วิธีที่ซับซ้อนต่ำสำหรับการแยกสัญญาณเคอร์โทซีสแบบผสมอย่างคงที่โดยไม่รู้แหล่งที่มา

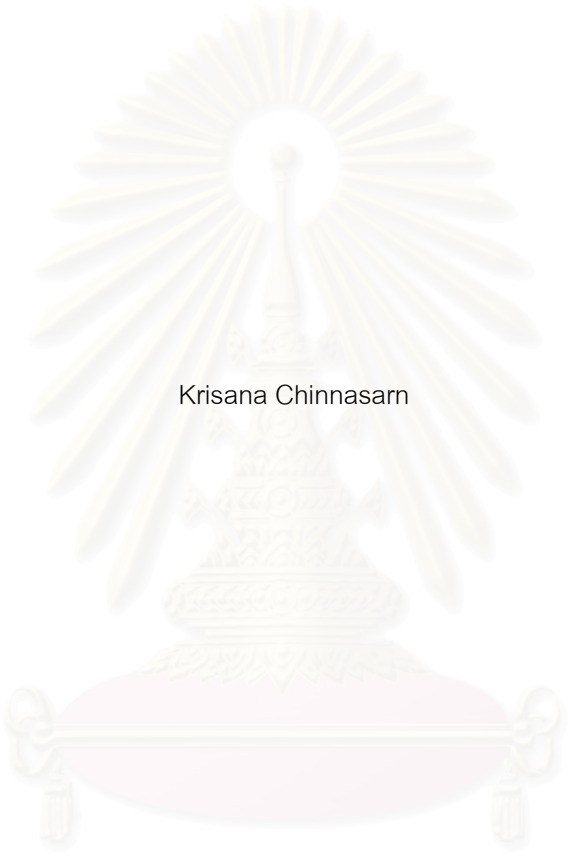
นายกฤษณะ ชินสาร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2546

ISBN 974-17-4056-5

LOW COMPLEXITY METHOD FOR BLIND SOURCE EXTRACTION FOR STATIONARY MIXED KURTOSIS SIGN SIGNALS

Krisana Chinnasarn



A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic year 2003

ISBN 974-17-4056-5

Thesis Title	LOW COMPLEXITY METHOD FOR BLIND SOURCE EXTRACTION FOR STATIONARY MIXED KURTOSIS SIGN SIGNALS
By	Mr. Krisana Chinnasarn
Field of Study	Computer Science
Thesis Advisor	Professor Dr. Chidchanok Lursinsap

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctor's Degree

…….. Dean of Faculty of Science
(Professor Dr. Piamsak Menasveta)

THESIS COMMITTEE

…….. Chairman
(Associate Professor Dr. Jack Asavanant)

…….. Thesis Advisor
(Professor Dr. Chidchanok Lursinsap)

…….. Member
(Associate Professor Dr. Supol Durongwatana)

…….. Member
(Assistant Professor Dr. Boonserm Kijsirikul)

…….. Member
(Dr. Rajalida Lipikorn)

…….. Member
(Associate Professor Dr. Yuttapong Rangsanseri)

บทคัดย่อวิทยานิพนธ์

นายกฤษณะ ชินสาร : วิธีที่ซับซ้อนต่ำสำหรับการแยกสัญญาณเคอร์โทซีสแบบผสมอย่างคงที่โดยไม่รู้แหล่งที่มา. (LOW COMPLEXITY METHOD FOR BLIND SOURCE EXTRACTION FOR STATIONARY MIXED KURTOSIS SIGN SIGNALS) อ. ที่ปรึกษา : ศาสตราจารย์ ดร. ชิดชนก เหลือสินทรัพย์, 77 หน้า. ISBN 974-17-4056-5.

วิทยานิพนธ์นี้นำเสนอขั้นตอนวิธีสำหรับการแยกสัญญาณในเวลาจริงที่ใช้ความซับซ้อนในการทำงานต่ำสำหรับการผสมสัญญาณอย่างคงที่และไม่รู้แหล่งที่มา วิธีการที่นำเสนอจะแบ่งสัญญาณผสมที่รับเข้ามาออกเป็นส่วนย่อย และแยกสัญญาณที่แบ่งออกเป็นส่วนย่อยนั้นด้วยฟังก์ชันการกระตุ้นที่มีความซับซ้อนต่ำโดยใช้เพียงตัวดำเนินการ "shift-and-add" สำหรับการทำงานระดับฮาร์ดแวร์จริง (VLSI level) นอกจากนั้น วิทยานิพนธ์นี้ยังได้นำเสนอวิธีการเลือกค่าเริ่มต้นที่เหมาะสมของเมตริกซ์ของการแยก วิธีการที่นำเสนอนี้ได้ทำการทดสอบกับข้อมูลทดสอบมาตรฐาน ซึ่งจัดเก็บไว้ที่ http://speech.kaist.ac.kr/~jangbal/ch1bss ผลการทดลองพบว่าวิธีการที่นำเสนอมีประสิทธิภาพเทียบเท่ากับวิธีการแก้ปัญหาอื่นแต่มีความซับซ้อนต่ำกว่าทั้งด้านการใช้พื้นที่ในหน่วยความจำและเวลาของหน่วยประมวลผลกลาง

ภาควิชา	**คณิตศาสตร์**	ลายมือชื่อนิสิต.....
สาขาวิชา	**วิทยาการคอมพิวเตอร์**	ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา	**2546**	

4373802423 : MAJOR COMPUTER SCIENCE

KEY WORD: Independent Component Analysis / blind source extraction / blind source separation / Information Maximization / Natural gradient.

KRISANA CHINNASARN: LOW COMPLEXITY METHOD FOR BLIND SOURCE EXTRACTION FOR STATIONARY MIXED KURTOSIS SIGN SIGNALS. THESIS ADVISOR: PROFESSOR CHIDCHANOK LURSINSAP, Ph.D., 77 pp. ISBN 974-17-4056-5.

This dissertation concerns the problem of how to make the extracting algorithm run in real time and how to reduce the computational complexity for the blind source separation. Our approach is to partition the observed signals into several pieces and to extract the partitioned observations with our proposed approximation activation function performing only the "shift-and-add" operation on the VLSI level. No division and exponential multiplication are needed. Moreover, an optimal initial demixing weight for speeding-up the separating time will be presented. The proposed algorithm is tested on the benchmarks available at http://speech.kaist.ac.kr/~jangbal/ch1bss. The experimental results signify that our solution provides a comparable efficiency as those of other approaches but lower in space and time complexity.

Department Mathematics Student's signature.....
Field of study Computer Science Advisor's signature.....
Academic year 2003

Acknowledgements

First of all I would like to thank the Thai Government who sponsors the research scholarships. I am greatful to my supervisor, Professor Dr.Chidchanok Lursinsap, to whom with his advice, guidance and care, help me to overcome the neccessary difficulties of the process of research and make this dissertation possible. I would also like to thank my co-supervisor, Dr.Vasile Palade at Oxford University Computing Laboratory, who gives me a wonderful suggestions in Ph.D. research methodologies.

My thanks also goes to dissertation committee for their advice and guidance in helping me focus on my research activities. And, I would like to thank Assoc.Prof.Suchada Siripant, Khamron Sunat and all my colleagues at the Advanced Virtual Intelligent Computing Center, Department of Mathematics, Chulalongkorn University, who give me a number of useful suggestions.

I would also like to thank all my colleagues at the Department of Computer Science, Burapha University, especially Seree Chinodom, Jira Jaturanon, Tomkanok Chantarujirakorn, Dr.Suwanna Rasmequan, Jarungjit Parnjai, Benchaporn Jantarakongkul, Pusit Kulkasem and John Gatewood Ham for their wormest care support and being patient during my doubtful stage.

Finally, my deepest gratitude goes to Hempolchom's family, Chinnasarn's family, Sukkapun's family and special to my wife, Sirima Chinnasarn, for their sponsor, love and care that inspire this research.

Table of Contents

List of Tables

List of Figures

CHAPTER I

Introduction

1.1 Introduction and Problem Review

Recently, blind source separation (BSS) or independent component analysis (ICA) has become a high potential real world application problem. ICA problem concerns the transformation and de-transformation of source signals in an unknown environment. More precisely, the source distributions $\mathbf{s}(t)$ and the mixture environments $\mathbf{B}$ are assumed to be totally unknown. The main objective of ICA problem is to recover the source signals from the observed signals $\mathbf{x}(t)$, which are collected from the receivers, such as microphones or sensors.

ICA can be seen as an extension to principal component analysis and factor analysis. ICA is a much more powerful technique than PCA, however, capable of finding the underlying factors or sources when these classic methods fail completely. The results of using the ICA technique are not only mutually independent but are also mutually decorrelated. The application of the ICA technique covers several essential areas such as speech separation, steganography, cryptography, data communication, double-talk detection or echo cancellation, sensor signal processing, microarray processing, biomedical source processing, fault diagnosis, feature extraction, financial time series analysis, and data mining [2, 12, 18, 19, 29, 39]. The measurements of the ICA technique are given as a set of sequential or parallel signal separation, time dependency, stationary and non-stationary sources, and linear and nonlinear mixtures.

A very well-known practical example of ICA application is *the cocktail party problem*. This problem assumes there are some people talking simultaneously in the room, having some microphones for receiving the voices. Herein, we assume that there are three people, $n = 3$, and three microphones, $m = 3$. Each of these recorded signals is a linear combination of the mixing matrix $\mathbf{B}$ and the speaker signals $s_i(t)$, $(1 \leq i \leq 3)$, formulated as:

$$\begin{aligned} x_1(t) &= b_{11}s_1(t) + b_{12}s_2(t) + b_{13}s_3(t) \\ x_2(t) &= b_{21}s_1(t) + b_{22}s_2(t) + b_{23}s_3(t) \\ x_3(t) &= b_{31}s_1(t) + b_{32}s_2(t) + b_{33}s_3(t) \end{aligned} \tag{1.1}$$

or more compactly in a matrix form as:

$$\mathbf{x}(t) = \mathbf{B}\mathbf{s}(t) \tag{1.2}$$

Figure 1.1 illustrates a cocktail party problem with n sources, $s_i(t) = [s_1(t), s_2(t), \ldots, s_n(t)]^T$, and k noises, $v_j(t) = [v_1(t), v_2(t), \ldots, v_k(t)]^T$, which are mixed in an unknown environment by the mixing matrix $\mathbf{B}$. Each microphone Mic_i records the time signal, $x_i(t)$ and t is the time index.

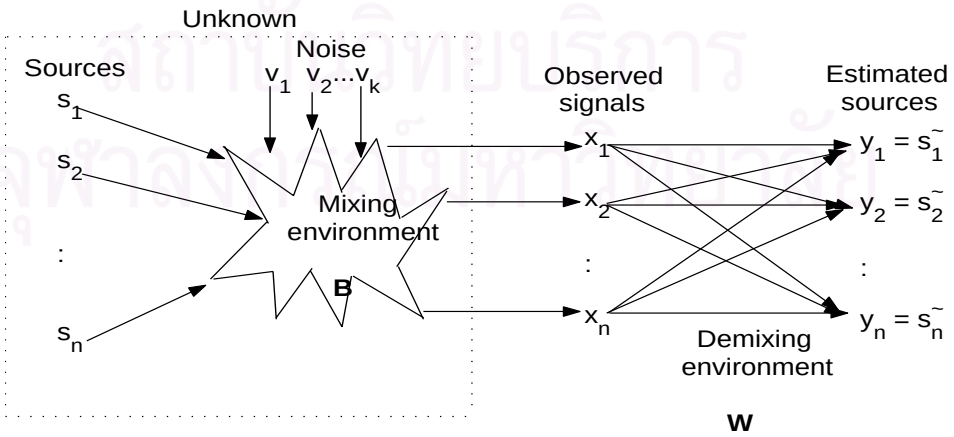


Figure 1.1: The cocktail party problem: an example of ICA structure.

Several solutions to the ICA problem have been published during the past two decades in the fields of signal processing [1, 2, 8, 9, 18, 32, 39, 40, 43], artificial neural networks [4, 11, 12, 13, 14, 17, 23, 26, 27, 33, 46], statistics [9, 19, 29], information theory [2, 3, 7, 8, 9, 11, 26, 38, 39, 45, 47], and other application fields [16, 18, 29]. Two significant points of research for ICA problems which are source density estimation—the probability density function of the sources— and the cost function or contrast function were proposed. The source density estimations are, for examples, Edgeworth expansion [19] and Gram-Charlier expansion [2, 26]. Approximation functions with low complexity computation were presented in consequence [13, 14, 43]. Contrast functions for ICA are based on the concept of information theory including Maximum Likelihood Estimation (ML) [8, 28], Information Maximization (Infomax) [2, 7, 18, 38, 39], and the concept of high order statistics involving the 2^{nd} order statistics [1, 9], 4^{th} order cumulant [5, 19, 30], and Kurtosis [9, 18, 29, 39]. The significant contributions of the ICA problem discussed in this dissertation are:

- increasing learning speed-up,
- separation of mixed Kurtosis signed signals,
- finding some low complexity activation functions for approximating probability density functions for the super-Gaussian and sub-Gaussian channels,
- finding an optimal subblock size for an unsupervised learning.

1.2 Statement of the Problem

In this dissertation, efficient learning methodologies will be proposed in order to find the optimal recovered signals. The problem to be solved in this dissertation can be classified as follows:

First, as described above, the distribution of sources for the BSS problem is totally unknown. For super-Gaussian source, it has a positive Kurtosis sign. Contrastingly, the sub-Gaussian source has a negative Kurtosis sign. But, the Kurtosis sign of source has been changed after the linear transformation $\mathbf{x} = \mathbf{Bs}$. Hence, it is difficult to determine an appropriate activation for each observed channel. Moreover, it is more difficult to do when the sources are mixed between super-Gaussian and sub-Gaussian distributions. In this dissertation, the algorithms for solving these described problems will be proposed.

Second, it has been known that the activation functions for demixing of the super-Gaussian and sub-Gaussian channels are $\phi(y_i) = \tanh(\alpha_i y_i)$ and $\phi(y_i) = y_i^3$, respectively, where $1 \leq i \leq n$. Each output channel y_i is independently evaluated via the activation function $\phi(y_i)$. Two weak points of both functions are listed below:

- They are of high order complexities which require high computational time per instruction.
- They are difficult to realize on the hardware level.

In order to obtain the low computational cost activation function, two approximation activations for separating the super-Gaussian and sub-Gaussian channels will be developed.

Third, in the batch learning method, the computation must be performed on learning data sets which requires the following costly resources:

- amount of computer memory.
- the number of CPU time computation.
- high computational complexity.

The learning methods which reduce the usage of computer memory, CPU time computation, computational complexity, and the online learning systems will be introduced.

1.3 Research Objectives

1. To approximate the low computational time activation functions for demixing super-Gaussian and sub-Gaussian distributions.
2. To propose a new contrast function for evaluating the dependency among output channel y_i and y_j where $1 \leq i, j \leq m$.
3. To propose the suitable online subblock size for reducing the computational complexity.
4. To propose and generalize the learning methodologies for separating the mixed Kurtosis sign sources.

1.4 Scope of the Study

1. The source signals are Gaussian, super-Gaussian and sub-Gaussian distributions.
2. The sources are independently distributed.
3. The number of sources is equal to the number of sensors ($n = m$).
4. The source signals are mixed together in a stationary environment by an unbiased mixing matrix.

1.5 Research Plans

1. Collect the super-Gaussian and sub-Gaussian sources from the standard and existing benchmark databases.
2. Study various proposed methods in the blind source separation researches.
3. Study the principal theories and various researches in neural networks and statistical learning techniques to analyze the data from steps 1 and 2.

4. Apply the neural network and statistical learning techniques from step 3 to the collected data above.

5. Design an appropriate algorithms from the study results and perform the experiments to validate the algorithms.

6. Conclude the experimental results by comparing the results with those from other methods.

1.6 Research Advantages

It is expected that the designed approach are:

1. applicable for separation of any super-Gaussian and sub-Gaussian signals.

2. used for a preprocessing procedure of other signal applications such as signal recognition.

CHAPTER II

Theories and Literature Reviews

This chapter, the basic concepts of probability theory, statistics, random processes, independent component analysis or blind source separation model, ICA learning methods, and some literatures on preprocessing, contrast functions, and activation functions are briefly revised.

