Multivariate Analysis to Explore Latent Structure by Minimum Beta-Divergence Method

Md Nurul Haque Mollah

DOCTOR OF PHILOSOPHY

Department of Statistical Science
School of Multidisciplinary Science
Graduate University for Advanced Studies,
JAPAN

2005 (School Year)

Contents

Abstract

This thesis deal with multivariate analysis, especially robust prewhitening for independent component analysis (ICA), exploring local PCA structures and extraction of local ICA structures by the minimum β-divergence method. In the context of minimum β-divergence method, the β-divergence between the empirical distribution of a sample (data distribution) and the specific distribution corresponding to the problem under study is minimized with respect to the parameters to be estimated (Minami and Eguchi, 2002; Mollah, Minami and Eguchi, 2006). The minimum β-divergence method with $\beta = 0$ reduces to the minimum Kullback-Leibler (K-L) divergence method. The minimum β-divergence method is robust against outliers. If a data set contains more than one data cluster, then minimum β-divergence method works on a cluster considering other clusters as outliers. Sequentially, it works in each cluster changing the initial value of the shifting parameter by the minimizer of the cumulative weight (Mollah, Minami and Eguchi, 2006; Mollah, Sultana, Minami and Eguchi, 2005b). In this thesis, we propose three main results obtain by the minimum β-divergence method.

1. An adaptive robust prewhitening procedure for ICA is proposed by minimizing β-divergence.

2. An adaptive algorithm to explore local PCA structures for dimensionality reduction is proposed by minimum β-Divergence method.

3. An extention of minimum β-divergence method (Minami and Eguchi, 2002) is proposed for exploring local ICA structures.

An adaptive robust prewhitening procedure named β-prewhitening is proposed by minimizing the empirical β-divergence over the space of all the Gaussian distributions. The performance of this new prewhitening is compared with the classical prewhitening by a performance index (newly proposed) and FastICA (Hyvärinen, 1999) using both synthetic and

real data sets. Simulation result shows that β-prewhitening efficiently improves the performance over the classical prewhitening when outliers exist; it reduce to classical prewhitening otherwise.

A comparatively new problem in multivariate analysis is to explore local PCA or ICA structures. An attempt is made to propose a new learning algorithm to explore local PCA structures in which observed data consists of several data clusters. The proposed method is based on a sequential application of the minimum β-divergence method to search an orthogonal matrix for each cluster sequentially. The proposed method searches local PCA structures sequentially on the basis of a rule of sequential change of the shifting parameter and a local kernel vector. If the initial choice of the shifting parameter vector and the local kernel vector belongs to a data cluster, then all data belonging to that cluster are transformed into a PCA structure considering the data in other clusters as outliers. The value of the kernel parameter ν plays a key in the performance of the proposed methods mentioned above. A cross-validation technique is proposed as an adaptive selection procedure for the tuning parameter ν.

This thesis also discusses a learning algorithm to explore local ICA structures in which observed data consists of several data clusters. The proposed method is based on a sequential application of the minimum β-divergence method to separate all source classes sequentially. The proposed method searches the recovering matrix of each class on the basis of a rule of sequential change of the shifting parameter. If the initial choice of the shifting parameter vector is close to the mean of a data class, then all of the hidden sources belonging to that class are recovered properly with independent and non-Gaussian structure considering the data in other classes as outliers. The value of the tuning parameter β is a key in the performance of the proposed methods. A cross-validation technique is proposed as an adaptive selection procedure for the tuning parameter β.

Chapter 1

Introduction

A common problem encountered in such disciplines as statistics, signal processing, and neural network research, is finding a suitable representation of multivariate data. For computational and conceptual simplicity, such a representation is often sought as a linear transformation of the original data. Well-known linear transformation methods include, for example, principal component analysis, factor analysis, and projection pursuit. A recently developed linear transformation method is independent component analysis (ICA). Their various applications include feature extraction, image processing, dimension reduction, blind source separation (BSS) and so on. Non-linear models for PCA or ICA and the dimension reduction by neural networks were also developed to deal with non-linear data structures. The non-linear approaches, however, have not been so successful due to computational and conceptual complexity. Thus local model approaches, which are used in connection with a suitable clustering algorithm, were considered. Along with the preprocessing with cluster analysis followed by the estimation of the local linear model of each cluster, iterative local PCA algorithms based on the minimization of the reconstruction errors (Kambhatla and Leen, 1997). Another alternative is the probabilistic approach to the local PCA, in which the task of PCA is formulated based on the estimation of probabilistic models (Tipping et al., 1997, 1999) and the models are enhanced to the PCA mixture models that are optimized by using the maximum likelihood techniques. This mixture was modified into a mixture of factor analyzers (FA) (Ghahramani et al., 2000), where variational Bayesian inference was used to infer the optimum number of Analyzers.

One problem with PCA/FA mixture models is that each component is a Gaussian, a strong assumption which is often violated in many natural clustering problems (Lee et al., 2000a,b).

