- Ph.D. Candidate in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand, (June 2002 - May 2007)
- M.Sc. Program in Computer Science, Faculty of Science, Prince of SongKla University, Thailand (June 1999 - May 2002).
- B.Sc. Program in Computer Science, Faculty of Science, Chiang Mai University, Thailand (June 1992 - May 1996).

Publication papers:

- B. Panyangam, K. Chinnasarn, and C. Lursinsap. Estimating Columns of Under-determined Mixing Matrix by Information Index Removal and Perturbed Mean Shift Algorithm. *Proceedings of the 3rd International Conference on Natural Computation (ICNC)*, Haikou, China, 2007. Accepted and scheduled to appear in August 2007.

Scholarship: The National Science and Technology Development Agency (NSTDA) of Thailand.