2.1 Probability of Events

Let $A, B, C, \cdots, N$ denote events. The probability of each event is a real number between 0 and 1 denoted by $P[A], P[B], P[C], \cdots, P[N]$. The notation of $P[.]$ with square brackets will always be used to denote the probability of the event. If an event is certain to occur, the probability of the event is 1. On the other hand, the probability of the null event is 0. If events A and B are complementary, then their probabilities must add to 1 as related in equation (2.1).

$$P[B] = 1 - P[A] \tag{2.1}$$

The joint probability of events A and B, denoted by either $P[AB]$ or $P[A\ and\ B]$ is the probability that both events A and B occur simultaneously.

$$P[A\ and\ B] = P(A \cap B) \tag{2.2}$$

2.2 Probability Distribution and Density Functions

2.2.1 Distribution of a Random Vector

In this dissertation, we assume the random variables are continuous-valued unless stated otherwise. Let $\mathbf{x}$ be a random vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.3}$$

where the components $x_1, x_2, x_3, \ldots, x_n$ of $\mathbf{x}$ are random variables. Let $\hat{\mathbf{x}}$ be a particular instance of $\mathbf{x}$

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}. \tag{2.4}$$

Compoments $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ are fixed real values. The probability of event

$$\mathbf{x} \leq \hat{\mathbf{x}} : x_1 \leq \hat{x}_1, x_2 \leq \hat{x}_2, \ldots, x_n \leq \hat{x}_n$$

is obviously a function of $\hat{\mathbf{x}}$. This function is called the *cumulative distribution function* (cdf) for the random vector $\mathbf{x}$. The cdf F of variable x_i at point $\hat{x}_i$ is defined as the probability that $x_i \leq \hat{x}_i$:

$$F_{\mathbf{x}}(\hat{\mathbf{x}}) \equiv P_{x_1,x_2,\ldots,x_n}(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \equiv P(\mathbf{x} \leq \hat{\mathbf{x}}) \tag{2.5}$$

where $-\infty \leq \hat{\mathbf{x}} \leq \infty$. Clearly, for continuous random variables, the cdf value is in the interval $0 \leq F_{\mathbf{x}}(\hat{\mathbf{x}}) \leq 1$. Normally, the probability function is described in terms of its density function rather than cdf. The *probability density function (pdf)* $p_{\mathbf{x}}(\hat{\mathbf{x}})$ of the

random variable x is acquired as the derivative of the distribution function with respect to all of the vector components. The multivariate *probability density functions* $p_{\mathbf{x}}(\hat{\mathbf{x}})$ is defined as the derivative of the cdf $F_{\mathbf{x}}(\hat{\mathbf{x}})$ with respect to all components of the random vector $\mathbf{x}$:

$$p_{\mathbf{x}}(\hat{\mathbf{x}}) = \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}\cdots\frac{\partial}{\partial x_n}F_{\mathbf{x}}(\hat{\mathbf{x}}) \tag{2.6}$$

2.2.2 Joint Distribution and Density Functions

If $\mathbf{x}$ and $\mathbf{y}$ are both random vectors (perhaps of different dimensionalitv), the vectors $\mathbf{x}$ and $\mathbf{y}$ can be concatenated to form a "supervector" $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{y}^T)$. The cdf for the supervector is called the *joint distribution function* of $\mathbf{x}$ and $\mathbf{y}$. The cdf can be described as follows:

$$F_{\mathbf{x},\mathbf{y}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = P(\mathbf{x} \leq \hat{\mathbf{x}}, \mathbf{y} \leq \hat{\mathbf{y}}) \tag{2.7}$$

The *joint density function* $p_{\mathbf{x},\mathbf{y}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of $\mathbf{x}$ and $\mathbf{y}$ is again defined by differentiating the joint distribution function $F_{\mathbf{x},\mathbf{y}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ with respect to all components of the random vectors $\mathbf{x}$ and $\mathbf{y}$. Hence, the relationship can be described as follows:

$$F_{\mathbf{x},\mathbf{y}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \int_{-\infty}^{\hat{\mathbf{x}}}\int_{-\infty}^{\hat{\mathbf{y}}} p_{\mathbf{x},\mathbf{y}}(\xi, \sigma)d\sigma d\xi \tag{2.8}$$

2.3 Independence of Signals

To obtain completely separated any two signals y_i and y_j, the values of y_i and y_j must be statistically independent at all times. There are various statistical independence tests that can be used in this context. Two random variables y_i and y_j are said to be statistically independent if the value of y_i does not affect the value of y_j, and vice versa [29]. The independence of sources can be considered in terms of probability density function. We denote by $p(y_i, y_j)$ the joint probability density function of y_i and y_j, and

$p_i(y_i)$ the marginal probability density function of y_i as follows:

$$\begin{aligned} p_i(y_i) &= \int_0^\infty p(y_i, y_j) dy_j \\ \text{and} & \\ p(y_i, y_j) &= p_i(y_i) p_j(y_j) \end{aligned} \tag{2.9}$$

Practically, it is not easy to test whether two signals y_i and y_j are independent by using $p(y_i, y_j)$, $p_i(y_i)$, and $p_j(y_j)$. The easier testing is by considering their correlation. Two random variables y_i and y_j are said to be uncorrelated if their covariance is zero. The covariance can be computed in terms of the correlated expected values and the multiplication of the expected values of y_i and y_j as follows:

$$E[y_i y_j] - E[y_i]E[y_j] = 0 \tag{2.10}$$

If the variables are independent, they are also uncorrelated. On the other hand, uncorrelatedness does not imply independence.

2.4 Independent Component Analysis

A very well-known practical example of an ICA application is *the cocktail party problem*. This problem assumes there are people talking simultaneously in a room, which is provided with some microphones for receiving what they are talking about. Herein, we assume that there are n people, and m microphones as illustrated in Figure 2.1 (for $n = m$). Each microphone Mic_j gives a recorded time signal, denoted as $x_j(k)$, where $1 \leq j \leq m$ and t is an index of time. Each of these recorded signals is a linear combination of the original signals $s_i(k)$, $(1 \leq i \leq n)$ using the mixing matrix $\mathbf{B}$, as given below:

$$x_j(k) = \sum_{i=1}^{n} b_{ji} s_i(k) \tag{2.11}$$

where b_{ji}, $1 \leq j, i \leq n$ are the weighted sum parameters, that depend on the distance between the microphones and the speakers [29]. If the sources s_i are near the receivers Mic_j or the elements of the mixing matrix $\mathbf{B}$ are *diagonal*, it is not a proper ICA problem. Commonly, the elements of mixing matrix $\mathbf{B}$ are nonsingular diagonal and permuted. In addition, the probability density function of $s_i(k)$ are unknown in advance. The only basic assumption of *the cocktail party problem* is that all of the sources $s_i(k)$ are identically and independently distributed (*iid*). The sources, observed and recovered signals, have zero mean, $E[\mathbf{s}] = E[\mathbf{x}] = E[\mathbf{y}] = 0$. The basic background of ICA were presented in [2, 18, 29].

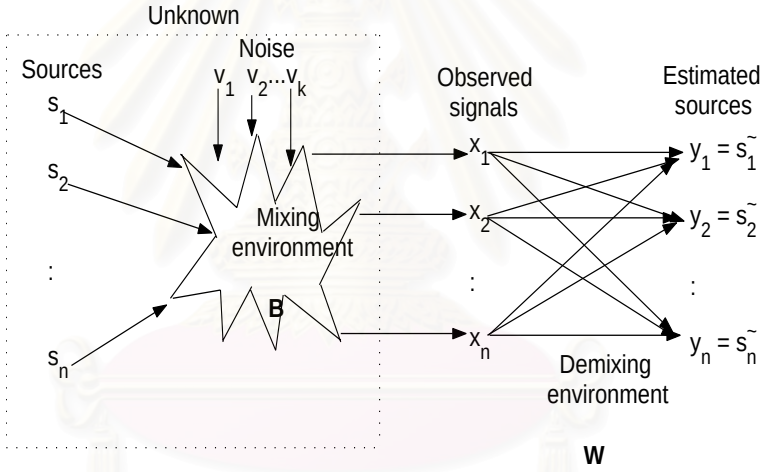


Figure 2.1: The cocktail party problem.

The objective of an ICA problem is to recover the source signals $\tilde{\mathbf{s}} = \mathbf{y} = \mathbf{W}\mathbf{x}$ from an observed signal $\mathbf{x} = \mathbf{B}\mathbf{s}$, where each component of the recovered signals $y_i(k)$ is *iid*. The equation for transforming the mixed signals is the following:

$$\tilde{\mathbf{s}} = \mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{B}\mathbf{s} = \mathbf{B}^{-1}\mathbf{B}\mathbf{s} = \mathbf{I}\mathbf{s} = \mathbf{s} \tag{2.12}$$

Equation (2.12) shows that the full rank demixing matrix $\mathbf{W}$ is needed for recovering the mixed signal x_i.

2.5 Principal Component Analysis

The brief concept of PCA and the notations used in this dissertation are given in this section. Principal Component Analysis, PCA in short, or Karhunen-Loeve transform in data communication is a standard technique used to approximate the source data with lower dimensional pattern vectors. In statistical pattern recognition problem, feature selection or feature extraction is a common task to do first. The minor components that have no effect in learning process will be eliminated. In other word, feature selection is a mapping process from data space to feature space, $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ where $m < n$ or we simply truncate the dimension of observed signal set $\mathbf{x}(t) = [x_1(t)\ x_2(t)\ \ldots\ x_n(t)]^T$ to $\hat{\mathbf{x}}(t) = [\hat{x}_1(t)\ \hat{x}_2(t)\ \ldots\ \hat{x}_m(t)]$, $m < n$, during the sampling period time $1 \leq t \leq P$.

The basic properties of the PCA are defined under the following conditions. Let $\mathbf{x} = \{\mathbf{x}(t)|1 \leq t \leq P\}$ denote a set of input signals during a sampling period P. The definition of each $\mathbf{x}(t)$ is the same as that given in the previous section. The dimension of $\mathbf{x}(t)$ is equal to n. The mean of $\mathbf{x}$ is constrained by

$$E[\mathbf{x}] = 0 \tag{2.13}$$

while its variance is limited to

$$\sigma^2 = \mathbf{q}^T R \mathbf{q} \tag{2.14}$$

where $\mathbf{q}$ denotes an n-dimensional unit vector and $\mathbf{R}$ denotes a correlation matrix of random variables $\mathbf{x}$.

$$\mathbf{R} = E[\mathbf{x}\mathbf{x}^T] \tag{2.15}$$

Dimensionality reduction is a process of eliminating the number of features needed for data representation. The small variance features in data space are eliminated and retained only those terms that have large variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2$ where $m \leq n$. In practice, we reduce some dimensions of $\mathbf{x}(t)$ by considering the eigenvalues of the correlation matrix R. We select $\lambda_1, \lambda_2, \ldots, \lambda_m$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$.

Figure 2.2 illustrates an unsupervised multilayer neural network which consists of PCA and ICA layers. After the PCA layer, the signal dimensions will be reduced from n to m. The m dimensions are the principal components. The remaining dimensions $n-m$ are the minor components or noises. PCA not only reduces the signal dimensions, but also decorrelates the signals.

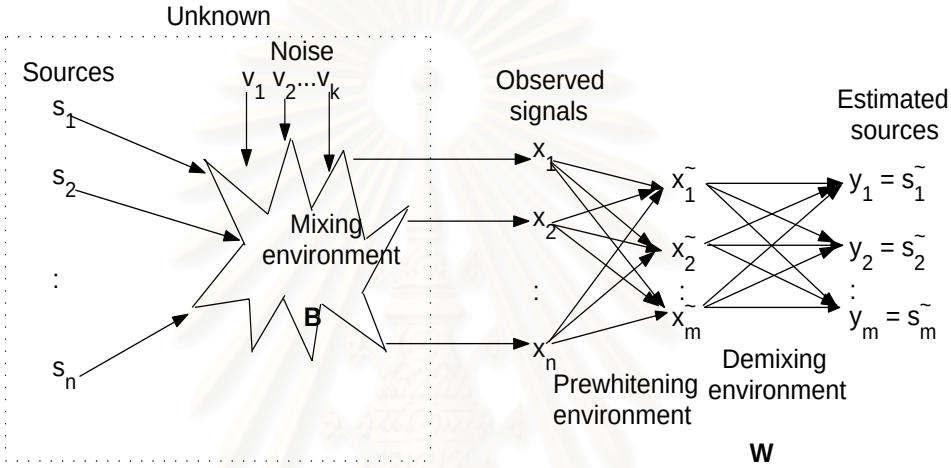


Figure 2.2: PCA works as a preprocessing of an ICA, $m < n$

2.6 Maximization Mutual Information Learning Algorithm

The main objective of the Information Maximization principle is to maximize an output entropy $\mathbf{y}$. Bell and Sejnowski [7] proposed a simple learning algorithm for blindly demixing linear mixtures $\mathbf{x}$ of independent sources $\mathbf{s}$ using information maximization. They showed that maximizing the joint entropy $H(\mathbf{y})$ of the output can approximately minimize the mutual information between the output components $y_i = \phi(y_i)$ where $\phi(y_i)$ is an invertible nonlinearity and $\mathbf{y} = \mathbf{W}\mathbf{x}$. Mutual information at the output neural node or processor is defined as follows:

$$I(y_1, \ldots, y_n) = H(y_1, \ldots, y_n) - H(y_1) + \ldots + H(y_n) \tag{2.16}$$

where $H(y_i)$ are the marginal entropies of the outputs, $H(y_1, \ldots, y_n)$ is the joint entropy of the output $\mathbf{y}$. Minimizing $I(y_1, \ldots, y_n)$ consists of maximizing the joint entropy and the marginal entropies. For $I(y_1, \ldots, y_n) = 0$, the joint entropy is equal to the sum of marginal entropies:

$$H(y_1, \ldots, y_n) = H(y_1) + \ldots + H(y_n) \tag{2.17}$$

As known, the original frame work of ICA is to estimate $\tilde{\mathbf{s}}(t) = \mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ of the sources $\mathbf{s}(t)$. In order to obtain a good estimate, we introduce an objective or loss function in terms of estimates $\mathbf{y}$ and $\mathbf{W}$.

$$L(\mathbf{W}) = E\{\rho(\mathbf{y}, \mathbf{W})\} \tag{2.18}$$

The loss function should be minimized when the component y_i become independent, that is, when $\mathbf{W}$ is a rescaled permutation of $\mathbf{B}^{-1}$. To minimize the dependency among the estimated signals y_i, the Kullback-Leibler divergence between the joint and estimated probability density functions of $\mathbf{y}(t)$ is used. Let $p(\mathbf{y})$ be the joint probability density function of $\mathbf{y}$, and $q(\mathbf{y})$ be an estimated probability density function of $\mathbf{y}$. $q(\mathbf{y})$, sometimes, is called the marginal pdf of $\mathbf{y}$. In which all y_i are statistically independent, $q(\mathbf{y})$ can be rewritten as follows:

$$q(\mathbf{y}) = \prod_{i=1}^{n} q_i(y_i) \tag{2.19}$$

We use the Kullback-Leibler divergence between the joint and estimated probability density functions of $\mathbf{y}$ as follows:

$$D_{pq} = \int_{-\infty}^{\infty} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} \tag{2.20}$$

The Kullback-Leibler divergence always takes a positive value and becomes zero if $p(\mathbf{y})$ and $q(\mathbf{y})$ are the same distribution. It is invariant with respect to invertible nonlinear transformations of variable y_i, including scaling and permutation [18, 26].

Amari [2] showed that Kullback-Liebler divergence $D(\mathbf{W})$ can be calculated from the average Mutual Information (MI) of y_i as follows:

$$\begin{aligned} D_{pq} &= \int_{-\infty}^{\infty} p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} - \int_{-\infty}^{\infty} p(\mathbf{y}) \log q(\mathbf{y}) d\mathbf{y} \\ &= \int_{-\infty}^{\infty} p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} - \sum_{i=1}^{n} \int_{-\infty}^{\infty} p(\mathbf{y}) \log q_i(y_i) d\mathbf{y} \\ &= -h(\mathbf{y}) + \sum_{i=1}^{n} h_i(y_i) \end{aligned} \tag{2.21}$$

where

$$\begin{aligned} h(\mathbf{y}) &= E[\log p(\mathbf{y})] &&= -\textstyle\int_{-\infty}^{\infty} p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} \\ h_i(y_i) &= E[\log q_i(y_i)] &&= -\textstyle\int_{-\infty}^{\infty} p(\mathbf{y}) \log q_i(y_i) d\mathbf{y} \end{aligned} \tag{2.22}$$

$h(\mathbf{y})$ is a differential entropy and $h(y_i)$ is a marginal entropy of variable $\mathbf{y}$, respectively. From $\mathbf{y} = \mathbf{W}\mathbf{x}$, the differential entropy can be calculated by

$$h(\mathbf{y}) = h(\mathbf{W}\mathbf{x}) = h(\mathbf{x}) + \log |\det(\mathbf{W})| \tag{2.23}$$

Applying (2.23) to (2.21), we get

$$D_{pq} = -h(\mathbf{x}) - \log |\det(\mathbf{W})| + \sum_{i=1}^{n} h_i(y_i) \tag{2.24}$$

To find $\mathbf{W}$ that minimizes $D_{pq}(\mathbf{W})$, we differentiate $D_{pq}(\mathbf{W})$ with respect to $\mathbf{W}$. The gradient directions can be derived as follows:

$$\begin{aligned} \frac{\partial D_{pq}(\mathbf{W})}{\partial \mathbf{W}} &= -\frac{\partial \log \left|\det(\mathbf{W}^T)\right|}{\partial \mathbf{W}} + \frac{\partial (\sum_{i=1}^{n} h_i(y_i))^T}{\partial \mathbf{W}} \\ &= -\mathbf{W}^{-T} + \frac{\partial (\sum_{i=1}^{n} \log q_i(y_i))^T}{\partial \mathbf{W}} \\ &= -\mathbf{W}^{-T} + \frac{d(\log q_i(y_i))^T}{dy_i} \cdot \frac{dy_i^T}{d\mathbf{W}} \\ &= -\mathbf{W}^{-T} + \frac{\frac{\partial q_i(y_i)}{\partial y}}{q_i(y_i)} x^T \\ &= -\mathbf{W}^{-T} + \frac{\acute{q}_i(y_i)}{q_i(y_i)} x^T \end{aligned} \tag{2.25}$$

In 1996, Amari *et al.* [2] reported the *natural gradient* learning for the ICA problem, where $\mathbf{W}^T\mathbf{W}$ is an an optimal rescaling of the entropy gradient. Hence, the gradient directions of $\frac{\partial D_{pq}(\mathbf{W})}{\partial \mathbf{W}}$ should be

$$\begin{aligned}\frac{\partial D_{pq}(\mathbf{W})}{\partial \mathbf{W}}\mathbf{W}^T\mathbf{W} &= [-\mathbf{W}^{-T} + \frac{\acute{q}_i(y_i)}{q_i(y_i)}x^T]\mathbf{W}^T\mathbf{W} \\ &= [-\mathbf{I} + \frac{\acute{q}_i(y_i)}{q_i(y_i)}y^T]\mathbf{W} \end{aligned} \tag{2.26}$$

By ordinary *steepest gradient descent* online learning, the learning equation is given by:

$$\begin{aligned}\Delta\mathbf{W}(t) &= \mathbf{W}(t+1) - \mathbf{W}(t) \\ &= -\eta(t)\frac{\partial D(\mathbf{W})}{\partial \mathbf{W}} \end{aligned} \tag{2.27}$$

where η is the learning rate which depends on the learning time t. Hence, $\mathbf{W}$ at time $t+1$ is adjusted by the following constructive steps:

$$\begin{aligned}\mathbf{W}(t+1) &= \mathbf{W}(t) + \eta(t)[I - \frac{\acute{q}_i(y_i)}{q_i(y_i)}y_i(t)^T]\mathbf{W}(t) \\ &= \mathbf{W}(t) + \eta(t)[\mathbf{I} - \phi(\mathbf{y})\mathbf{y}(t)^T]\mathbf{W}(t) \end{aligned} \tag{2.28}$$

The function $\phi(y_i)$ is the nonlinear activation function which depends on the probability density function of the source signals s_i, η is the learning rate, $\mathbf{I}$ is an identity matrix, and t is the time index. It has been known that the activation functions for demixing super-Gaussian and sub-Gaussian channels are $\phi(y_i) = \tanh(\alpha_i y_i)$ and $\phi(y_i) = y_i^3$, respectively, where $1 \leq i \leq n$ [18].