Although mixtures of Gaussian (MoG) are capable of modeling most distributions given enough components, the problem still remains of automatically grouping Gaussians which together describe some larger-scale feature. A solution is reached by extending the mixtures of probabilistic PCA/FA model to a ICA Mixture models. Previous work (Lee et al., 2000b; Penny et al., 2001) is improved by incorporating a very flexible ICA model that can generate arbitrary densities using MoGs, and by bringing the formalism into the Bayesian arena. Bayesian inference is used to infer the optimum number of ICAs needed and automatically determine their ideal dimensionalities. The ICA mixture model has been applied to multi-class problems by using the EM algorithm (Lee et al., 2000a,b).

Many estimation methods for ICA requires prewhitening of observed signals, because it reduces the complexity of the ICA problems (Hyvärinen, Karhunen and Oja, 2001; Cichoki and Amari, 2002). In the case of fixed-point algorithms (Hyvärinen, Karhunen and Oja, 2001), it plays a significant role on the performance of the algorithms. In particular, Hyvärinen (1999) proposed FastICA fixed-point algorithm for robust BSS. However, the performance of this algorithm is not so good sometimes. A main cause of this weak performance may be from the classical prewhitening procedure, which is known to be sensitive to outliers. Thus estimate of independent components under classical prewhitening gives misleading results in presence of outliers or noisy data. There exist some robust prewhitening procedure like batch algorithm based on the subspace approach for ICA (Cichoki and Amari, 2002). However, this type of robust prewhitening may be suffer from the non-robust classical centering, (Hyvärinen et al., 2001, page 154). On the other hand, the performance of classical prewhitening procedure is better than the performance of robust prewhitening procedure for noiseless data sets, while this performance is completely reverse for noisy data sets. It is also difficult task to know in advance whether a data set is noise free or not. Therefore, existing prewhitening procedures are not always suitable. In this thesis, a new prewhitening procedure named β-prewhitening is proposed by minimizing β-divergence from the adaptive robustness point of view.

In both classical PCA or ICA, only one data cluster is considered in the entire data space. However, in some situations, number data clusters may be more than one. Then, classical PCA or ICA gives misleading results. To overcome this problem, Lee, Lewicki and Sejnowski (2000b) proposed ICA mixture models to explore local ICA structures and Tipping et al.

(1999) proposed mixture of probabilistic PCA algorithm to explore local PCA structures. However, there exist one problem in their method is that the number c of classes need to know in advance which is very difficult task in practice. To overcome this inconvenient, this thesis introduces minimum β-divergence method in both local ICA and PCA contexts.

1.0.1 Organization of Thesis

This thesis is organized within eight chapters. Before going to main research we discussed some related areas in chapter 2. Both PCA and ICA are the fundamental areas of research in this thesis. However, ICA is a comparatively new addition in multivariate analysis. As such, chapter 3 is the largest of the theory chapters and serves as an introduction to ICA. This chapter explains the concept of independent components and their analysis by ICA. This is followed by a formulation of the linear ICA model under some assumptions. Then we discuss some preprocessing for ICA that greatly helps solving the ICA problem. Then we discuss basic principles of ICA estimation. Then we discuss some existing ICA algorithms and their problems. At the end of this section we present the main objectives of research.

In chapter 4 we proposed a new adaptive robust preprocessing system by minimizing β-divergence for ICA. Also we investigate the performance of this new proposal by a performance index and FastICA in a comparison of the classical preprocessing. In chapter 5, we proposed a new method for exploring local PCA structures based on minimum β-divergence method. To demonstrate the validity of this method, we present some simulation results using synthetic data set in this chapter also. In chapter 6, an attempt is made to propose a new method for exploring local ICA structures based on minimum β-divergence method. We present some simulation results using artificial and natural signals to demonstrate the validity of the last proposal. Chapter 7 presents some incomplete research directions for future study. Chapter 8 included the concluding remarks.

Chapter 2

Classical Linear Transformations

A central problem in neural network research, as well as in statistics and signal processing, is finding a suitable representation of the data, by means of a suitable transformation. It is important for subsequent analysis of the data, whether it be pattern recognition, data compression, de-noising, visualization or anything else, that the data is represented in a manner that facilitates the analysis. As a trivial example, consider speech recognition by a human being. The task is clearly simpler if the speech is represented as audible sound, and not as a sequence of numbers on a paper.

Let us concentrate on the problem of representing continuous-valued multidimensional variables. Let us denote by $\boldsymbol{x}$ an m-dimensional random variable; the problem is then to find a function f so that the n-dimensional transform $\boldsymbol{s} = (s_1, s_2, ..., s_n)^T$ defined by

$$\boldsymbol{s} = f(\boldsymbol{x}) \tag{2.1}$$

has some desirable properties. In most cases, the representation is sought as a linear transform of the observed variables, i.e.,

$$\boldsymbol{s} = W\boldsymbol{x} \tag{2.2}$$

where W is a matrix to be determined. Using linear transformations makes the problem computationally and conceptually simpler, and facilitates the interpretation of the results. Thus we treat only linear transformations here. Most of the methods described here can be extended for the non-linear case. Such extensions are, however, outside the scope of this thesis. Several principles and methods have been developed to find a suitable linear transformation. These include principal component analysis, factor analysis, projection pursuit,

independent component analysis, and many more. Usually, these methods define a principle that tells which transform is optimal. The optimality may be defined in the sense of optimal dimension reduction, statistical 'interestingness' of the resulting components s_i, simplicity of the transformation W, or other criteria, including application-oriented ones.

Recently, a particular method for finding a linear transformation, called independent component analysis (ICA), has gained wide-spread attention. As the name implies, the basic goal is to find a transformation in which the components s_i are statistically as independent from each other as possible. ICA can be applied, for example, for blind source separation, in which the observed values of $\boldsymbol{x}$ correspond to a realization of an m-dimensional discrete-time signal $\boldsymbol{x}(t), t = 1, 2, ...$ Then the components $s_i(t)$ are called source signals, which are usually original, uncorrupted signals or noise sources. Often such sources are statistically independent from each other, and thus the signals can be recovered from linear mixtures x_i by finding a transformation in which the transformed signals are as independent as possible, as in ICA. Another promising application is feature extraction, in which s_i is the coefficient of the i-th feature in the observed data vector $\boldsymbol{x}$. The use of ICA for feature extraction is motivated by results in neurosciences that suggest that the similar principle of redundancy reduction explains some aspects of the early processing of sensory data by the brain. ICA has also applications in exploratory data analysis in the same way as the closely related method of projection pursuit.

2.1 Second-Order Methods

The most popular methods for finding a linear transform as in (2.2) are second-order methods. This means methods that find the representation using only the information contained in the covariance matrix of the data vector $\boldsymbol{x}$. Of course, the mean is also used in the initial centering. The use of second-order techniques is to be understood in the context of the classical assumption of Gaussianity. If the variable $\boldsymbol{x}$ has a normal, or Gaussian distribution, its distribution is completely determined by this second-order information. Thus it is useless to include any other information. Another reason for the popularity of the second-order methods is that they are computationally simple, often requiring only classical matrix manipulations.

The two classical second-order methods are principal component analysis and factor analysis (Harman, 1967; Jolliffe, 2002; Kendall, 1975). One might roughly characterize the second-order methods by saying that their purpose is to find a faithful representation of the data, in the sense of reconstruction (mean-square) error. This is in contrast to most higher-order methods which try to find a meaningful representation. Of course, meaningfulness is a task-dependent property, but these higher-order methods seem to be able to find meaningful representations in a wide variety of applications (Comon, 1994; Friedman, 1987; Jutten et al., 1991).

2.1.1 Principal Component Analysis (PCA)

PCA is widely used in signal processing, statistics, and neural computing (?Kendall, 1975). In some application areas, this is also called the (discrete) Karhunen-Love transform, or the Hotelling transform. The basic idea in PCA is to find the components $s_1, s_2, ..., s_n$ so that they explain the maximum amount of variance possible by n linearly transformed components. PCA can be defined in an intuitive way using a recursive formulation. Define the direction of the first principal component, say $\boldsymbol{w}_1$, by

$$\boldsymbol{w}_1 = \underset{\|\boldsymbol{w}\|=1}{\operatorname{argmax}} \, \mathrm{E}\left\{(\boldsymbol{w}^T\boldsymbol{x})^2\right\} \tag{2.3}$$

where $\boldsymbol{w}_1$ is of the same dimension m as the random data vector $\boldsymbol{x}$. Thus the first principal component is the projection on the direction in which the variance of the projection is maximized. Having determined the first $k-1$ principal components, the k-th principal component is determined as the principal component of the residual:

$$\boldsymbol{w}_k = \underset{\|\boldsymbol{w}\|=1}{\operatorname{argmax}} \, \mathrm{E}\left\{\left[\boldsymbol{w}^T\left(\boldsymbol{x} - \sum_{i=1}^{k-1} \boldsymbol{w}_i\boldsymbol{w}_i^T\boldsymbol{x}\right)\right]^2\right\} \tag{2.4}$$

The principal components are then given by $s_i = \boldsymbol{w}_i^T\boldsymbol{x}$. In practice, the computation of the $\boldsymbol{w}_i$ can be simply accomplished using the (sample) covariance matrix $\mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\} = B$. The $\boldsymbol{w}_i$ are the eigenvectors of B that correspond to the n largest eigenvalues of B.

The basic goal in PCA is to reduce the dimension of the data. Thus one usually chooses $n << m$. Indeed, it can be proven that the representation given by PCA is an optimal linear dimension reduction technique in the mean-square sense (?). Such a reduction in dimension has important benefits. First, the computational overhead of the subsequent processing

stages is reduced. Second, noise may be reduced, as the data not contained in the n first components may be mostly due to noise. Third, a projection into a subspace of a very low dimension, for example two, is useful for visualizing the data. Note that often it is not necessary to use the n principal components themselves, since any other orthonormal basis of the subspace spanned by the principal components (called the PCA subspace) has the same data compression or denoising capabilities.