2.7 Kurtosis Measurements

As known, the ICA learning is a blind separating procedure for the non-Gaussian channels. In the ICA mixtures, at most one Gaussian channel is allowed. Because of the Gaussian distribution property, the transformation of two Gaussianities are also Gaussianity with another variable [29]. The non-Gaussianity can be categorized into super-Gaussian and sub-Gaussian distributions. Super-Gaussianity has a sharp peak and a

large tail probability density function (pdf). On the other hand, sub-Gaussianity has a flat pdf. As we described in the previous section, the nonlinear activation function $\phi(y)$ in equation (2.28) is determined by the degree of non-Gaussianity. Hence, we need some measurement tools for determining the degree of non-Gaussianity of random variable s_i. In this dissertation, the *Kurtosis* [29] is used for selecting the nonlinear activation function, which is an appropriate measure for the degree of non-Gaussianity.

$$Kurtosis(s_i) = \frac{E[s_i^4]}{(E[s_i^2])^2} - 3 \tag{2.29}$$

where

$$Kurtosis(s) \begin{cases} < 0, & (\text{if } s_i \text{ is a sub-Gaussianity.}) \\ = 0, & (\text{if } s_i \text{ is a Gaussianity.}) \\ > 0, & (\text{if } s_i \text{ is a super-Gaussianity.}) \end{cases} \tag{2.30}$$

Figure 2.3 shows the family of Gaussian distributions. The thick line is super-Gaussian distribution. The dotted-and-dashed line is Gaussian distribution. The dotted line is sub-Gaussian distribution. Super-Gaussianity has a sharp peak and a long tail distribution. On the other hand, sub-Gaussianity has a flat peak and a short tail distribution. For whitened data $E[s_i^2] = 1$, its Kurtosis is reduced to

$$Kurtosis(s_i) = E[s_i^4] - 3 \tag{2.31}$$

Kurtosis has some useful properties as follows:

- Additivity, if x and y are two statistically independent random variables, then

$$Kurtosis(x + y) = Kurtosis(x) + Kurtosis(y)$$

- Scalar Productivity, for any scalar parameter α,

$$Kurtosis(\alpha x) = \alpha^4 Kurtosis(x)$$

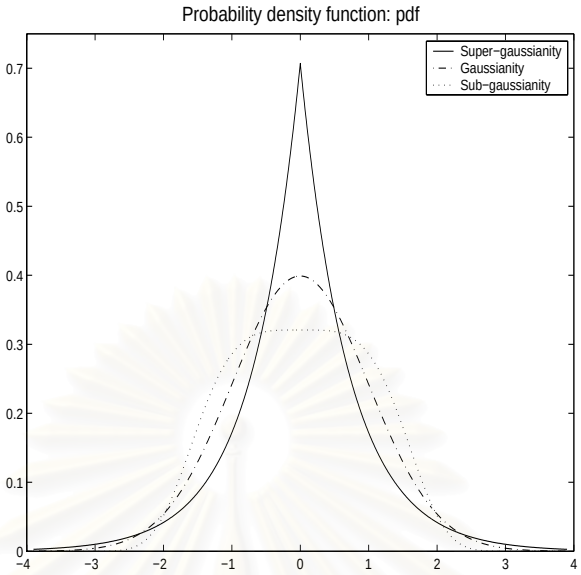


Figure 2.3: Gaussian family.

2.8 Blind Source Extraction

In the mixing and filtering of the blind source separation problem, unknown input sources $s_j(t)$ $(1 \leq j \leq n)$ may have different mathematical or physical models, which depend on the nature of applications. For example, in this dissertation, input sources are classified into positive, negative, and zero Kurtosis values. There are two main approaches to separate the mixtures. The first approach is to separate all sources simultaneously. In the second approach, the sources are sequentially extracted one by one or group by group [18]. The Kurtosis signs of the prewhitened signals as a cost function to extract signals into two subgroups are used in our proposed solution.

2.9 Literature Reviews

Comon [19] proposed *Independent Component Analysis: A new Concept?* and approximated the source distribution by Edgeworth expansion of the mutual information which consist of cumulants of increasing orders. Computational time of the ICA of a data

matrix is within a polynomial time.

Amari *et al.* [2] proposed a novel learning algorithm for blind signal separation. The proposed algorithm minimizes a statistical dependency amoung the outputs. In the simulation, the number of sources are known but the source signals and the mixing matrix are unknown. They used Gram-Charlier expansion for estimating the probability density function of the sources. They derived an efficient learning rule which minimizes the Mutual Information of the outputs using the natural gradiant descent. Finally, they obtained a polynomial activation function $\phi(y)$ of the 11^{th} order for demixing the sub-Gaussianity. The acivation function is rewritten below:

$$f(\mathbf{y}) = \frac{3}{4}y^{11} + \frac{25}{4}y^{9} - \frac{14}{3}y^{7} - \frac{47}{4}y^{5} + \frac{29}{4}y^{3} \tag{2.32}$$

Douglas *et al.* [21] presented two nonlinear activation functions for switching between sub-Gassianity and super-Gaussianity as follows:

$$\phi_{sub}(\mathbf{y}) = \mathbf{y}^3 \quad \text{and} \quad \phi_{sup}(\mathbf{y}) = \tanh(10\mathbf{y}) \tag{2.33}$$

Douglas and Cichocki [22] proposed *Neural Networks for Blind Decorrelation of Signals*. They analyzed and extended a class of adaptive neural networks for second order blind decorrelation of instantaneous signal mixtures. They used a locally-adaptive multilayer decorrelation networks. Their simulations confirmed and pointed out the usefulness of the locally-adaptive networks for decorrelating signals in both space and time.

Karhunen *et al.* [33] presented *A Class of Neural Networks for Independent Component Analysis*. They used an extension of principal component analysis for developing an ICA learning algorithm. They proposed a multilayer feedforward neural network for performing complete ICA. The proposed neural network provides good results from the test examples for both artificial and real-world data. In 1998, Karhunen *et al.* [34] proposed *The nonlinear PCA criteria in blind source separation: Relation with other*

approaches. In this paper, they derived the nonlinear principal component analysis in blind source separation appropriate for comparison with the other BSS or Independent Component Analysis. The choice of the optimal nonlinearity was explained.

Chen, Amari and Lin [11] proposed *a unified algorithm for principal and minor components extraction* using eigenvectors. This algorithm can extract true principal components and true minor components. The proposed algorithm is of practical significance in neural network implementation. The algorithm is based on natural gradient ascent/descent methods (a ponential flow in a Riemannian space).

Cichocki, Douglas and Amari [16] proposed *Robust techniques for independent component analysis (ICA) with noisy data*. A recurrent dynamic neural network is introduced for simultaneous unbiased estimation of unknown mixing matrix, blind source separation and noise reduction in the extracted output signal. The shape parameters of the nonlinearities are adjusted using gradient-based rules.

D.Charles [10] described *Constrained PCA techniques for the identification of common factor data*. An unsupervised learning network is presented that operates similarly to Principal Factor Analysis. The network responds to the covariance of the input data.

Mansour and Jutten [42] used higher order statistics for solving the problem of blind source separation. It was proved that the forth-order cross-cumulant is the simplest criteria for separating the sources when the two sources have the same Kurtosis sign. If not, they required a decorrelation as preprocessing.

Zarzoso *et al.* [59] proposed forth-order statistics estimator for blind source separation. Proposed estimator works well when the sources Kurtosis sum is zero. Heuristic decision rule is used for choosing between the proposed estimator and an other estimator.

Bell and Sejnowski [7] derived a new self-organising learning algorithm which maximises the information transferred in a network of non-linear units. The algorithm did not assume any knowledge of the input distributions. Successfully separating unknown

mixtures is up to ten speakers. And they derived dependencies of information transfer on time delays.

Te-Won Lee *et al.* [38] presented *A Unifying Information-theoretic Framework for Independent Component Analysis.* They showed that different theories recently proposed for Independent Component Analysis (ICA) lead to the same iterative learning algorithm for blind separation of mixed independent sources. They also reviewed those theories and suggested that information theory could be used to unify several lines of research.

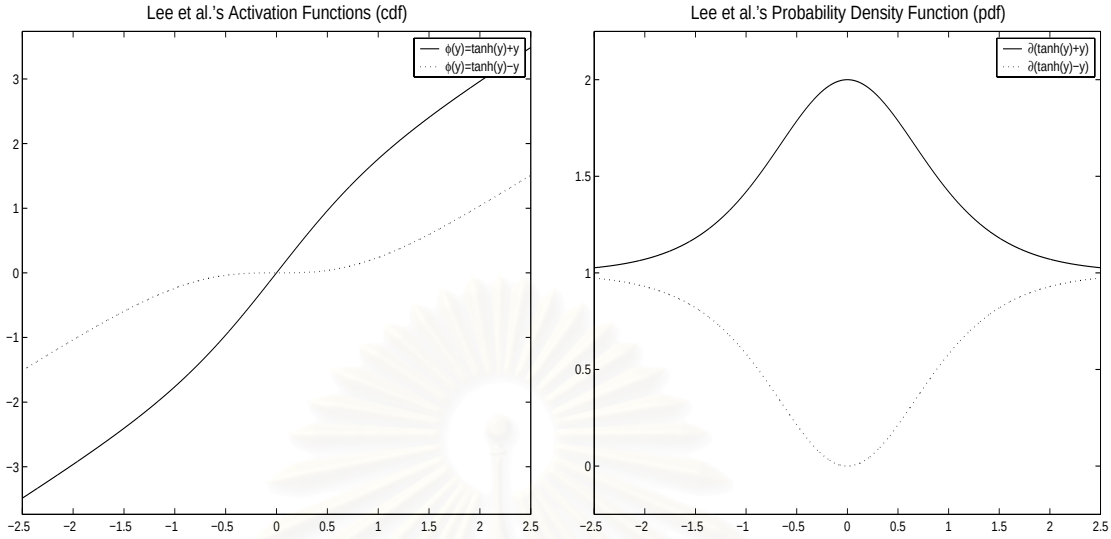
In 1999, Te-Won Lee *et al.* [39] presented an extension of the Information Maximization algorithm of Bell and Sejnowski [7]. Proposed extended infomax is able to blindly separate mixed signal with sub-Gaussian and super-Gaussian source distributions. They used a simple learning rule which was proposed in Girolami's Ph.D. thesis. Bell and Sejnowski [7] learning rule is optimized by natural gradient as in [2, 3]. They demonstated that the extended infomax algorithm is able to separate 20 sources with varity of source distributions. They suggested $\phi(\mathbf{y}) = \tanh(\mathbf{y}) - \mathbf{y}$ for sub-Gaussian distribution, and $\phi(\mathbf{y}) = \tanh(\mathbf{y}) + \mathbf{y}$ for super-Gaussian distribution, respectively. A simple learning equation for separating the mixture of non-Gaussian distribution is expressed as follows:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \eta(\mathbf{I} - \mathbf{K}\tanh(\mathbf{y})\mathbf{y}^T - \mathbf{y}\mathbf{y}^T)\mathbf{W}_t \tag{2.34}$$

where $K = diag[k_1, \ldots, k_m]^T$ is a diagonal matrix of signs. If we know the distribution of sources, then we can assign negative values for k_i if the sources are sub-Gaussian distributed, and positive values if the sources are super-Gaussian distributed, respectively. If the distribution of sources is unknown, the switching between the sub-Gaussian and super-Gaussian learning rule is given by the following:

$$k_i \;=\; sign(E[\mathrm{sech}^2]E[y_i^2] - E[\tanh(y_i)y_i]) \tag{2.35}$$

Figure 2.4 shows an *Extended Infomax* activation functions and their probability density functions for both super-Gaussian and sub-Gaussian distributions.



(**a**) (**b**)

Figure 2.4: Lee *et al.*'s activation functions and their derivatives or their probability density functions. The thick line is for super-Gaussianity. The dashed line is for sub-Gaussianity.

Cardoso [8] presented *Infomax and Maximum Likelihood for Blind Source Separation.* The proposed infomax algorithm is equivalent to maximum likelihood.

Hyvarinen [28] presented *Independent Component Analysis in the presence of Gaussian noise by maximizing joint likelihood.* The noise in the presence is nonlinear. For super-Gaussian data can be recovered by shrinkage operation and analyzed by competitive learning. For sub-Gaussian components anti-competitive learning can be used.

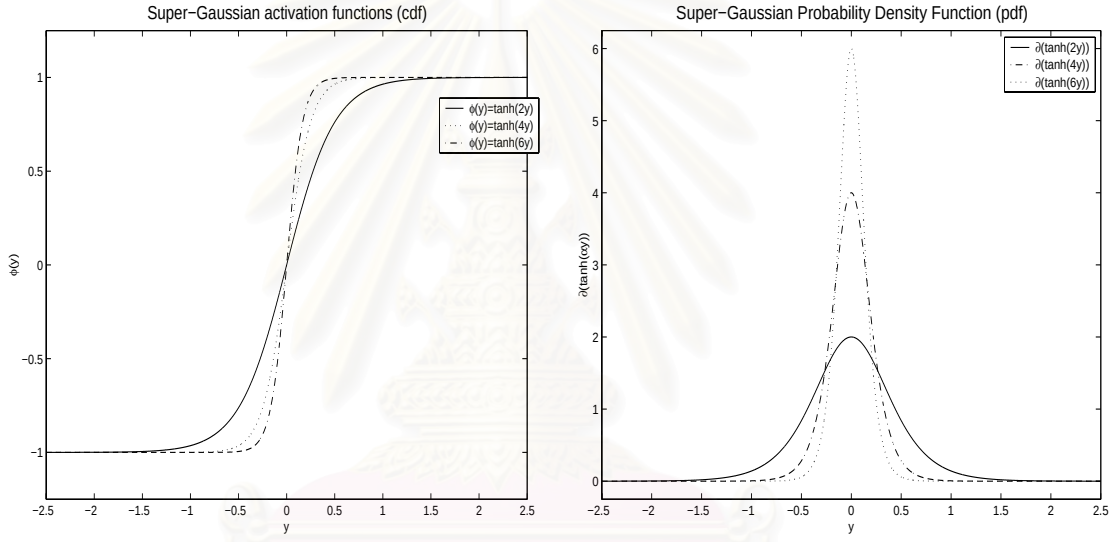
Xu, Cheung, and Amari [57] described a *Learned parametric mixture based ICA algorithm.* It is based on linear mixture and its separation capability is shown to be superior to the original model with prefixed nonlinearity. Experiments with sub-, super-, and combination Gaussians of sources confirm the applicability of the algorithms.

Cichocki *at al.* [18] explained the optimal nonlinear activation functions for the super-Gaussian and sub-Gaussian distibutions. For example, the super-Gaussian source signals

require the nonlinear function given by

$$\phi_i(y_i) = \tanh(\alpha_i y_i) \tag{2.36}$$

where $\alpha_i = 1/\sigma_{y_i}^2$. For the sub-Gaussian source signals, they suggested the nonlinear function $\phi_i(y_i) = y^3$. It can be concluded that the Cichocki's functions are the generalization of Douglas's functions. Figure 2.5 shows the graphical shape of family of super-Gaussian activation functions.



(**a**) (**b**)

Figure 2.5: (**a**) Family of Super-Gaussian activation functions and (**b**) their derivatives.

CHAPTER III

Proposed ICA Learning Methods

Four optimization techniques for Independent Component Analysis will be discussed, studied, and summarized in this dissertation. They are rewritten as follows:

- Increasing learning speed-up.
- Separation of mixed Kurtosis signed signals.
- Finding some low complexity activation functions.
- Proposing new learning methods using partial observations.

3.1 Increasing Learning Speed-up

We revise an important experimental result of Amari [2] and Haykin [26]. Our improvement is based on these few observations of Amari's and Haykin's results. Firstly, only a small fixed step of learning rate value can make the separation of signals $\mathbf{y}$ converged but a larger learning rate values in the range of $0.5 \leq \eta \leq 0.9$ causes output signals $\mathbf{y}$ to diverge. Secondly, the convergence speed can be increased by gradually reducing the learning rate until it is equal to zero. The learning rate may be initially set to any value. Thirdly, the reduction of learning rate in the current iteration step is done by dividing the learning rate from the previous iteration step, namely $\eta_{t+1} = \eta_t/1.005$. However, we find that this simple approach works well when $0.1 \leq \eta \leq 0.5$, but when $0.6 \leq \eta \leq 0.9$ the convergence speed is reduced and more iterations are required.

Instead of using a fixed divisor throughtout the learning period, we use different divisors for different learning rates. The learning rate should be divided proportionally to its value. If the learning rate is large then it should be divided by a large divisor. In addition, at each iteration t, a momentum term $\Delta\mathbf{M}$ and a momentum rate β are added to adjust the weight $\mathbf{W}$. Let $\mathbf{W}_t$ be the weight $\mathbf{W}$ at iteration t and $\Delta\mathbf{M}_t$ the momentum term at iteration t. The momentum term $\Delta\mathbf{M}_t$ is adjusted by using this rule $\Delta\mathbf{M}_t = \beta\Delta\mathbf{W}_t$. The stopping condition is defined in terms of the difference between $D(\mathbf{W}_t)$ at time t and $D(\mathbf{W}_{t-1})$ at time $t-1$. Figure 3.1 shows the typical ICA batch learning algorithm which we proposed in 2001 [12].