2.1.2 Factor Analysis

A method that is closely related to PCA is factor analysis (Harman, 1967; Kendall, 1975). In factor analysis, the following generative model for the data is postulated:

$$\boldsymbol{x} = A\boldsymbol{s} + \boldsymbol{b} \tag{2.5}$$

where $\boldsymbol{x}$ is the vector of the observed variables, $\boldsymbol{s}$ is the vector of the latent variables (factors) that cannot be observed, A is a constant $m \times n$ matrix, and the vector $\boldsymbol{b}$ is noise, of the same dimension, m, as $\boldsymbol{x}$. All the variables in $\boldsymbol{s}$ and $\boldsymbol{b}$ are assumed to be Gaussian. In addition, it is usually assumed that $\boldsymbol{s}$ has a lower dimension than $\boldsymbol{x}$. Thus, factor analysis is basically a method of reducing the dimension of the data, in a way similar to PCA. There are two main methods for estimating the factor analytic model (Kendall, 1975). The first method is the method of principal factors. As the name implies, this is basically a modification of PCA. The idea is here to apply PCA on the data $\boldsymbol{x}$ in such a way that the effect of noise is taken into account. In the simplest form, one assumes that the covariance matrix of the noise $\Sigma = \mathrm{E}\{\boldsymbol{b}\boldsymbol{b}^T\}$ is known. Then one finds the factors by performing PCA using the modified covariance matrix $C - \Sigma$, where C is the covariance matrix of $\boldsymbol{x}$. Thus the vector $\boldsymbol{s}$ is simply the vector of the principal components of with noise removed. A second popular method, based on maximum likelihood estimation, can also be reduced to finding the principal components of a modified covariance matrix. For the general case where the noise covariance matrix is not known, different methods for estimating it are described in (Harman, 1967; Kendall, 1975).

Nevertheless, there is an important difference between factor analysis and PCA, though this difference has little to do with the formal definitions of the methods. Equation (2.5) does not define the factors uniquely (i.e. they are not identifiable), but only up to a rotation (Harman, 1967; Kendall, 1975). This indeterminacy should be compared with the possibility of

choosing an arbitrary basis for the PCA subspace, i.e., the subspace spanned by the first n principal components. Therefore, in factor analysis, it is conventional to search for a 'rotation' of the factors that gives a basis with some interesting properties. The classical criterion is parsimony of representation, which roughly means that the matrix has few significantly non-zero entries. This principle has given rise to such techniques as the varimax, quartimax, and oblimin rotations (Harman, 1967). Such a rotation has the benefit of facilitating the interpretation of the results, as the relations between the factors and the observed variables become simpler.

2.2 Higher-Order Methods

Higher-order methods use information on the distribution of $\boldsymbol{x}$ that is not contained in the covariance matrix. In order for this to be meaningful, the distribution of $\boldsymbol{x}$ must not be assumed to be Gaussian, because all the information of (zero-mean) Gaussian variables is contained in the covariance matrix. For more general families of density functions, however, the representation problem has more degrees of freedom. Thus much more sophisticated techniques may be constructed for non-Gaussian random variables. Indeed, the transform defined by second-order methods like PCA is not useful for many purposes where optimal reduction of dimension in the mean-square sense is not needed. This is because PCA neglects such aspects of non-Gaussian data as clustering and independence of the components. We shall here review three conventional methods based on higher-order statistics: projection pursuit, redundancy reduction, and blind deconvolution.

2.2.1 Projection Pursuit

Projection pursuit (Friedman et al., 2001; Friedman, 1987; Huber, 1985; Jones et al., 1987) is a technique developed in statistics for finding 'interesting' projections of multidimensional data. Such projections can then be used for optimal visualization of the clustering structure of the data, and for such purposes as density estimation and regression. Reduction of dimension is also an important objective here, especially if the aim is visualization of the data. In basic (1-D) projection pursuit, we try to find directions $\boldsymbol{w}$ such that the projection of the data in that direction, $\boldsymbol{w}^T\boldsymbol{x}$, has an 'interesting' distribution, i.e., displays some structure. It has been argued by Huber (1985) and Jones and Sibson (1987) that the Gaussian distribution is the least interesting one, and that the most interesting directions are those that

show the least Gaussian distribution.

2.2.2 Redundancy Reduction

According to Barlow (1989a,b) and several other authors (Deco et al, 1995; Atick, 1992; Field, 1994; Schmidhuber et al., 1996), an important characteristic of sensory processing in the brain is 'redundancy reduction'. One aspect of redundancy reduction is that the input data is represented using components (features) that are as independent from each other as possible. Such a representation seems to be very useful for later processing stages. Theoretically, the values of the components are given by the activities of the neurons, and $\boldsymbol{x}$ is represented as a sum of the weight vectors of the neurons, weighted by their activations. This leads to a linear encoding like the other methods in this Section. One method for performing redundancy reduction is sparse coding (Field, 1994). Here the idea is to represent the data $\boldsymbol{x}$ using a set of neurons so that only a small number of neurons is activated at the same time. Equivalently, this means that a given neuron is activated only rarely. If the data has certain statistical properties (it is 'sparse'), this kind of coding leads to approximate redundancy reduction (Field, 1994). A second method for redundancy reduction is predictability minimization (Schmidhuber,Eldracher and Foltin, 1996). This is based on the observation that if two random variables are independent, they provide no information that could be used to predict one variable using the other one.