Algorithm: ICA Batch Learning
Input: Observed signals, $\mathbf{x}$
Output: Recovered signals $\mathbf{y}$
begin
  Load all observed signals $\mathbf{x}$
  i=0;
  while i $\leq$ NumberOfIterations
    Randomly initialize weight matrix $\mathbf{W}_0$
    $appropriate\ divisor = 1.0 + 10^{-2}\eta$
    Compute $\mathbf{y} = \mathbf{W}\mathbf{x}$
    Compute Kullback-Liebler Divergence $D_{pq}(\mathbf{W}_0)$
    $\Delta\mathbf{M}_0 = \mathbf{0}$
    Set $t = 1$
    repeat
      $\Delta\mathbf{W}_t = \eta_t(\mathbf{I} - f(\mathbf{y})\mathbf{y}^T)\mathbf{W}_{t-1}$
      $\mathbf{W}_{t+1} = \mathbf{W}_t + \Delta\mathbf{W}_t + \Delta\mathbf{M}_{t-1}$
      $\Delta\mathbf{M}_t = \beta\Delta\mathbf{W}_t$
      $\eta_{t+1} = \frac{\eta_t}{appropriate\ divisor}$
      Compute Kullback-Liebler Divergence $D_{pq}(\mathbf{W}_t)$
    until $|D(\mathbf{W}_t) - D(\mathbf{W}_{t-1})| < \epsilon$
  End While
end.

Figure 3.1: Standard ICA Batch learning algorithm.

3.2 Blind Source Extraction

We start with the demixing of mixed Kurtosis sign sources. Simply, the sources in an ICA problem are assumed to be identically and independently distributed, which are either super-Gaussian or sub-Gaussian distributed. The super-Gaussianity has a positive Kurtosis sign. In contrast, the sub-Gaussianity has a negative Kurtosis sign. Details of the distribution of sources were discussed in Section 2.7. For demixing the super-Gaussianity using learning algorithms proposed in [2], an activation function $\phi(y) = \tanh(\alpha y)$ is used. On the other hand, an activation function $\phi(y) = y^3$ is used for demixing the sub-Gaussianity [18]. Some experimental results in [2], [12] and [13] confirmed that sources could be recovered if they are identically distributed. On the other hand, when the sources s_i and s_j have different Kurtosis signs, the learning algorithm in [2] cannot recover the sources simultaneously. In [2], the authors used the *Kullback-Leibler Divergence:* (D_{pq}) between the joint probability density function, $p_{\mathbf{x},\mathbf{y}}(x, y)$, and the marginal *pdf*, $q_{\mathbf{x}}(x).q_{\mathbf{y}}(y)$, as a cost function. The relationship among them can be written as follows:

$$D_{pq} = \int_{-\infty}^{+\infty} p_{\mathbf{x},\mathbf{y}}(x, y) \log \frac{p_{\mathbf{x},\mathbf{y}}(x, y)}{q_{\mathbf{x}}(x).q_{\mathbf{y}}(y)} dxdy \tag{3.1}$$

The D_{pq} values are usually positive and will be zero when the joint *pdf* is equal to the marginal *pdf*. In case of identical sources, it is easy to estimate an optimal marginal *pdf* for the current joint *pdf*. For example, x and y are super-Gaussian distributed which have a sharp peak and a long tail. The activations for x and y will support one another. In contrast, it is more difficult to find the suitable distributions for $p_{\mathbf{x}}(x)$ and $p_{\mathbf{y}}(y)$ when they are nonidentically distributed. For example, $p_{\mathbf{x}}(x)$ is super-Gaussian distributed but $q_{\mathbf{y}}(y)$ is sub-Gaussian distributed. The sub-Gaussian source is flatter than the super-Gaussian source. Possibly, the mixed signals are Gaussian distributed. If it is so, it is obvious that each mixed channel is independent from each other [30] and, then, we cannot recover the sources from the mixtures. In order to solve this problem,

we first extract an observed signal into subgroups via the Kurtosis signs of a prewhitened observed signals.

The mixing and demixing processes of the unknown source signals $s_i(k), 1 \leq i \leq n$, which may have distinguished mathematical or physical model [18] are our main concern. The source distribution $p(\mathbf{s})$ has been transformed to $\hat{p}(\mathbf{s})$ that depends on the mixing matrix $\mathbf{B}$. In other words, after the transformation with the mixing matrix $\mathbf{B}$, the Kurtosis sign of the source might be changed. In order to recover the Kurtosis sign of the sources, we need a prewhitening step on the observed signals $\mathbf{x}(k)$. The proof on the properties of the Kurtosis of source s_i and prewhitened channel $\tilde{x}_i$ will be described as follows:

Let s_i and s_j be independently distributed,

$$E[\mathbf{s}\mathbf{s}^T] = \mathbf{I}. \tag{3.2}$$

The Kurotosis of s_i is described as:

$$kurt(\mathbf{s}) = \frac{E[\mathbf{s}^4]}{(E[\mathbf{s}^2])^2} - 3 \tag{3.3}$$

After the linear transformation with mixing matrix $\mathbf{B}$, it will be

$$\mathbf{x} = \mathbf{B}\mathbf{s} \tag{3.4}$$

or

$$\begin{aligned} x_1 &= b_{11}s_1 + b_{12}s_2 + b_{13}s_3 \\ x_2 &= b_{21}s_1 + b_{22}s_2 + b_{23}s_3 \\ x_3 &= b_{31}s_1 + b_{32}s_2 + b_{33}s_3. \end{aligned} \tag{3.5}$$

Then, its Kurtosis may be changed to

$$kurt(x_i) = b_{i1}^4 kurt(s_1) + b_{i2}^4 kurt(s_2) \tag{3.6}$$

The prewhitening step will decorrelate the existing correlation between the observed channels.

$$\tilde{\mathbf{x}} = \mathbf{Z}\mathbf{x} = \mathbf{Z}\mathbf{B}\mathbf{s} \tag{3.7}$$

where

$$\mathbf{Z} = \mathbf{D}^{-1/2}\mathbf{V}^T, \tag{3.8}$$

$$\mathbf{D}^{1/2} = \begin{bmatrix} \frac{1}{\sqrt{d_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{d_2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{d_3}} \end{bmatrix}, \tag{3.9}$$

and

$$\mathbf{V}^T = \begin{bmatrix} v_{11}^{max} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32}^{max} \\ v_{13} & v_{23}^{max} & v_{33} \end{bmatrix}. \tag{3.10}$$

The product of minimum values, $v_{12}, v_{13}, v_{21}, v_{22}, v_{31}$ and v_{33}, can be discarded. Hence, we approximately obtain:

$$\mathbf{Z} \approx \begin{bmatrix} \frac{v_{11}^{max}}{\sqrt{d_1}} & 0 & 0 \\ 0 & 0 & \frac{v_{32}^{max}}{\sqrt{d_2}} \\ 0 & \frac{v_{23}^{max}}{\sqrt{d_3}} & 0 \end{bmatrix} = \begin{bmatrix} z_{11} & 0 & 0 \\ 0 & 0 & z_{32} \\ 0 & z_{23} & 0 \end{bmatrix}. \tag{3.11}$$

Then $\mathbf{Z}$ is orthogonal matrix. After the de-transformation with $\mathbf{Z}$, we obtain:

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} z_{11}b_{11}s_1 \\ z_{32}b_{22}s_2 \\ z_{23}b_{33}s_3 \end{bmatrix}. \tag{3.12}$$

Then the Kurtosis of $\tilde{x}_i$ is

$$\begin{bmatrix} Kurtosis(\tilde{x}_1) \\ Kurtosis(\tilde{x}_2) \\ Kurtosis(\tilde{x}_3) \end{bmatrix} = \begin{bmatrix} z_{11}^4 b_{11}^4 Kurtosis(s_1) \\ z_{32}^4 b_{22}^4 Kurtosis(s_2) \\ z_{23}^4 b_{33}^4 Kurtosis(s_3) \end{bmatrix}. \tag{3.13}$$

It means that each component $\tilde{x}_i$ and $\tilde{x}_j$ are decorrelated or mutually independent after the prewhitening transformation. In other words, the Kurtosis sign of each source is recovered after the prewhitening step. The blind source extraction approach has been used for separating the source signals. The prewhitened signals are classified into 2 sub-groups, which are the positive and the negative Kurtosis signed signals. Then, the super-Gaussian source separation is performed. In this stage, the positive Kurtosis signed signals are selected. Next, the negative Kurtosis signed signals are consequently fed into the previous separating procedure. Figure 3.2 shows the diagram of our proposed learning procedure.

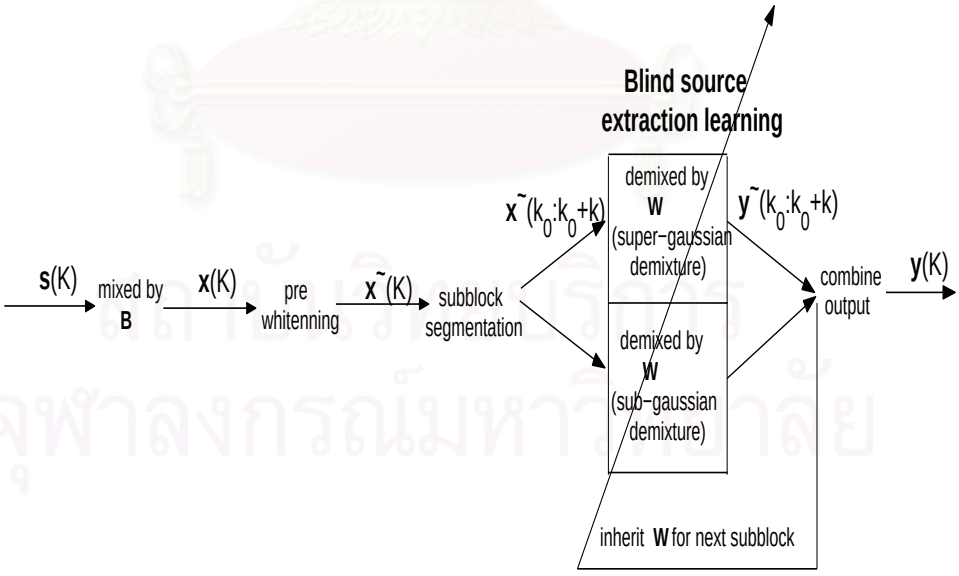


Figure 3.2: Blind source extraction.

3.3 Activation Functions for Mixed Kurtosis Sign Sources

Third, as described above, the activation functions for demixing super-Gaussianity and sub-Gaussianity are $\phi(y_i) = \tanh(\alpha_i y_i)$ and $\phi(y_i) = y_i^3$, respectively, where $1 \leq i \leq n$. Two weak points of both functions are listed below:

- They are of high order complexity which requires a high computational time per instruction.
- They are difficult to implement on the circuit level.

In order to obtain the low computational cost activation functions, some quadratic approximation activation functions for separating the super-Gaussian and sub-Gaussian channels are proposed. In the next two subsections, the low computational functions for demixing of the super-Gaussian and sub-Gaussian source signals will be presented. The proposed functions are the quadratic function.

3.3.1 Low Computational Function for Super-Gaussianity

In 1992, Kwan [36] presented the **KTLF** (**K**wan **T**anh-**L**ike activation **F**unction), which is the 2^{nd} order function. This function is an approximation of $\tanh(2\mathbf{y})$ function. He divided the approximation curve into three regions, which are the upper bound $\mathbf{y} \geq L$, the nonlinear logistic tanh-like area $-L < \mathbf{y} < L$, and the lower bound $\mathbf{y} \leq -L$. All regions are described below:

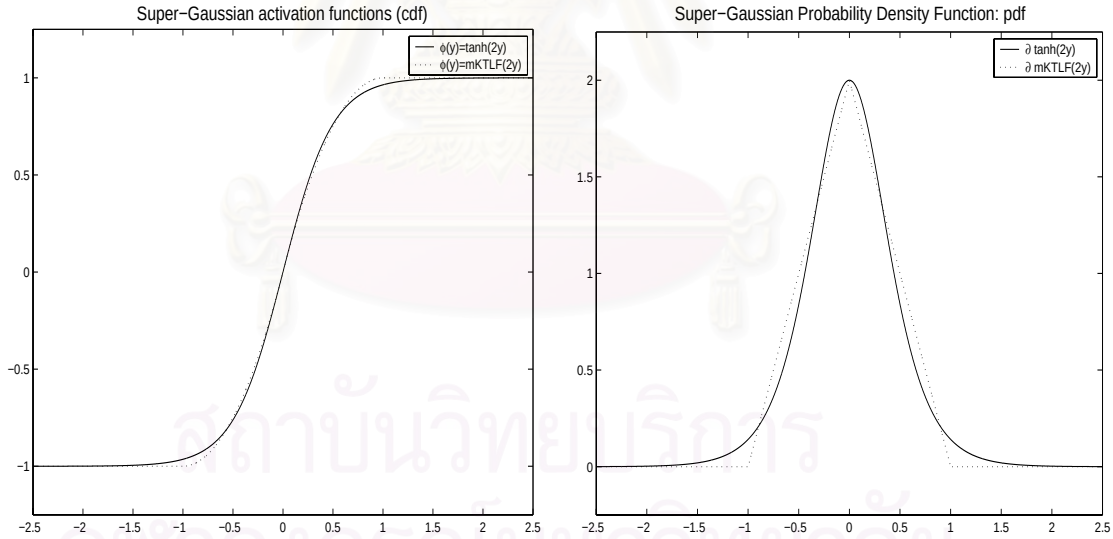
$$\phi(\mathbf{y}) = \begin{cases} 1, & (\mathbf{y} \geq L) \\ \frac{\mathbf{y}}{L}(\gamma - \theta\frac{\mathbf{y}}{L}), & (0 \leq \mathbf{y} < L) \\ \frac{\mathbf{y}}{L}(\gamma + \theta\frac{\mathbf{y}}{L}), & (-L < \mathbf{y} < 0) \\ -1, & (\mathbf{y} \leq -L) \end{cases} \tag{3.14}$$

The shape of **KTLF** curve is controlled by $\gamma = 2/L$ and $\theta = 1/L^2$. The approximation function given in equation (3.14) corresponds to the $\tanh(2\mathbf{y})$ function. Consequently,

the term $\frac{\alpha}{2}$ is needed for controlling $\mathbf{y}$, and we also suggest $L = 1$. Then, the modified equation **mKTLF** can be rewritten as follows:

$$\phi(\mathbf{y}) = \begin{cases} 1, & (\mathbf{y} \geq 1) \\ \hat{\mathbf{y}}(2 - \hat{\mathbf{y}}), & (0 \leq \mathbf{y} < 1) \\ \hat{\mathbf{y}}(2 + \hat{\mathbf{y}}), & (-1 < \mathbf{y} < 0) \\ -1, & (\mathbf{y} \leq -1) \end{cases} \tag{3.15}$$

where $\alpha_i = 1/\sigma_{y_i}^2$ is an upper-peak of the derivative of the activation function and $\hat{y}_i = \frac{\alpha_i y_i}{2}$. The bigger α value it is, the lower distribution it has. In other words, the channel has a sharper peak than the other's. Figure 3.3 shows $\tanh(\alpha\mathbf{y})$, its approximation (the dash line) and their derivatives. From the figure we can conclude that the fraction $\frac{\alpha}{2}$ is fitted for all $\tanh(\alpha\mathbf{y})$.



(**a**) (**b**)

Figure 3.3: (**a**) The $\phi(\mathbf{y}) = \tanh(2\mathbf{y})$ activation function and its approximation from equation (3.16). (**b**) Their derivatives.

Rearrange the term $\hat{\mathbf{y}}(2 \mp \hat{\mathbf{y}})$ in equation (3.15), we obtain then:

$$\phi(\mathbf{y}) = \begin{cases} 1, & (\mathbf{y} \geq 1) \\ 1-(1-\hat{\mathbf{y}})^2, & (0 \leq \mathbf{y} < 1) \\ -1+(1+\hat{\mathbf{y}})^2, & (-1 < \mathbf{y} < 0) \\ -1, & (\mathbf{y} \leq -1) \end{cases} \tag{3.16}$$

3.3.2 Low Computational Function for Sub-Gaussianity

In this subsection, a new 2^{nd} order approximation of $\phi(\mathbf{y}) = \mathbf{y}^{11}$ [2] and $\phi(\mathbf{y}) = \mathbf{y}^3$ [21, 18] are proposed. Given the graphical representation of the sub-Gaussian activation functions illustrated in Figure 3.4, it can be seen that the sub-Gaussian activation functions can be separated into two regions: the positive and the negative regions. For demixing the sub-Gaussian distribution, we propose the bisection paraboloid function given in equation (3.17), which is a good approximation for the previously reported functions in the literatures [13, 14].