2.2.3 Blind Deconvolution

Blind deconvolution is different from the other techniques discussed in this Section in the sense that (in the very simplest case) were are dealing with one-dimensional time signals (or time series) instead of multidimensional data, though blind deconvolution can also be extended to the multidimensional case. Blind deconvolution is an important research topic with a vast literature. We shall here describe only a special case of the problem that is closely connected to ours. In blind deconvolution, a convolved version $x(t)$ of a scalar signal $s(t)$ is observed, without knowing the signal $s(t)$ or the convolution kernel (Donoho, 1981; Haykin, 1994, 1996; Shalvi et al., 1993). The problem is then to find a separating filter h so that $s(t) = h(t) * x(t)$.

The equalizer $h(t)$ is assumed to be a FIR filter of sufficient length, so that the truncation

effects can be ignored. A special case of blind deconvolution that is especially interesting in our context is the case where it is assumed that the values of the signal $s(t)$ at two different points of time are statistically independent. Under certain assumptions, this problem can be solved by simply whitening the signal $x(t)$. However, to solve the problem in full generality, one must assume that the signal $s(t)$ is non-Gaussian, and use higher-order information (Haykin, 1994; Shalvi et al., 1993). Thus the techniques used for solving this (special case of the) problem are very similar to the techniques used in other higher-order methods discussed in this Section.

Chapter 3

A Short Note on ICA

3.1 History of ICA

Independent Component Analysis was first formulated by Herault and Jutten (1986) in an attempt to solve the BSS problem in signal processing. Their approach has been further developed by Jutten et al. (1991), Comon et al. (1991), Karhunen et al. (1994), Cichocki et al. (1994). ICA was formally defined by Comon (1994), in which he proposed mutual information as the most natural measure of independence. In the same paper, he derived an approximation to the mutual information based on Edgeworth expansions in terms of cumulants. Similar expansions have been proposed by Amari et. al. (1996), while algebraic methods using cumulants have been explored by Cardoso et al (1996) and Cardoso (1999). It was also shown by Comon (1994) that the negentropy could be used as a proxy for the mutual information. Comon showed that maximizing the non-Gaussianity of the source signals was equivalent to minimizing the mutual information between them. An approximation to this measure was used in a nonlinear PCA implementation of ICA by Karhunen et al. (1994), Oja (1997), and by Hyvärinen et al. (1997) in their FastICA algorithm. Girolami also used negentropy approximations in projection pursuit formulations of ICA Girolami et al. (1996).

Linsker (1992) showed that linear mappings of Gaussian densities that maximize information transmittance - the 'INFOMAX' principle Linsker (1989) - perform PCA. It was further shown by Nadal and Parga (1994), that non-linear-mappings that follow this principle are capable of producing factored distributions in the source space. In an effort to model information transfer in neurons, Bell and Sejnowski (1995) extended the INFOMAX principle to

non-linear mappings of non-Gaussian densities. The input data was first mapped to intermediate variables by a linear transform, then these were mapped by sigmoidal non-linearities to an output. By maximizing the entropy of the output through adjusting the linear transform parameters, they showed that the intermediate variables were a linear ICA projection of the input data. Similar algorithms were independently suggested by Cardoso et al (1996).

Other algorithms for performing ICA have been proposed from different viewpoints. Maximum Likelihood Estimation (MLE) approaches to ICA were first proposed by Gaeta and Lacoume (1990) and elaborated by Pham, Garrat and Jutten (1992). Pearlmutter and Parra (1996), Mackay (1996) and Cardoso (1997) showed that the infomax approach of Bell and Sejnowski (1995) and the maximum likelihood estimation approach are equivalent. Everson and Roberts (1995) extended these methods by incorporating a flexible generalized-exponential model for the source densities that could learn both super- and sub-Gaussian distributions. This is equivalent to learning the non-linearity in the INFOMAX case. They also noted that an unmixing matrix that has independent columns must also be decorrelated. This information was used to constrain the learning to the manifold of decorrelating matrices, thereby greatly speeding up the process.

3.1.1 What is Independent Component?

The components of a random vector $\boldsymbol{y} = (y_1, y_2, \ldots y_m)^T$ are said to be independent of each other if and only if the density function of $\boldsymbol{y}$ is factorized (Papoulis, 1992) as

$$p(\boldsymbol{y}) = \prod_{i=1}^{m} p_i(y_i), \tag{3.1}$$

where

$$p_i(y_i) = \int p(\boldsymbol{y}) dy_1 \ldots dy_{i-1} dy_{i+1} \ldots dy_m,$$

is the marginal density of y_i, $(i = 1, 2, \ldots, m)$. If components of $\boldsymbol{y}$ are independent of each other, then most important property of their independence is

$$\mathrm{E}\left\{\prod_{i=1}^{m} h_i(y_i)\right\} = \prod_{i=1}^{m} \mathrm{E}\left\{h_i(y_i)\right\}, \tag{3.2}$$

where, $h_i(y_i)$ is any measurable function of y_i.

3.1.2 Uncorrelated Components are Only Partly Independent

Let us assume components of $\boldsymbol{y} = (y_1, y_2, \ldots y_m)^T$ are independent of each other, then they are pair-wise independent also, that is

$$p\,(y_i, y_j) = p_i(y_i)p_j(y_j), \quad (i \neq j; i, j = 1, 2, ..., m) \tag{3.3}$$

which implies

$$\mathrm{E}\,(y_i, y_j) = \mathrm{E}(y_i)\mathrm{E}(y_j), \quad (i \neq j; i, j = 1, 2, ..., m) \tag{3.4}$$

Then pair-wise covariances are

$$\mathrm{Cov}\,(y_i, y_j) = \mathrm{E}\,(y_i, y_j) - \mathrm{E}(y_i)\mathrm{E}(y_j) = 0, \quad (i \neq j; i, j = 1, 2, ..., m) \tag{3.5}$$

which implies that pair-wise correlations coefficients are

$$r_{ij} = \frac{\mathrm{Cov}\,(y_i, y_j)}{\mathrm{Var}(y_i)\mathrm{Var}(y_j)} = 0, \quad (i \neq j; i, j = 1, 2, ..., m) \tag{3.6}$$

Therefore, multiple correlation coefficient of y_i on $y_1, y_2, ..., y_{i-1}, y_{i+1}, ..., y_m$ is

$$R_{i\cdot 12...(i-1)(i+1)...m} = \left(1 - \frac{\det(\rho)}{\det(\rho_{ii})}\right)^{\frac{1}{2}} = 0, \quad (i = 1, 2, ..., m) \tag{3.7}$$

where $\rho = (r_{ij})$ is the correlation matrix of order $m \times m$ which reduces to identity matrix by (3.6), and $\det(\rho_{ii})$ is the cofactor of the element in the i-th row and i-th column of ρ. Hence independent components are mutually uncorrelated. However, uncorrelatedness does not imply independence. For example, let us assume that two components y_1 and y_2 are dependent of each other by $y_2 = y_1^2$, where $y_1 \sim N(0, 1)$. Then $\mathrm{E}(y_1) = 0 = \mathrm{E}(y_1^3)$. Therefore, the covariance between y_1 and y_2 is

$$\mathrm{Cov}(y_1, y_2) = \mathrm{E}(y_1 y_2) - \mathrm{E}(y_1)\mathrm{E}(y_2) = \mathrm{E}(y_1^3) - \mathrm{E}(y_1)\mathrm{E}(y_2) = 0.$$

which concludes the uncorrelatedness y_1 and y_2. Thus independent components are always uncorrelated but the converse is not true. Hence uncorrelatedness is a weaker form of independence. It should be noted here that uncorrelated Gaussian components are always independent. Since independence implies uncorrelatedness always, many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem.

3.2 Concept of ICA

Independent component analysis (ICA) is a statistical tool for revealing hidden factors that underlie sets of random variables, measurements, or signals. The ICA defines a generative model for the observed multivariate data. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA. ICA is superficially related to principal component analysis (PCA) and factor analysis (FA). ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely. ICA can be considered an extension of principal component analysis (PCA). In PCA, the input data is decorrelated to find the components that are maximally correlated according to second order statistics. PCA gives orthogonalized and normalized outputs according to the second order statistics by minimizing the second order moments. The principal components can still be dependent however. In ICA, the aim is to process a number of measured signal vectors X and extract a set of statistically independent vectors Y which are estimates of some unknown source signals S which have been linearly mixed together via a mixing matrix A to form the observed input data. ICA seeks to ensure maximum independence, typically by minimizing the higher order moments of the outputs. When the higher order moments are zero (for non-Gaussian input signals), the outputs are independent.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation (BSS) is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process. Many algorithms have been proposed to perform ICA. These may be divided into block-based or on-line adaptive techniques. Block-based algorithms take all the data in at once and produce the output. On-line adaptive algorithms process each data point in a continuous sense. A disadvantage of many algorithms, especially on-line adaptive algorithms, is the need to

select tuning parameters such as the learning rate. If the learning rate is chosen to be too small, then the solution may not be found or it may be found very slowly. If the learning rate is chosen too large, then the algorithm may 'blow-up'. For many practical problems, it is useful to be able to have an algorithm that is capable of finding a solution without user intervention at all.