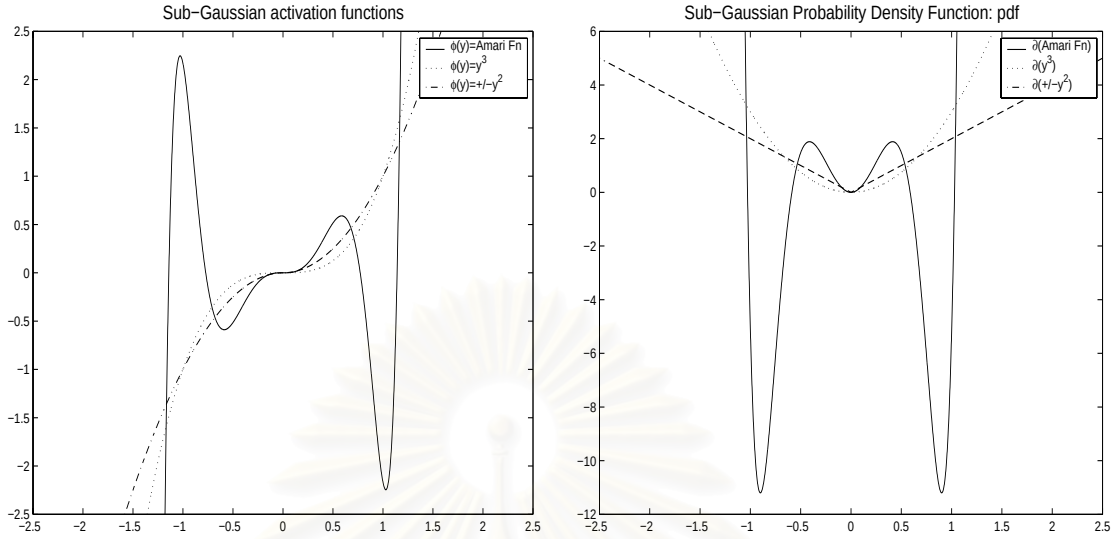
$$\phi(\mathbf{y}) = \begin{cases} +\mathbf{y}^2, & (\mathbf{y} \geq 0) \\ -\mathbf{y}^2, & (\mathbf{y} < 0) \end{cases} \tag{3.17}$$

Figure 3.4 shows sub-Gaussian activation function in the literature [2, 21] and the bisection paraboloid function given in equation (3.17).

3.4 Considerations on the Online Learning Subblock Size

In current batch mode learning [26], all incoming data signal must be retained and used during the computation. This process is inappropriate and infeasible if the blind separation must be used for online applications and must be implemented on the VLSI circuit level due to the following constaints:

- amount of computer memory



(**a**) (**b**)

Figure 3.4: (**a**) Graphical representation of an activation function of 11^{th}, 3^{rd}, and 2^{nd} order activation function. (**b**) Their derivatives.

- the number of CPU time computation
- high computational complexity

Moreover, the recovered results for batch or online learning method will be produced after the learning system reaches the saturated region or the local minimum. The proposed solutions to the learning methods which reduce the usage of computer memory, CPU time computation, computational complexity, and can be used in online real world learning systems will be proposed.

The ICA problem is a multi-dimensional data analysis problem as Principal Component Analysis (PCA). Each source has its own properties and characteristics such as Kurtosis sign and value, variance and waveform or shape, etc. After the mixture stage, signal characteristics will be changed. For example, its Kurtosis will change from negative sign to positive sign. A variance will change from $\sigma_{s_i}^2$ to $\sigma_{x_j}^2$ after the mixture. Precisely, ICA algorithm task is to recover the properties of each channel.

In this subsection, we describe an online subblock ICA learning algorithm. We used an unsupervised multi-layer feed forward neural network for demixing the non-Gaussian channels. Our learning method is a combination of the online and the batch learning techniques. First, unknown or observed signals x_j are fed into the input layer, where $1 \leq j \leq m$. Second, the signals are, then, passed to the prewhitening step via PCA layer. The results from this stage are whitened. Third, the prewhitened signals are separated into subblocks size k, $x_j(k_0 : k_0 + k)$, where k_0 is the starting index of the subblock. And the Mutual Information learning method, detailed in Section 2.6, is used. The output signals $y_i(k_0 : k_0 + k)$ are produced by $y_i(k_0 : k_0 + k) = \mathbf{W}x_j(k_0 : k_0 + k)$, where $\mathbf{W}$ is called the demixing matrix. If the output channels $y_i(k_0 : k_0 + k)$ depend on each other, the natural gradient descent in equation (2.28) updates the demixing matrix $\mathbf{W}$ and produced the output signals $y_i(k_0 : k_0 + k)$ until they become independent.

Figure 3.5 shows an algorithm for finding the partial observation length. In this subsection, we describe an algorithm for finding the partial observation length. As described in Section 3.3, the source distribution functions not only have their own characteristics, but also their own principal directions. After the prewhitening step, the observed signals become whitened, $E[\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}(k)^T] \cong E[\mathbf{s}(k)\mathbf{s}(k)^T] = \mathbf{I}$. Then, we found a significant point that, for each row i^{th} of eigenvector matrix $\mathbf{V}$, one of them will be maximized. The maximal value $\|v_{ij}\|, (1 \leq j \leq n)$ guides the principal direction of each component. This criterion is used for estimating an optimal subblock length. In the simulation, we require an eigenvector $\|v_{ij}\| \approx 1$ for each principal direction. The optimal value for the subblock length obtained is $k = 4096$. The subblock size is based on four benchmark sources accessible at http://speech.kaist.ac.kr/~jangbal/ch1bss. Next, an ICA based on the MI algorithm [2] performs a separating process on a coming subblock. Figure 3.6 displays an ICA online subblock learning method. This algorithm will repeatedly calculate the demixing matrix $\mathbf{W}$ until the $D_{pq}(\mathbf{W})$ approaches zero.

Algorithm: Find Subblock Length
Aim: To find an optimal subblock length for ICA learning.
Input: Observed signals: $\mathbf{x}$.
Output: Optimal subblock length: k.
Begin
  Start with $i = 2$;
  k = power(2, i);
  Repeat
    xSubblock = LoadSubblock(k);
    xPre = prewhitening(xSubblock);
    $\mathbf{V}$ = eigenvector(cov(xPre));
    $i = i + 1$;
    k = power(2, i);
  until $argmax \|v_{ij}\|$ is reached
End

Figure 3.5: Algorithm for finding an optimal subblock size, k.

Figure 3.6 shows an ICA subblock learning algorithm. This algorithm is derived from the algorithm presented in Figure 3.1. It can be seen that the algorithm produces the recovered signal $y(k_0 : k_0 + k)$ in every r iterations. In contrast, the typical batch and online learning will produce output after the demixing weight reaches the saturated region. Hence, practically, our proposed learning method produced the result faster than the typical batch learning. The result of computer simulation based on CPU time usage is shown in Section 4.2. Theoretically, the increase of speed in obtaining results for the online subblock method is proved in the following theorem.

Theorem 1. *ICA online subblock learning is of lower computational complexity than the batch learning.*

Proof Let us consider that K is the total time index of the signal and k is the time index number for each subblock, where $k < K$. The learning equation (2.28) can be rewritten as follows:

Algorithm : ICA Subblock Learning
Aim: Separate an observed signal
Input : 1. Prewhitened observed signal: $\tilde{\mathbf{x}}$.
2. Optimal subblock size: k
3. Initial demixing matrix: $\mathbf{W}$
Output: recovered signal: $\mathbf{y}$
Begin
noBlock = size(xPre) / k;
for r:=1 **to** noBlock
$\mathbf{x}$ = LoadSubblock(k);
$D(\mathbf{W}) = Kullback(\mathbf{x}, \mathbf{W})$;
$\Delta\mathbf{M} = 0$;
do
$\mathbf{y} = \mathbf{W}*\mathbf{x}$;
$\Delta\mathbf{W} = \eta_t(\mathbf{I} - \phi(\mathbf{y})\mathbf{y}^T)\mathbf{W}$;
$\mathbf{W} = \mathbf{W} + \Delta\mathbf{W} + \Delta\mathbf{M}$;
$\Delta\mathbf{M} = \beta * \Delta\mathbf{W}$;
$D(\mathbf{W}) = Kullback(\mathbf{x}, \mathbf{W})$;
while $(D(\mathbf{W}) > 0)$
$\mathbf{y}_{r^{th}} = \mathbf{W} * \mathbf{x}$;
endfor
End

Figure 3.6: Algorithm for calculating the demixing matrix $\mathbf{W}$ for online subblock learning.

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \eta[\mathbf{I} - \phi(\mathbf{y})\mathbf{y}^T]\mathbf{W}_t + \beta\Delta\mathbf{W}_t \tag{3.18}$$

The computational complexity of equation (3.18) depends on the correlation $\phi(\mathbf{y})\mathbf{y}^T$, where $\mathbf{y}^T$ is a transpose of $\mathbf{y}$. For the batch learning method with time index K, the complexity of equation (3.18) is of $O(K^3)$

On the other hand, for the online subblock learning method, we have $\frac{K}{k}$ subblocks. The computational complexity of equation (3.18) is of $\frac{K}{k}O(k^3) = O(K.k^2)$. It is obvious that $O(K.k^2) < O(K^3)$, where $k < K$. Hence, the ICA online subblock learning is of lower computational complexity than the batch learning method. □

CHAPTER IV

Experimental Results

The following four essential points will be addressed in our experiments, which are:

- Increasing learning speed up.
- Separation of mixed Kurtosis signed signals.
- Finding some low complexity activation functions.
- Proposing new learning methods using partial observations.

4.1 Results on Increasing Learning Speed-Up

We simulate our algorithm on the computer using three synthetic signals, a random mixing matrix $\mathbf{B}$, and a initial random de-mixing matrix $\mathbf{W}$. Each signal contains 2500 data points. The stopping condition is defined in terms of the difference between $D_{pq}(\mathbf{W}_t)$ at time t and $D_{pq}(\mathbf{W}_{t-1})$ at time $t-1$. The convergent test is set as $\Delta D_{pq}(\mathbf{W}) \leq 0.000001$. We simulate five iterations for each step with the learning rate values of $0.1 \leq \eta \leq 0.9$ and step size of 0.1.

$$\begin{aligned} 1.s_1(t) &= 0.1\sin(400t)\cos(30t) \\ 2.s_2(t) &= 0.01\text{sign}[\sin(500t + 9\cos(40t))] \\ 3.s_3(t) &= \text{uniform noise in range [-1,1]} \end{aligned} \tag{4.1}$$

Our improvement is based on these few observations of Amari [2] and Haykin [26] results. Firstly, only a small fixed step of learning rate value can make the separation

of signals $\mathbf{y}$ to converge but a larger learning rate values in the range of $0.5 \leq \eta \leq 0.9$ cause output signals $\mathbf{y}$ to diverge. Secondly, the convergence speed can be increased by gradually reducing the learning rate until it is equal to zero. The learning rate may be initially set to any value. Thirdly, the reduction of learning rate in the current iteration step is done by dividing the learning rate from the previous iteration step, namely $\eta_{t+1} = \eta_t/1.005$. However, we find that this simple approach works well when $0.1 \leq \eta \leq 0.5$, but when $0.6 \leq \eta \leq 0.9$ the convergence speed is reduced and more iterations are required. Instead of using a fixed divisor throughtout the learning period, we use different divisors for different learning rates. The learning rate should be divided proportionally to its value. If the learning rate is large then it should be divided by a large divisor. In addition, at each iteration t, a momentum term $\Delta\mathbf{M}$ and a momentum rate β are added to adjust the weight $\mathbf{W}$. Let $\mathbf{W}_t$ be the weight at iteration t and $\Delta\mathbf{M}_t$ the momentum term at iteration t. The momentum term $\Delta\mathbf{M}_t$ is adjusted by using this rule $\Delta\mathbf{M}_t = \beta\Delta\mathbf{W}_{t-1}$ as reported in [12]. We use an activation function defined in equation (2.32) [2]. Five types of examples are provided to measure the efficiency of the algorithm.

1. Fixed learning rate value: $0.1 \leq \eta \leq 0.9$
2. Approach Learning rate to 0 by $\eta_t = \eta_{t-1}/1.005$
3. Approach Learning rate to 0 by $\eta_t = \frac{\eta_{t-1}}{1.0+\eta 10^{-2}}$ and 0.01 momentum rate.
4. Approach Learning rate to 0 by $\eta_t = \frac{\eta_{t-1}}{1.0+\eta 10^{-2}}$ and 0.10 momentum rate.
5. Approach Learning rate to 0 by $\eta_t = \frac{\eta_{t-1}}{1.0+\eta 10^{-2}}$ and 0.20 momentum rate.

Figure 4.1 shows three original signals $s_i(t)$, observation signals $x_i(t) = \sum_{i,j=1}^{3} b_{ji}s_i(t)$ and recovered signals $y_i(t) = \sum_{i,j=1}^{3} w_{ji}x_i(t)$ on the 1^{st}, 2^{nd} and 3^{rd} column, respectively. The recovered signals were permuted and rescaled. For example, input signals $s_1(t)$, $s_2(t)$ and $s_3(t)$ were recovered on the $y_3(t)$, $y_1(t)$ and $y_2(t)$, respectively.

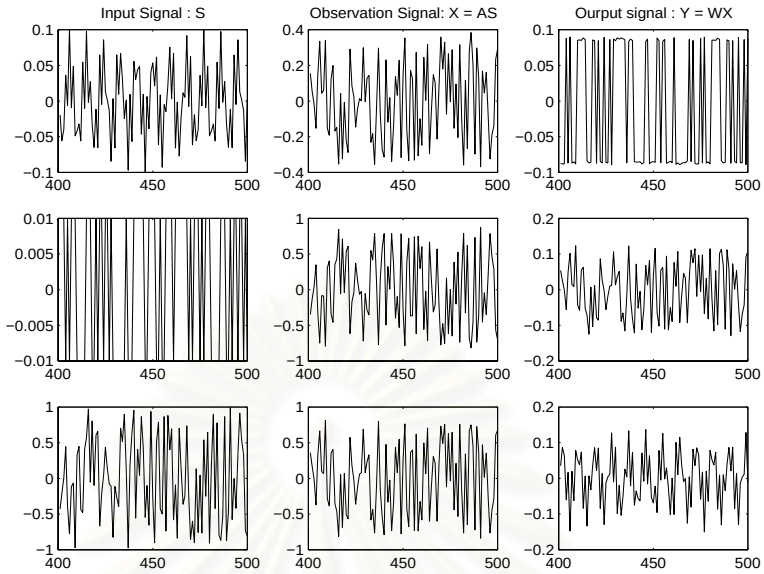


Figure 4.1: Successful separation of ICA Examples.

The comparison on the number of epochs of all experiments are illustrated in Figure 4.2. The dashed-and-dotted line is for fixed step of learning rate value $\eta_{t+1} = \eta_t$. The thick-and-marked line is for varied step of learning rate value $\eta_{t+1} = \eta_t/1.005$. The dotted line is for varied step of learning rate value with momentum term $\eta_{t+1} = \eta_t/1.0 + \eta 10^{-2}, \beta = 0.01$. The thick line is for varied step of learning rate value with momentum term $\eta_{t+1} = \eta_t/1.0 + \eta 10^{-2}, \beta = 0.10$. The thick-and-dotted line is for varied step of learning rate value with momentum term $\eta_{t+1} = \eta_t/1.0 + \eta 10^{-2}, \beta = 0.20$. The dashed-and-dotted line shows that only a small fixed step of learning rate value can make the separation of signals $\mathbf{y}$ to converge but a larger learning rate values in the range of $0.5 \leq \eta \leq 0.9$ cause output signals $\mathbf{y}$ to diverge. The reduction of learning rate in the current iteration step, done by dividing the learning rate from the previous iteration step, can make the separation $\mathbf{y}$ converged for all learning rate values. However, we find that this simple approach works well when $0.1 \leq \eta \leq 0.5$, but when $0.6 \leq \eta \leq 0.9$ the convergence speed is reduced and more iterations are required. Instead of using a fixed divisor throughtout the learning period, we use different divisors for different learning

rates. The learning rate should be divided proportionally to its value. If the learning rate is large then it should be divided by a large divisor. In addition, the difference of the momentum rate can make the separation **y** converge at different speed.

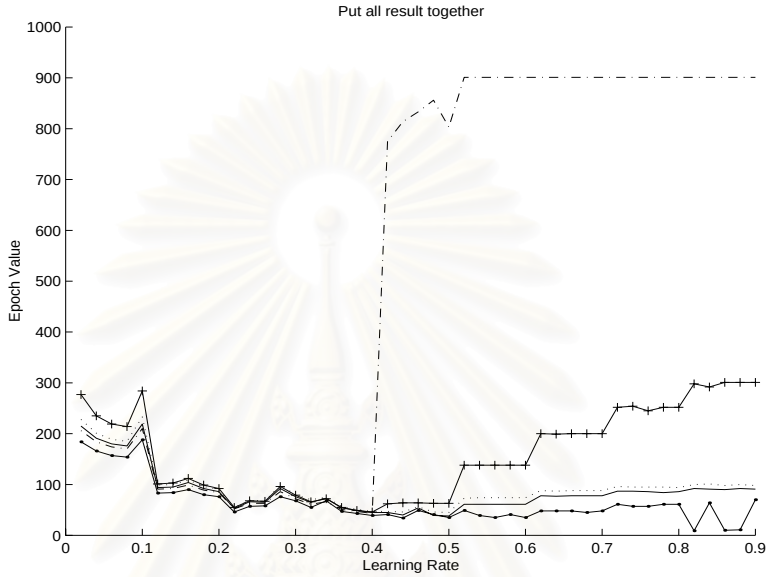


Figure 4.2: Comparison experimental results. Five types of lines are used to denote the results. The dashed-and-dotted line is for fixed η. The thick-and-marked line is for $\eta = \eta/1.005$. The dotted line is for $\eta = \eta/1.0 + \eta 10^{-2}, \beta = 0.01$. The thick line is for $\eta = \eta/1.0 + \eta 10^{-2}, \beta = 0.10$. The thick-and-dotted line is for $\eta = \eta/1.0 + \eta 10^{-2}, \beta = 0.20$.

4.2 Results on Low Computation Complexity Learning Methods

4.2.1 Initial Conditions and Learning Criteria

A mixing matrix **B** is randomly generated. As presented in [12], we initialized the learning rate value $\eta = 0.05$ and the momentum rate value $\beta_{t+1} = 0.1\eta_t$. At each learning iteration, the learning rate was decreased by 1.0005 ($\eta_{t+1} = \eta_{t+1}/1.0 + \eta 10^{-2}$). An initial demixing matrix $\mathbf{W}_0$ is set as the eigenvector of the prewhitened signals, detailed

in Section 3.2. For improving the learning performance, we exploit the relationship between two consecutive online subblocks. The final demixing matrix $\mathbf{W}$ of the r^{th} subblock is set to the initial demixing matrix of the $(r+1)^{th}$ subblock. The weight inheritance will maintain the output channel. The simulations are run on Pentium 4 with CPU speed of 2.4 GHz.

The simulations for both uni-distributed and multi-distributed mixtures are performed. The uni-distributed mixture simulation performs the demixing algorithm on only one source distribution, which is either super-Gaussian or sub-Gaussian distribution. On the other hand, the multi-distributed mixture performs the demixing procedure on the mixed Kurtosis sign sources. The source signals are possibly super-Gaussian and sub-Gaussian distribution. In this kind of mixtures, the sequential source separation, or blind source extraction, is used.

In this experiment, the source signals are uni-distributed. The super-Guassian data sets consist of four sound sources taken from http://speech.kaist.ac.kr/~jangbal/ch1bss. For sub-Gaussianity, we simulated our algorithm using the following three synthetic signals:

$$\begin{aligned} \mathbf{s}_1(t) &= 0.1\sin(400t)\cos(30t) \\ \mathbf{s}_2(t) &= 0.01\text{sign}[\sin(500t + 9\cos(40t))] \\ \mathbf{s}_3(t) &= \text{uniform noise in range [-0.05,0.05]} \end{aligned} \tag{4.2}$$

Each channel contains 46,560 data points. The online subblock size for both distributions were found to be $4,096$ data points. It is known that all activation functions in Section 3.3 can recover the source signals from the observed signals, but the recovered signals will be permuted and scaled over the output channels [18].

4.2.2 Performance Correlation Index

The *performance index* between the demixing matrix $\mathbf{W}$ and the mixing matrix $\mathbf{B}$ [2] is adopted and its equation is rewritten as below:

$$PI = \sum_{i=1}^{N}(\sum_{j=1}^{N}\frac{|p_{ij}|}{max_k\,|p_{ik}|}-1)+\sum_{j=1}^{N}(\sum_{i=1}^{N}\frac{|p_{ij}|}{max_k\,|p_{kj}|}-1) \tag{4.3}$$

where $\mathbf{P}=\mathbf{WB}$. In practice, the separating equation (2.28) is mainly significant with respect to the cross correlation between the activation function of the output $\phi(\mathbf{y})$ and the output $\mathbf{y}^T$. It can be observed that the cross correlation gradually becomes an identity matrix during the learning iterations. It means that the recovered signals y_i and y_j are getting more and more independent from each other after each learning iteration. Then, the performance index from equation (4.3) can be replaced with the following *performance correlation index*:

$$PCI = \sum_{i=1}^{N}(\sum_{j=1}^{N}\frac{|c_{ij}|}{max_k\,|c_{ik}|}-1)+\sum_{j=1}^{N}(\sum_{i=1}^{N}\frac{|c_{ij}|}{max_k\,|c_{kj}|}-1) \tag{4.4}$$

The matrix $C=\phi(\mathbf{y})\mathbf{y}^T$ is close to the identity matrix when the recovered signals y_i and y_j are mutually uncorrelated or linearly independent. Obviously, the performance correlation index approaches zero when the recovered signals, y_i and y_j, become independent, which is similar to the properties of the Kullback-Leibler divergence.

4.2.3 Similarity Measure

Similarity measure is used for evaluating the difference of the waveforms between the source vector and its corresponding recovered vector. The *Scalar Product* or *Dot Product* between the source $\mathbf{s}$ and the recovered signal $\mathbf{y}$ is used. Assume $\mathbf{s}$ is the source signal, and $\mathbf{y}$ is the recovered source signal. $\mathbf{s}$ and $\mathbf{y}$ have the same distribution and their dot product or cosine is defined as follows:

$$\mathbf{s}.\mathbf{y} = \|\mathbf{s}\|\,\|\mathbf{y}\|\cos(\theta) \tag{4.5}$$

or

$$\cos(\theta) = \frac{\mathbf{s}.\mathbf{y}}{\|\mathbf{s}\| \, \|\mathbf{y}\|} \tag{4.6}$$

Vectors $\mathbf{s}$ and $\mathbf{y}$ are similar when its $\cos(\theta)$ approaches one.

Experiment 1: Uni-distributed Mixtures

Figure 4.3 and 4.4 show the source, the mixed, and the recovered signals for super-Gaussianity and sub-Gaussianity, respectively, which have been produced using our activation functions described in Sections 3.3.1 and 3.3.2, respectively. The order of the recovered signals were manually rearranged so that each recovered signal corresponds to the similar original signal.

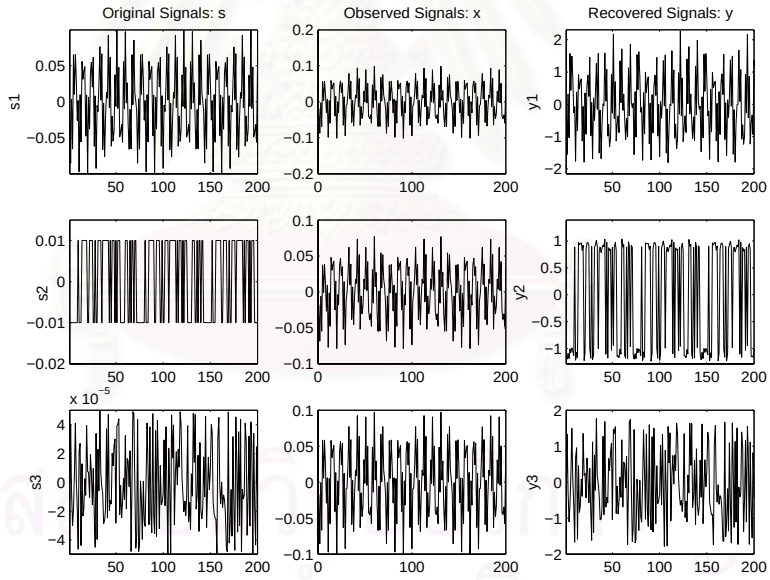


Figure 4.3: The source, the mixed and the recovered signals for sub-Gaussianity.

Figure 4.5 shows the source, the mixed and the recovered signals for super-Gaussianity, which have been produced using the online subblock learning method and our activation functions described in Section 3.3.1. Figures 4.5(**a**), (**b**) and (**c**) are the results from the first subblock ($1 \leq t \leq 4096$), the seventh subblock ($24577 \leq t \leq 28672$) and the eleventh subblock ($40961 \leq t \leq 45056$), respectively.

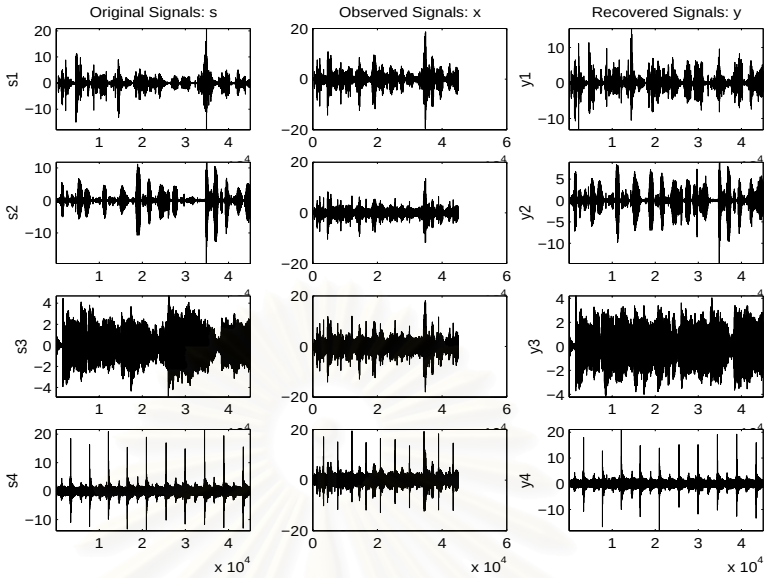


Figure 4.4: The source, the mixed and the recovered signals for super-Gaussianity.

In this experiment, the Kurtosis value is not considered because after the linear transformation in equation (1.2), the Kurtosis sign of the source s_i will not be changed if the mixing matrix $\mathbf{B}$ is unbiased. The similarity measure in this dissertation is the $\cos(\theta)$ between the source signal s_i and its recovered signal y_j, which is given in Tables 4.1, 4.2 and 4.3. For the super-Gaussian source signals, all of the algorithms in Section 3.3 are based on the hyperbolic-Cauchy distribution ($\tanh(4\mathbf{y})$). In this simulation, the learning parameter $\alpha_i = 4$ is suitable for the demixing of the super-Gaussian source signals.

Tables 4.1, 4.2 and 4.3 show the recovered signals from our proposed algorithm (**LF-ICA**) which are similar to the results obtained from using other activation functions. For some recovered channels, in both super-Gaussian and sub-Gaussian distributions, our proposed learning method produces better results than the others'. Moreover, our proposed learning method is of lower computational complexity and easy for hardware implementation. It means that our proposed algorithm is applicable to real-world problems.

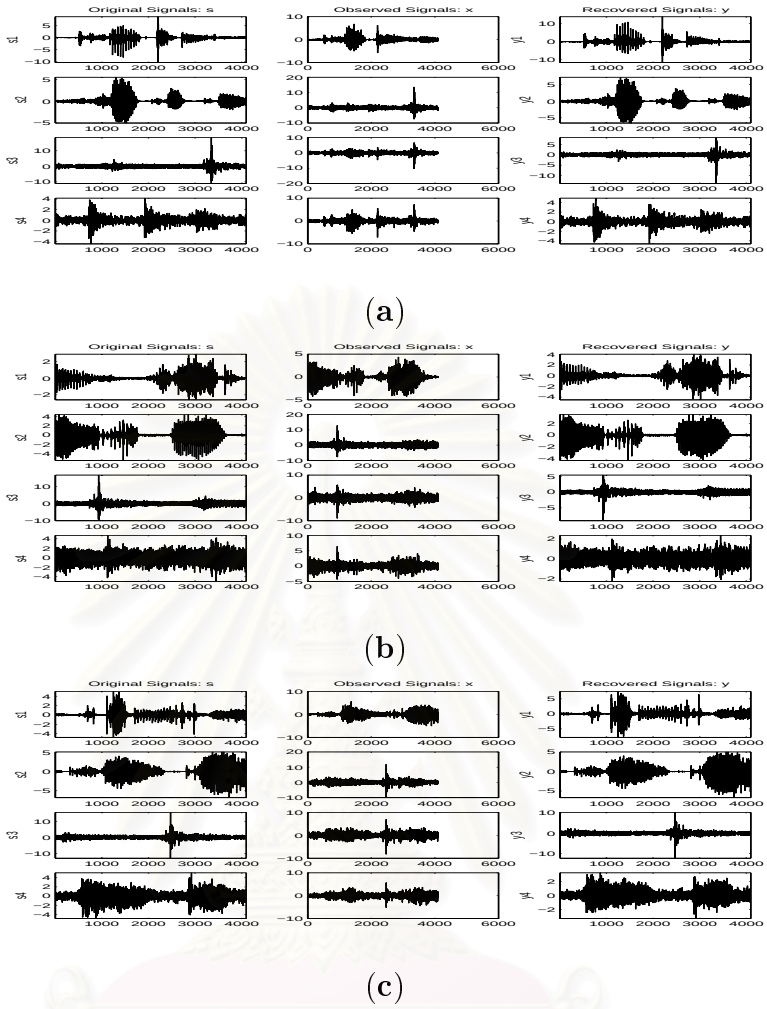


Figure 4.5: The source, the mixed and the recovered online subblock signals for super-Gaussianity. (**a**) The first subblock. (**b**) The seventh subblock. (**c**) The eleventh subblock.

Tables 4.4 and 4.5 show the CPU time usage for both the online subblock learning and the batch learning methods. We simulated all functions from the literatures [18, 39], including our proposed activation functions [13]. For example, considering our proposed function for demixing super-Gaussian channels in Table 4.4, the total CPU time usage for an online subblock learning is 190.2830 seconds. The first output subblock is produced after the learning procedure is run on CPU 34.6500 seconds. Then, in 28.2100 second, the next output subblock is produced, and so forth. For batch mode learning, data sets

Table 4.1: The Similarity Measure using Cichocki's function for an online subblock learning and batch learning methods based on uni-distributed mixtures.

Learning Methods	Subblock Number	Sources						
		sup_1	sup_2	sup_3	sup_4	sub_1	sub_2	sub_3
Online Subblock Method	1	-1.0000	1.0000	-0.9986	-0.9997	0.9966	0.9998	1.0000
	2	-1.0000	0.9999	-0.9964	-1.0000	0.9978	0.9996	0.9998
	3	-0.9999	1.0000	-0.9999	-0.9994	0.9986	0.9994	1.0000
	4	-1.0000	1.0000	-0.9993	-0.9999	0.9999	0.9996	0.9999
	5	-1.0000	1.0000	-0.9996	-0.9996	0.9999	0.9994	0.9999
	6	-0.9997	1.0000	-0.9998	-0.9987	0.9978	0.9991	0.9999
	7	-1.0000	1.0000	-0.9999	-0.9988	0.9926	0.9986	0.9999
	8	-1.0000	0.9998	-0.9995	-0.9997	0.9885	0.9991	1.0000
	9	-1.0000	1.0000	-0.9984	-0.9989	0.9834	0.9996	0.9999
	10	-1.0000	0.9999	-0.9999	-0.9997	0.9846	0.9998	1.0000
	11	-0.9999	0.9998	-0.9996	-0.9998	0.9855	1.0000	1.0000
Batch Method		-1.0000	1.0000	-0.9999	-1.0000	0.9978	0.9996	0.9999

in this simulation require 201.4500 seconds of CPU time, which is greater than the total CPU time of an online subblock learning method. Not only the online subblock learning method requires lower CPU time for demixing super-Gaussianity, but also it is of lower CPU time for separating sub-Gaussianity, as shown in Table 4.5.

Experiment 2: multi-distributed Mixtures

In this experiment, the source signals are mixtures between super-Gaussian and sub-Gaussian distributions. The super-guassian data sets consist of four sound sources taken from http://speech.kaist.ac.kr/~jangbal/ch1bss. For sub-Gaussianity, we simulated our algorithm using the three synthetic signals given in Section 4.2.3. Each channel contained 46,560 data points. Similar to the previous section, the optimal result is obtained when the subblock size is greater than or equal to $4,096$ data points. In this simulation, the blind source extraction method was first used in the learning methodology. It can be observed that the blind source extraction method converges to the local minimum

Table 4.2: The Similarity Measure using Extended Infomax function for an online subblock learning and batch learning methods based on uni-distributed mixtures.

Learning Methods	Subblock Number	Sources						
		sup_1	sup_2	sup_3	sup_4	sub_1	sub_2	sub_3
Online Subblock Method	1	-0.9999	1.0000	-0.9983	-0.9997	0.9959	0.9998	0.9999
	2	-0.9999	0.9999	-0.9978	-0.9995	0.9970	0.9995	0.9997
	3	-0.9999	1.0000	-0.9999	-0.9995	0.9980	0.9991	1.0000
	4	-0.9999	1.0000	-0.9999	-0.9999	1.0000	0.9993	0.9998
	5	-1.0000	0.9999	-0.9999	-0.9999	0.9998	0.9989	0.9999
	6	-1.0000	1.0000	-0.9999	-0.9993	0.9974	0.9983	0.9998
	7	-1.0000	0.9999	-0.9999	-0.9995	0.9927	0.9978	0.9998
	8	-0.9999	0.9998	-0.9998	-0.9999	0.9880	0.9985	0.9999
	9	-1.0000	1.0000	-0.9985	-0.9994	0.9799	0.9991	0.9999
	10	-0.9999	0.9998	-0.9998	-0.9998	0.9815	0.9995	1.0000
	11	-1.0000	0.9999	-0.9998	-0.9999	0.9824	0.9999	1.0000
Batch Method		-1.0000	1.0000	-0.9995	-0.9992	0.9971	0.9992	0.9997

better than the parallel blind source separation if the non-identically and independently distributed sources are mixed.

As known from the previous simulation, the optimal hyperbolic-Cauchy function for separating the observed signals in this paper is $\phi_i(y_i) = \tanh(4y_i)$. Hence, the results in this simulation are based on the Cichocki [18], and *Extended Infomax* [39] functions, and our low computational functions (**LF-ICA**)[13] with respect to $\tanh(4y_i)$. Similar to the experiment on the uni-distributed mixtures, all activation functions from Section 3.3 are able to recover the source signals from the observed signals. Figure 4.6 shows the source, the mixed, and the recovered signals using the activation functions described in Sections 3.3.1 and 3.3.2. The signals s_1 to s_4 are super-Gaussian distributions. The remaining signals are sub-Gaussian distributions.

Table 4.6 shows the similarity measure between the sources and the recovered signals when two source distributions are mixed. Table 4.7 shows the Kurtosis of the

Table 4.3: The Similarity Measure using **LF-ICA** function for an online subblock learning and batch learning methods based on uni-distributed mixtures.