3.2.1 Definition of ICA

Let $\boldsymbol{s}(t) = \left(s_1(t), s_2(t), ..., s_q(t)\right)^T$ be a vector of q source (original) signals whose components are assumed to be mutually independent and non-Gaussian with zero mean vector. In practice, we cannot observe vector of original signals $\boldsymbol{s}(t)$ directly, but observe random vector $\boldsymbol{x}(t) = \left(x_1(t), x_2(t), ..., x_m(t)\right)^T$ of m mixed signals by the linear transform

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t), \qquad t = 1, 2, ..., n \tag{3.8}$$

where t is the time index and A is an unknown full rank $m \times q$ mixing matrix. The ICA of a random vector $\boldsymbol{x}(t)$ consists of finding a linear transform

$$\boldsymbol{y}(t) = W\boldsymbol{x}(t), \qquad t = 1, 2, ..., n \tag{3.9}$$

so that components of $\boldsymbol{y}(t) = \left(y_1(t), y_2(t), ..., y_q(t)\right)^T$ are as mutually independent as possible, where W is a $q \times m$ transformation matrix obtained by ICA algorithm (Comon, 1994; Cardoso et al, 1996; Hyvärinen, 1999c; Lee, 2001; Cichoki et al., 2002). It is also known as recovering matrix or unmixing matrix or pseudo-inverse of A or generalized inverse of A. There are two principal approaches to solve the ICA problem. The first approach is to separate all sources simultaneously and the second approach is to separate all sources sequentially (Cichoki et al., 2002).

3.2.2 Identifiability of the ICA Model

The identifiability of the noise-free ICA model has been treated byComon (1994) and Cardoso et al (1996) . By imposing the following fundamental restrictions, the identifiability of the model (3.8) can be assured.

1. The component of source vector $\boldsymbol{s}(t) = \Big(s_1(t), s_2(t), ..., s_q(t)\Big)^T$ are mutually independent at each time instant t.

2. At most one source signal can be normally distributed and the rest (q-1) source signals must be non-Gaussian.

3. The number of sensors or observed linear mixtures m must be greater than or equal to the number of sources q, i.e., $m \geq q$.

4. The matrix A must be of full column rank.

Usually, it is also assumed that $\boldsymbol{x}(t)$ and $\boldsymbol{s}(t)$ are centered, which is in practice no restriction, as this can always be accomplished by subtracting the mean from the random vector. The third restriction, is not completely necessary. Even in the case where $m < q$, the mixing matrix seems to be identifiable (Hyvärinen, 1999c).

3.2.3 Ambiguities of ICA

1. The variances (energies) of the estimated independent components might be different from the variance of the original independent components. That is, scaling factor of the mixing matrix as well as the sources cannot be determined. Note that this still leaves the ambiguity of the sign: one could multiply an independent component by -1 without affecting the model. However, this ambiguity is, fortunately, insignificant in most applications (Hyvärinen et al., 2001).

2. The order of the independent components cannot be determine.

3.2.4 Why Gaussian Components are Forbidden for ICA

Assume that components of 2-dimensional source vector $\boldsymbol{s} = (s_1, s_2)^T$ are Gaussian. Then

$$p(\boldsymbol{s}) = \frac{1}{2\pi}\exp\left(-\frac{\|\boldsymbol{s}\|^2}{2}\right) \tag{3.10}$$

Let $\boldsymbol{x} = A\boldsymbol{s}$ be a linear transform, where A is an orthogonal matrix, i.e., $A^{-1} = A^T$. Then

$$p(\boldsymbol{x}) = \frac{1}{2\pi}\exp\left(-\frac{\left\|A^T\boldsymbol{x}\right\|^2}{2}\right)\left|\det A^T\right| \tag{3.11}$$

Due to the orthogonality of A, We have $\left\|A^T\boldsymbol{x}\right\|^2 = \|\boldsymbol{x}\|$ and $|\det A| = 1$. Thus we have,

$$p(\boldsymbol{x}) = \frac{1}{2\pi}\exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right) \tag{3.12}$$

Obviously, the original distribution (3.10) and mixed distribution (3.12) are identical. Hence orthogonal transformation of the Gaussian components remain Gaussian and mutually independent. Also Gaussian distribution is rotationally symmetric [Insert FIGURE]. Therefore, it does not contain any information on the directions of the columns of the orthogonal mixing matrix A. This is why A cannot be estimated. Thus, in the case of Gaussian variables, we can only estimate the ICA model up to an orthogonal transformation. In other words, the matrix A is not identifiable for Gaussian independent components.

What happens if we try to estimate the ICA model and some of the components are Gaussian, some non-Gaussian? In this case, we can estimate all the non-Gaussian components, but the Gaussian components cannot be separated from each other. In other words, some of the estimated components will be arbitrary linear combinations of the Gaussian components. Actually, this means that in the case of just one Gaussian component, we can estimate the model, because the single Gaussian component does not have any other Gaussian components that it could be mixed with (Hyvärinen et al., 2001).

3.2.5 Relations to Classical Linear Transformation

ICA is closely related to several of the methods described in chapter 2.

1. By definition, ICA can be considered a method for achieving redundancy reduction. Indeed, there is experimental evidence that for certain kinds of sensory data, the conventional ICA algorithms do find directions that are compatible with existing neurophysiological data, assumed to reflect redundancy reduction (Bell et al., 1997; Hurri, 1997).