Learning Methods	Subblock Number	Sources						
		sup_1	sup_2	sup_3	sup_4	sub_1	sub_2	sub_3
Online Subblock Method	1	-1.0000	1.0000	-0.9987	-0.9996	0.9956	0.9997	0.9999
	2	-1.0000	0.9999	-0.9970	-0.9999	0.9961	0.9995	0.9999
	3	-0.9999	1.0000	-0.9999	-0.9995	0.9975	0.9990	1.0000
	4	-1.0000	1.0000	-0.9937	-0.9962	0.9998	0.9992	0.9999
	5	-1.0000	1.0000	-0.9997	-0.9995	1.0000	0.9988	0.9999
	6	-0.9997	1.0000	-0.9997	-0.9987	0.9983	0.9982	0.9999
	7	-1.0000	1.0000	-0.9999	-0.9988	0.9942	0.9975	0.9999
	8	-1.0000	0.9998	-0.9996	-0.9996	0.9895	0.9982	1.0000
	9	-1.0000	1.0000	-0.9986	-0.9988	0.9821	0.9990	0.9999
	10	-1.0000	1.0000	-0.9999	-0.9996	0.9826	0.9995	1.0000
	11	-0.9999	0.9998	-0.9993	-0.9998	0.9829	0.9999	1.0000
Batch Method		-1.0000	1.0000	-0.9999	-1.0000	0.9979	0.9990	0.9999

seven source signals, observed signals, prewhitened observed signals and recovered signals which produced by the Cichocki's function [18], *Extended Infomax* function [39], and our low complexity functions [13]. Before the linear transformation done by the mixing matrix $\mathbf{B}$, the super-Gaussian and the sub-Gaussian source signals have positive and negative Kurtosis values. After the linear transformation, they become positive Kurtosis values, as shown in column 3. After decorrelation process, $\tilde{\mathbf{x}}(k) = diag(\frac{1}{\sqrt{d_1}}, \ldots, \frac{1}{\sqrt{d_m}})\mathbf{V}_x^T\mathbf{x}(k)$, the Kurtosis sign of each channel is recovered. Then, we can extract the prewhitened signals into two subgroups which are the positive and the negative Kurtosis signs. Next, the positive Kurtosis signed channels are fed into an online subblock learning method which were proposed in Section 3.4. After all online subblock of positive Kurtosis signs are performed, the negative Kurtosis signed channels are fed into the separating networks. The simulation results confirm that the prewhitened Kurtosis signed channel is similar to the recovered Kurtosis signed channel.

Table 4.4: The CPU time usage for online subblock learning and batch learning methods for super-Gaussianity.

Learning Methods	subblock number	CPU time usage (sec)		
		Cichocki	Infomax	**LF-ICA**
Online subblock learning	1	40.1680	28.8310	34.6500
	2	32.2660	10.8360	28.2100
	3	16.1730	9.2940	14.2700
	4	13.7300	4.7370	13.1790
	5	13.5500	6.1390	12.3880
	6	13.7690	5.2980	12.6280
	7	18.4570	6.3990	16.1030
	8	13.4790	6.4690	13.5290
	9	20.7300	7.8620	19.0780
	10	15.1210	12.2970	13.9600
	11	13.6990	6.8500	12.2880
Total for Online learning		211.1420	105.0120	190.2830
Batch learning		224.3320	511.3150	201.4500

Obviously, if the source and its corresponding recovered channel have the same Kurtosis sign, then, they are similarly distributed.

Figure 4.7 shows the number of iterations (*epochs*) per each coming online subblock. The first 11 subblocks are the epochs for demixing positive Kurtosis signs. The remaining subblocks are for demixing negative Kurtosis signs. Normally, a natural sound signal is super-Gaussian distributed. As known, the natural sound is composed of multiples of frequency F_0. The components for the super-Gaussian distribution are more complicated than for sub-Gaussian distribution. Hence, the super-Gaussianity is more computational intensive than the sub-Gaussianity, both in terms of epochs and CPU time usage. Considering the Cichocki function and the **LF-ICA** function for positive Kurtosis sign separation, our proposed function requires higher number of epochs than the Cichocki function, see Figure 4.7. But our proposed function is of lower CPU time than the Cichocki function, see the first 11 subblocks in Figure 4.8. An *Extended Infomax*

Table 4.5: The CPU time usage for online subblock learning and batch learning methods for sub-Gaussianity.

Learning Methods	subblock number	CPU time usage (sec)		
		Cichocki	Infomax	**LF-ICA**
Online subblock learning	1	1.1620	3.5850	2.4640
	2	0.7410	1.0210	1.3510
	3	0.5710	1.8230	0.9910
	4	0.5310	1.4320	1.1620
	5	0.4600	1.4720	0.7810
	6	0.6510	1.9120	1.2020
	7	0.7010	2.4940	1.2420
	8	0.5810	1.0110	0.9910
	9	0.5500	1.4820	1.0610
	10	0.3500	1.0620	0.6610
	11	0.5210	1.1620	0.9120
Total for Online learning		6.8190	18.4560	12.8180
Batch learning		24.5560	22.3220	18.7470

function is of the lowest computational complexity for demixing positive Kurtosis signed channels. But the performance correlation index in Figure 4.9 shows that the correlation index values from an *Extended Infomax* function will not approach zero. It means that their results, y_i and y_j, are much more correlated among the recovered channels than the other two functions. In contrast, for demixing sub-Gaussianity, all functions produce low correlated recovered channels because the component of the sub-gaussianity is of lower complexity. For example, $\mathbf{s}_1(t) = 0.1\sin(400t)cos(30t)$ from equation (4.2) is composed of $F_0 = 400$ and $F_0 = 30$ Hz. Hence, the sub-Gaussianity requires fewer separating time operation than the super-Gaussianity. In this case, the *Extended Infomax* function requires much more computational complexity than Cichocki and **LF-ICA** functions.

Figure 4.9 and 4.10 illustrate the performance correlation index, during the learning process, using our activation functions described in Sections 3.3.1 and 3.3.2, and the existing activation functions for the non-Gaussian mixtures from Section 2.9. Figure 4.9

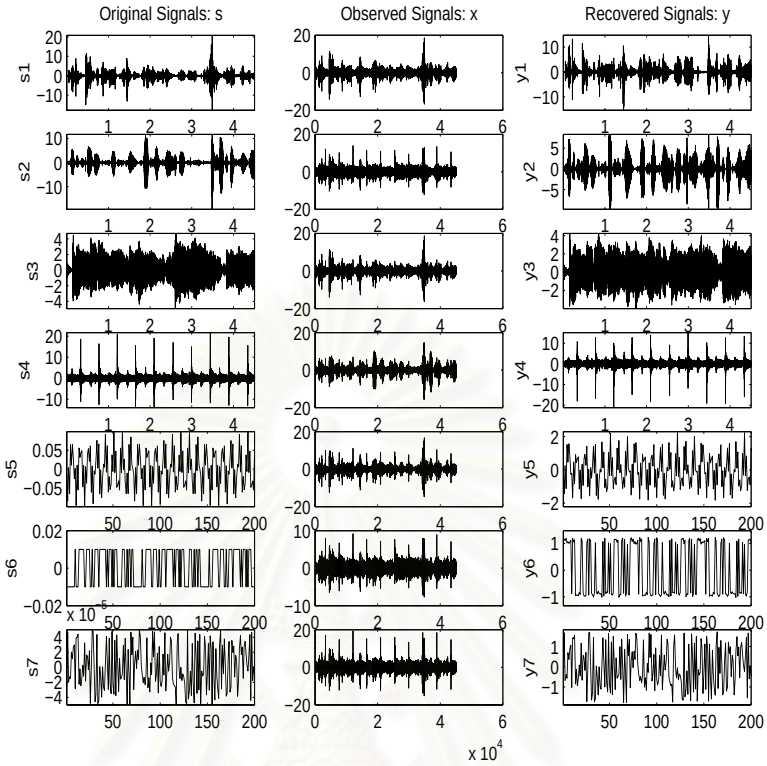


Figure 4.6: The source, the mixed and the recovered signals of the sub-Gaussian and super-Gaussian distributions.

corresponds to the mixture of super-Gaussian signals and Figure 4.10 corresponds to the mixture of sub-Gaussian distributions. Three types of lines are used to denote the performance correlation index results. The thick line is for the Cichocki's function [18]. The dotted line is for an *Extended Infomax* function, presented by Lee *et al* [39]. The dashed line is for **LF-ICA** [13]. Each graphical line for super-Gaussian separation from Figure 4.9 has 11 peaks, which is the same number as the number of the online subblocks. For each learning subblock, the performance correlation index approaches zero when the output signals y_i and y_j become independent. The next subblock will be fed into the blind separating stage, after the output signals in the current subblock become mutually independent. The performance correlation index is obviously increased in the first epoch for every new subblock. Considering the mutual independent degree among the three methods used in our simulation, it can be concluded that the Cichocki's function [18]

Table 4.6: The Similarity Measure using Cichocki, *Extended Infomax* and our low computational function **LF-ICA** for multi-distributed mixtures.

	Online Subblock Learning			Batch Learning		
types	Cichocki	Infomax	**LF-ICA**	Cichocki	Infomax	**LF-ICA**
Sup_1	0.8841	-0.9423	-0.9733	-0.9996	0.9967	-0.9996
Sup_2	-0.9989	-0.9482	-0.9454	-0.9995	0.9968	-0.9995
Sup_3	-0.9579	0.9667	0.9532	0.9998	0.9996	0.9998
Sup_4	-0.9396	0.9413	0.9190	0.9998	-0.9996	-0.9997
Sub_1	0.9927	-0.9913	-0.9921	0.9933	-0.9868	-0.9904
Sub_2	-0.9993	-0.9988	-0.9988	-0.9960	-0.9906	-0.9936
Sub_3	0.9999	-0.9996	-0.9997	0.9999	0.9998	0.9998

and the **LF-ICA** [13] have higher degree of independence than the *Extended Infomax* function [39].

4.3 Analytical Considerations on Complexity

For the mixture of the super-Gaussian signals, the unknown source signals can be recovered by $\tanh(\alpha \mathbf{y})$ and its approximation, as given in equation (3.16), and the *Extended Infomax* function. Considering the same input vector, both $\tanh(\alpha \mathbf{y})$ and its approximation activation functions produce a similar output vector because the curve of the approximation was matched to the curve of $\tanh(\alpha \mathbf{y})$, as illustrated in Figure 3.3. Hence, they required the same number of *epochs* for recovering the source signals, as shown in the first eleven subblocks in Figure 4.7. But an approximation function requires fewer computational micro-operations per instruction than $\tanh(\alpha \mathbf{y})$, and it is more suitable for hardware implementation. Moreover, an approximation function of $\tanh(\alpha \mathbf{y})$ requires lower CPU time than $\tanh(\alpha \mathbf{y})$ as shown in Figure 4.8. Comparing the number of iterations per each subblock between the *Extended Infomax* function and the MI function, the *Extended Infomax* function requires lower number of *epochs* than

Table 4.7: The Kurtosis of the seven source signals and the Kurtosis of the recovered signals via Cichocki, *Extended Infomax* and our low computational function **LF-ICA**.

Source types	Source Kurtosis	Observed Kurtosis	Prewhitened Kurtosis	Recovered Kurtosis		
				Cichocki	Infomax	**LF-ICA**
Sup_1	$3.91e+01$	$1.66e+01$	$1.42e+01$	$1.58e+01$	$1.60e+01$	$1.81e+01$
Sup_2	$3.06e+01$	$4.99e+01$	$1.78e+01$	$1.56e+01$	$1.55e+01$	$1.32e+01$
Sup_3	$7.87e-01$	$1.15e+01$	$8.28e+00$	$2.62e-01$	$2.58e-01$	$2.62e-01$
Sup_4	$3.83e+01$	$8.10e+00$	$1.09e+01$	$3.53e+01$	$3.37e+01$	$3.50e+01$
Sub_1	$-4.69e-06$	$1.09e+01$	$-1.20e+00$	$-6.18e-01$	$-5.62e-01$	$-5.94e-01$
Sub_2	$-2.00e-08$	$7.08e+01$	$-1.18e+00$	$-1.95e+00$	$-1.90e+00$	$-1.93e+00$
Sub_3	$-8.53e-19$	$1.27e+01$	$-2.78e-01$	$-1.20e+00$	$-1.20e+00$	$-1.20e+00$

the MI. But the *Extended Infomax* function produces the correlated output as illustrated in Figure 4.9. The recovered signals from the *Extended Infomax* function have higher degree of correlation than the MI function.

Regarding the recovery of the mixture of more sub-Gaussian signals, the curves of $\phi(\mathbf{y}) = \pm\mathbf{y}^2$ did not exactly match either $\mathbf{y}^3$ or $\mathbf{y}^{11}$, but they produced the same results with higher convergent speed as shown in Figure 4.10. The lower activation function needs smaller memory representation during the running process. And, also, the $\phi(\mathbf{y}) = \pm\mathbf{y}^2$ requires only "Shift-and-Add" micro-operations per instruction and no multiplication operation.

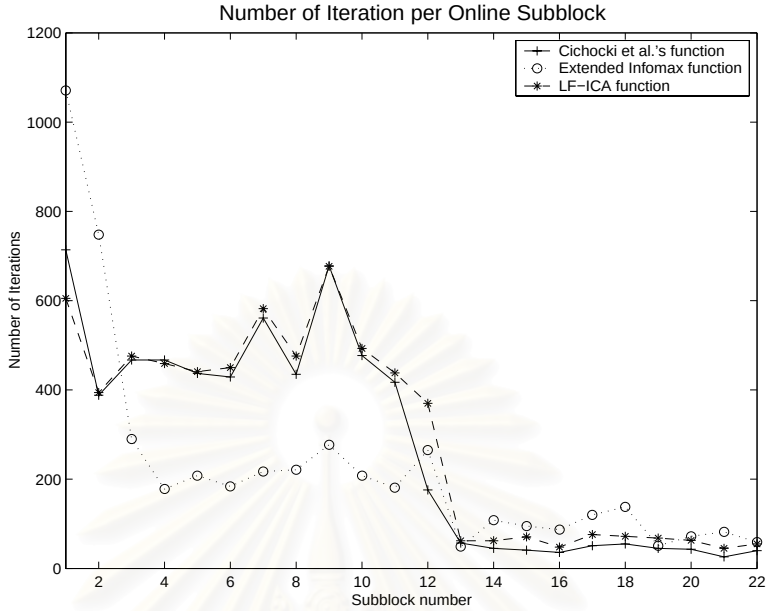


Figure 4.7: Number of iterations per online subblock. The first 11 subblocks are the computational complexity for demixing super-Gaussian distribution. The remaining subblocks are for demixing sub-Gaussian distribution.

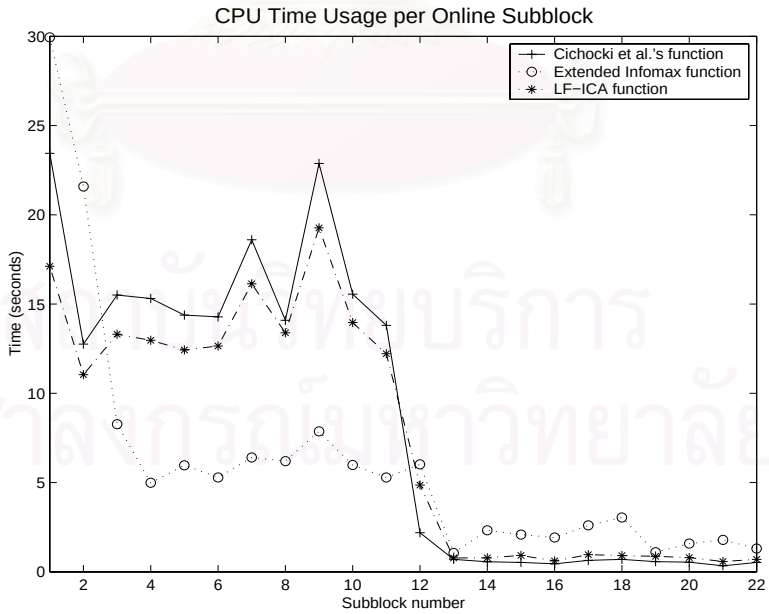


Figure 4.8: CPU time usage per online subblock. The first 11 subblocks are the computational complexity for demixing super-Gaussian distribution. The remaining subblocks are for demixing sub-Gaussian distribution.

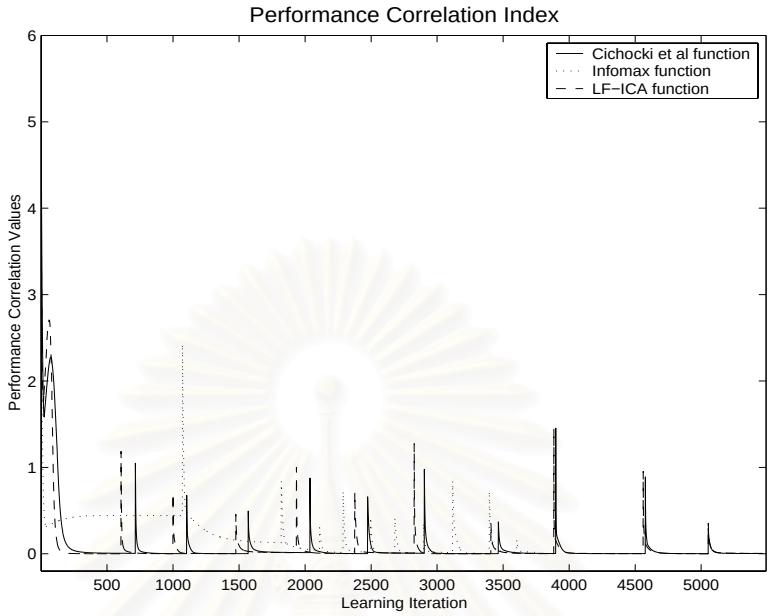


Figure 4.9: Performance correlation index for the sequential source separation for demixing super-Gaussianity.

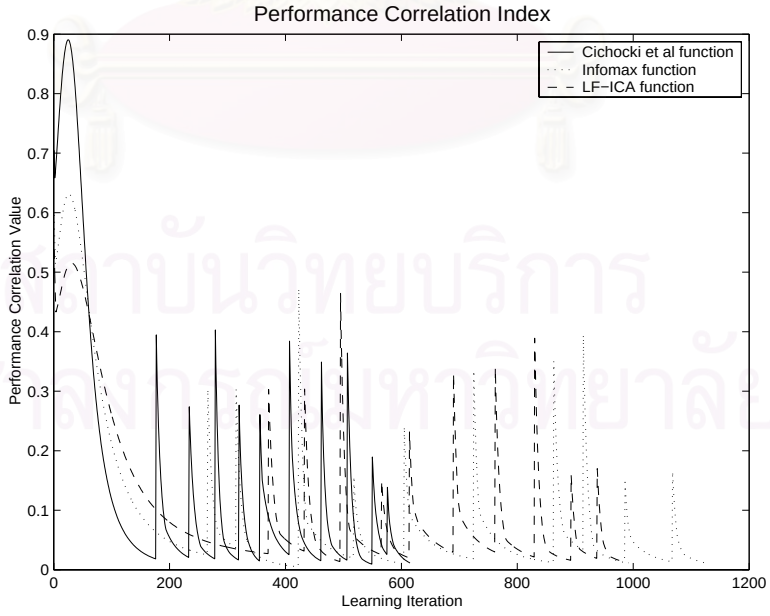


Figure 4.10: Performance correlation index for the sequential source separation for demixing sub-Gaussianity.

CHAPTER V

Conclusions

The main problem of ICA or BSS concerns the unknown information of source and mixing matrix $\mathbf{B}$. We try to solve the inversion matrix of $\mathbf{B}$, called demixing matrix $\mathbf{W}$ here. The convergence test is measured by Kullback-Liebler divergence. In this dissertation, four optimization techniques for independent component analysis problem are proposed, which include (1) increasing the learning convergence using the momentum term and an appropriate learning rate divisor, (2) deriving some low computational activation functions, (3) proposing blind source extraction and, (4) presenting an unsupervised learning method for finding an optimal online subblock size.

First, we study an effect of learning parameter for independent component analysis. We found that the momentum term and an appropriate divisor for each learning rate value are significant factors for the ICA learning mechanism. Figure 4.2 shows that the fixed learning rate value will diverge if the learning rate is too high. The result is better than the fixed learning rate value if we divide all learning rate values with 1.005. For better results, we look for an optimal divisor for each learning value. For example, if η is 0.8, its optimal divisor is 1.008. Moreover, the convergent speed is increased when the momentum term is added in the learning equation.

Second, some low complexity activation functions are presented in order to reduce computational time per instruction and implement the functions on a circuit level. We obtain two quadratic functions for demixing super-Gaussian and sub-Gaussian channels. It is confirmed by the computer simulation that our proposed activation functions

produce the outputs as good as those from the existing high order activation functions. The recoverd signals from our proposed functions are not exactly similar to the results from the exising functions because our functions are only approximated functions. But the proposed activation functions are of lower computational time per instruction than the exising functions. They require only "shift-and-add" operations per instruction and they are feasible to implement on the VLSI level.

Third, a sequential blind source separation or blind source extraction for an independent component analysis for the non-Gaussian mixtures using a two layered neural network is presented in this dissertation. This method is, mainly, to avoid a weak point of the mutual information learning criteria. Mutual Information learning works well for identical and independent distributions. For non-identical sources, MI diverges to a local minima. After the prewhitening process, the prewhitened signals have been separated into two subgroups of positive and negative Kurtosis signs. This property is used to extract prewhitened signals. We, first, operate on the positive Kurtosis or super-Gaussian distribution. Then, we follow with the negative Kurtosis or sub-Gaussian distribution. Experimantal results claim that blind source extraction is better than the typical blind source separation when the sources are non-identically distributed. It is observed that in some cases more than one Gaussian noises are in the system, the Kurtosis objective functions are unsuitable. The reason behind this observation needs further investigation.

Forth, in case that we want to implement the previously proposed method at the VLSI level or on chip. A possible subblock size of the input must be known in advance. But in the assumption of the ICA problem, we have no information about the sources and the mixtures. This dissertation proposed a novel technique for finding an feasible online subblock for the ICA problem based on the observation on the eigenvalue of the prewhitened signals when the input size is changed and found that subblock size $k = 4096$ is suitable for the current data set.

Finally, it can be said that a momentum term and an appropriate divisor for each learning rate, proposed activation functions, online sub-block learning algorithm, and sequential blind source separation are efficient methods for demixing the non-Gaussian mixtures, with respect to the convergence speed and learning abilities. The blind source separation problem may have the following further studies.

1. Source signals have more than two Gaussian distributions.
2. Sources, observed and recovered signals are non zero mean and non unit variance.
3. Number of sensors is less than number of sources, $m \leq n$.
4. Observation is on only one channel.
5. Ensemble learning system for ICA problem.
6. Applications of ICA problem.

References

[1] K.Abed-Meraim, Yong Xiang, J.H.Manton and Y.Hua. Blind Source Separation Using Second-Order Cyclostationary Statistics *IEEE Transections on Signal Processing*, Vol. 49, No. 4, pp. 694-701, 2001.

[2] S.-I.Amari, A.Cichocki, and H.H.Yang. A New Learning Algorithm for Blind Signal Separation, *MIT Press*, pp.757-763, 1996.

[3] S.-I.Amari. Natural Gradient Works Efficiently in Learning, *Neural Computation*, Vol. 10, pp. 251-276, 1998.

[4] H.Attias and C.E.Schreiner. Blind Source Separation and Deconvolution: The Dynamic Component Analysis Algorithm, *Neural Computation*, Vol. 10, pp. 1373-1424, 1998.

[5] A.D.Back and T.P.Trappenberg. Selecting Inputs for Modeling Using Mormalized Higher Order Satistics and Independent Component Analysis *IEEE Transections on Neural Networks*, Vol. 12, No. 3, pp. 612-617, 2001.

[6] S.Barnett, *Matrix Methods for Engineers and Scientists*, McGRAW-Hill Book Company (UK) Limited, 1979.

[7] A.J.Bell and T.J.Sejnowski. An information-maximization approach to blind separation and blind deconvolution, *Neural Computation*, Vol. 7, pp. 1004-1034, 1995.

[8] J.-F.Cardoso. Infomax and Maximum Likelihood for Blind Source Separation, *IEEE Signal Processing Letters*, Vol. 4, No. 4, pp. 112-114, 1997.

[9] J.-F.Cardoso. Blind Signal Separation: Statistical Principles, *Proceedings of IEEE*, Vol. 86, No. 10, pp. 2009-2025, Oct 1998.

[10] D.Charles. Constrained PCA techniques for identification of common factors in data, *Neurocomputing*, Vol. 22, pp. 145-156, 1998.

[11] T.Chen, S.-I.Amari and Q.Lin. A Unified Algorithm for Pricipal an Minor Component Extraction *Neural Networks*, Vol. 11, pp. 385-390, 1998.

[12] K.Chinnasarn and C.Lursinsap. Effects of Learning Parameters on Independent Component Analysis Learning Procedure, *Proceedings of the* 2^{nd} *International Conference on Intelligent Technologies*, pp. 312-316, 2001.

[13] K.Chinnasarn, C.Lursinsap, and V.Palade. Low Complexity functions for Stationary Independent Component Mixtures, *Proceedings of the* 7^{nd} *Knowledge-Based Intelligent Information & Engineering Systems*, 2003.

[14] K.Chinnasarn, C.Lursinsap, and V.Palade. Sequential Source Separation for Mixed Kurtosis Sign Sources, *Proceeding of the* 3^{rd} *International Symposium on Communications and Information Technologies (ISCIT2003)*, Songkla, Thailand, 530-535, 2003.

[15] K.Chinnasarn, C.Lursinsap, and V.Palade. Blind Extraction of Mixed Kurtosis Signed Signals Using Partial Observations and Low Complexity Activation Functions, accepted for publication to *the International Journal of Computational Intelligence and Applications*, in a special issue on "Computational Intelligence for Signal and Image Processing".

[16] A.Cichicki, S.C.Douglas, and S.-I.Amari. Robust techniques for independent component analysis (ICA) with noise data, *Neurocomputing*, Vol. 22, pp. 113-129, 1998.

[17] A.Cichocki, J.Karhunen, W.Kasprzak and R.Vigario. Neural Networks for blind separation with unknown number of sources, *Neurocomputing*, Vol. 24, pp. 55-93, 1999.

[18] A.Cichocki and S.-I.Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, John Wiley & Sons, Ltd., 2002.

[19] P.Comon. Independent Component Analysis: A New Concept?, *Signal Processing*, Vol. 36. pp. 287-314, 1994.

[20] H.Dai and C.Macbeth. Effects of Learning Parameters on Learning Procedure and Performance of a BPNN, *Neural Networks*, vol.10, No.8, pp.1505-1521, 1997.

[21] S.C.Douglas, T.-P.Chen, and A.Cichocki. Multichannel blind separation and deconvolution of sources with arbitary distributions, in, *Proc NNSP*, Amelia Island, FL, Sep. 1997, pp.436-445.

[22] S.C.Douglas and A.Cichocki. Neural Networks for Blind Deconvolution of Signals, in, *IEEE Transections on Signal Processing in Sepcial Issue on Neural Networks for Signal Processing*, 1997.

[23] S.C.Douglas. Self-Stabilized Gradient Algorithms for Blind Source Separation with Orthogonality Constraints, *IEEE Transections on Neural Networks*, Vol. 11, No. 6, pp. 1490-1497, 2000.

[24] R.O.Duda, P.E.Hart and D.G.Stork. *Pattern Classification*, 2^nd edition, John Wiley & Sons, Inc., 2001.

[25] S.Fiori. Blind Signal Processing by the Adaptive Activation Function Neurons, *Neural Networks*, Vol. 13, pp. 597-611, 2000.

[26] S.Haykin. *Neural Network a Comprehensive Foundation*, 2^nd, Prentice Hall, 1999.

[27] A.Hyvarinen and E.Oja. A Neuron that Learns to Separate One Signal from a Mixture of Independent Sources, *IEEE International Conference on Neural Networks*, Vol. 1, pp. 62-67, 1996.

[28] A.Hyvarinen. Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood, *Neurocomputing*, Vol. 22, pp. 49-67, 1998.

[29] A.Hyvarinen and E.Oja. Independent Component analysis: algorithms and applications, *Neural Networks*. Vol. 13 pp. 411-430, 2000.

[30] A.Hyvarinen. Blind Source Separation By Nonstationary of Variance: A Cumulant-Based Approach, *IEEE Transections on Neural Networks*, Vol. 12, No. 6, pp. 1471-1474, 2001.

[31] S.Ikeda and K.Toyama. Independent Component Analysis for Noisy Data–MEG Data Analysis, *Neural Networks*, No. 13, pp. 1063-1074, 2000.

[32] M.Joho, H.Mathis and G.S.Moschytz. Combined Blind/Nonblind Source Separation Based on Natural Gradient, *IEEE Signal Processing Letters*, Vol. 8, No. 8, pp. 236-238, 2001.

[33] J.Karhunen, E.Oja, L.Wang, R.Vigario and J.Joutsensalo. A Class of Neural Networks for Independent Component Analysis, *IEEE Transections on Neural Networks*, Vol. 8, No. 3, pp. 486-504, 1997.

[34] J.Karhunen, P.Pajunen, and E.Oja. The nonlinear PCA criterion in blind source separation: Relations with other approaches, *Neurocomputing*, Vol. 22, pp. 5-20, 1998.

[35] M.Kawamoto, K.Matsuoka and N.Ohnishi. A method of blind separation for convolved non-stationary signals, *Neurocomputing*, Vol. 22, pp. 157-171, 1998.

[36] H.K.Kwan. Simple Sigmoid-like activation function suitable for digital hardware implementation, *Electronics Letter* Vol.28, no.15, pp.1379-1380, 1992.

[37] R.J.Larsen and M.L.Mark. *An Introduction to Mathematical Statistics and Its Applications*, 2^{nd}, Prentice Hall, 1986.

[38] T.-W.Lee, M.Girolami, A.J.Bell and T.J.Sejnowski. A Unifying Information-Theoretic Framework for Independent Component Analysis *International Journal on Mathematical and Computer Models*, 1998.

[39] T.-W.Lee, M.Girolami and T.J.Sejnowski. Independent Component Analysis Using an Extended Informax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources *Neural Computation*, Vol. 11, No. 2, pp. 409-433, 1999.

[40] T.-W.Lee, M.S.Lewicki and T.J.Sejnowski. ICA Mixture Models for Unsupervised Classification on Non-Gaussian Classes and Automatic Context Switching in Blind Signal separation, *IEEE Transactions on Pattern analysis and Machine Intelligence*, Vol. 22, No.10, 2000.

[41] C.F.V.Loan. *Introduction to Scientific Compuiting: A Matrix-Vector Approach Using MATLAB*, Prentice-Hall Inc, 2000.

[42] A.Mansour and C.Jutten. Fourth-Order Criteria for Blind Source Separation *IEEE Transections on Signal Processing*, Vol. 43, No. 8, pp. 2022-2025, 1995.

[43] H.Mathis, T.P.V.Hoff and M.Joho. Blind Separation of Signals with Mixed Kurtosis Signa Using Threshold Activation Functions, *IEEE Transections on Neural Networks*, Vol. 12, No. 3, pp. 618-624, 2001.

[44] H.Mathis and S.C.Douglas. On the Existence of Universal Nonlinearities for Blind Source Separation, *IEEE Transections on Signal Processing*, Vol. 50, No. 5, pp. 1007-1016, 2002.

[45] M.Ohata and K.Matsuoka. Stability Analyses of Information-Theoretic Blind Separation Algorithms in the Case Where the Sources Are Nonlinear Processes, *IEEE Transections on Signal Processing*, Vol. 50, No. 1, pp. 69-77, 2002.

[46] E.Oja. From neural learning to independent component, *Neurocomputing*, Vol. 22, pp. 187-199, 1998.

[47] P.Pajunen. Blind source separation using algorithmic information theory, *Neurocomputing*, Vol. 22, pp. 35-48, 1998.

[48] H.Park, S.-I.Amari and K.Fukumizu. Adaptive Natural Gradient Learning Algorithms for Various Stochastic Models *Neural Networks*, No. 13, pp. 755-764, 2000.

[49] T.Phiasai, S.Arunrungrusmi and K.Chamnongthai. Face Recognition System with PCA and Moment Invariant Method, *The IEEE International Symposium on Circuits and Systems: ISCAS 2001*, Vol: 2 , pp.165 -168, 2001.

[50] D.N.Politis, Computer-Intensive Methods in Statistical Analysis *IEEE Signal Processing Magazine*, pp. 39-55, Jan 1998.

[51] M.-O.Pun. *A Simple variable step algorithm for blind source separation(BSS)*, Master Thesis, University of Tsukuba. 1999.

[52] S.Robert and R.Everson. *Independent Component Analysis: Priniples and Practice*, Cambridge University Press, 2001.

[53] Y.Tan, J.Wang and J.M.Zurada. Nonlinear Blind Source Separation Using a Radial Basis Function Network, *IEEE Transections on Neural Networks*, Vol. 12, No. 1, pp. 124-134, 2001.

[54] Y.Tan and J.Wang. Nonlinear Blind Source Separation Using Higher Order Statistics and Genetic Algorithm, *IEEE Transections on Evolutionary Computation*, Vol. 5, No. 6, pp. 600-612, 2001.

[55] J.-M.Wu and S.J.Chiu. Independent Component Analysis Using Potts Models, *IEEE Transections on Neural Networks*, Vol. 12, No. 2, pp. 202-211, 2001.

[56] Y.Wu,K.-W.Tam and F.Li. Determination of Number of Sources With Multiple Arrays in Correlated Noise Fields, *IEEE Transections on Signal Processing*, Vol. 50, No. 6, pp. 1257-1260, 2002.

[57] L.Xu, C.-C.Cheung. and S.-I.Amari. Learned parametric mixture based ICA algorithm, *Neurocomputing*, Vol. 22, pp. 69-80, 1998.

[58] X.-H.Yu. and G.-A.Chen. Efficient Backpropagation Learning Using Optimal Learning Rate and Momentum, *Neural Networks*, Vol.10, No.3, pp.517-527, 1997.

[59] V.Zarzoso, A.K.Nandi, F.Herrmann and J.Millet-Roig. Combined Estimation Scheme for Blind Source Separation with Arbitary Source PDFs, *Electronics Letters*, Vol. 37, No. 2, pp. 132-133, 18^{th} January 2001.

[60] A.M.Zoubir and B.Boashash. The Bootstrap and Its Application in Signal Processing, *IEEE Signal Processing Magazine*, pp. 56-76, Jan 1998.

Biography

Name: Mr.Krisana CHINNASARN.
Date of Birth: 25^{th} June, 1970.
Educations:

- Ph.D., Program in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand, (June 2000 - April 2004)
- Ph.D. Visiting student, Oxford University Computing Laboratory, Oxford, UNITED KINGDOM, (October 2002 - June 2003).
- M.Sc. Program in Computer Science and Information Technology, King Mongkut's Institute of Technology Ladkrabang, Thailand. (June 1994 - November 1997).
- B.Sc. Program in Statistics Srinakarinwirot University, Mahasarakham, Thailand. (June 1989 - March 1992).

Publication papers:

- K.Chinnasarn and C.Lursinsap. Effects of Learning Parameters on Independent Component Analysis Learning Procedure, *Proceedings of the* 2^{nd} *International Conference on Intelligent Technologies*, pp. 312-316, 2001.
- K.Chinnasarn, C.Lursinsap, and V.Palade. Low Complexity function for Stationary Independent Component Mixtures, *Proceedings of the* 7^{nd} *Knowledge-Based Intelligent Information and Engineering Systems*, 2003.
- K.Chinnasarn, C.Lursinsap, and V.Palade. Sequential Source Separation for Mixed Kurtosis Sign Sources, *Proceeding of the* 3^{rd} *International Symposium on Communications and Information Technologies (ISCIT2003)*, Songkla, Thailand, 530-535, 2003.
- K.Chinnasarn, C.Lursinsap, and V.Palade. Blind Extraction of Mixed Kurtosis Signed Signals Using Partial Observations and Low Complexity Activation Functions, accepted for publication to *the International Journal of Computational Intelligence and Applications*, in a special issue on "Computational Intelligence for Signal and Image Processing".

Scholarship: Thai Government.