2. In the noise-free case, the estimation of the ICA model means simply finding certain 'interesting' projections, which give estimates of the independent components. Thus

ICA can be considered a special case of projection pursuit. The conventional criteria used for finding the 'interesting' directions in projections pursuit coincide essentially with the criteria used for estimating the independent components

3. Another close affinity can be found between ICA and blind deconvolution (more precisely, the special case of blind deconvolution where the original signal is i.i.d. over time). Due to the assumption that the values of the original signal s(t) are independent for different t, this problem is formally closely related to the problem of independent component analysis. Indeed, many ideas developed for blind deconvolution can be directly applied for ICA, and vice versa. Blind deconvolution, and especially the elegant and powerful framework developed in Donoho (1981), can thus be considered an intellectual ancestor of ICA.

4. Comparing ICA model (3.8) after adding a bias term with the definition of factor analysis in Eq. (2.5), the connection between factor analysis and ICA becomes clear. Indeed, ICA may be considered a non-Gaussian factor analysis. The main difference is that usually in ICA, reduction of dimension is considered only as a secondary objective, but this need not be the case. Indeed, a simple combination of factor analysis and ICA can be obtained using factor rotations. Above we saw that after finding the factor subspace, a suitable rotation is usually performed. ICA could also be conceived as such a rotation, where the criterion depends on the higher-order statistics of the factors, instead of the structure of the matrix . Such a method is roughly equivalent to the method advocated in (Hyvärinen et al., 1996; Karhunen et al., 1997) which consists of first reducing the dimension by PCA, and then performing ICA without further dimension reduction.

5. Using ICA model (3.8), the relation to principal component analysis is also evident. Both methods formulate a general objective function that define the 'interestingness' of a linear representation, and then maximize that function. A second relation between PCA and ICA is that both are related to factor analysis, though under the contradictory assumptions of Gaussianity and non-Gaussianity, respectively. The affinity between PCA and ICA may be, however, less important than the affinity between ICA and the other methods discussed above. This is because PCA and ICA define their objective functions in quite different ways. PCA uses only second-order statistics,

while ICA is impossible using only second-order statistics. PCA emphasizes dimension reduction, while ICA may reduce the dimension, increase it or leave it unchanged. However, the relation between ICA and nonlinear versions of the PCA criteria, as defined in (Karhunen et al., 1994; Oja, 1997), is quite strong.

3.3 Preprocessing for ICA

It is usually very useful to do some preprocessing before applying an ICA algorithm on the data, In this section, we discuss some preprocessing techniques that make the problem of ICA estimation simpler and better conditioned. The preprocessing is also known as prewhitening.

3.3.1 Centering

In the general ICA model (3.8), it is assumed that both the source vectors $\boldsymbol{s}(t)$, $(t = 1, 2, ..., n)$ and the mixed vectors $\boldsymbol{x}(t)$, $(t = 1, 2, ..., n)$ have zero mean vector. To make it hold in practice, centering is the necessary preprocessing. In the centering procedure, the original mixed vector $\boldsymbol{x}_t$ is processed by

$$\boldsymbol{x}(t) = \boldsymbol{x}_t - \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}_j, \quad (t = 1, 2, ..., n) \tag{3.13}$$

before doing ICA. Thus the independent components are made zero mean as well, since

$$\mathrm{E}(\boldsymbol{s}) = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{s}(t) = \frac{A}{n}\sum_{t=1}^{n}\boldsymbol{x}(t) = 0, \tag{3.14}$$

The mixing matrix A remains the same after this preprocessing, so one can always do this without affecting the estimation of the mixing matrix.

3.3.2 Whitening

After centering, another useful preprocessing strategy in ICA is the whitening of the observed variables before the application of the ICA algorithm. A zero mean random vector $\boldsymbol{z}$ is said to be white or sphere if $\mathrm{E}(\boldsymbol{z}\boldsymbol{z}^T) = \mathrm{I}$, (identity matrix). In the whitening procedure, a zero mean random vector $\boldsymbol{x}(t)$ is processed by

$$\boldsymbol{z}(t) = V^{-\frac{1}{2}}\boldsymbol{x}(t) \tag{3.15}$$

where,

$$V = \mathrm{E}(\boldsymbol{x}\boldsymbol{x}^T) = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{x}(t)\boldsymbol{x}(t)^T \tag{3.16}$$

such that

$$\mathrm{E}(\boldsymbol{z}\boldsymbol{z}^T) = V^{-1/2}\mathrm{E}(\boldsymbol{x}\boldsymbol{x}^T)\left(V^{-\frac{1}{2}}\right)^T = V^{-\frac{1}{2}}V\left(V^{-1/2}\right) = \mathbf{I} \tag{3.17}$$

The whitening matrix $V^{-1/2}$ can be computed by the eigen-value decomposition (EVD) of the covariance matrix $V = EDE^T$, where E is the orthogonal matrix of eigenvectors of V and $D = \mathrm{diag}(d_1, d_2, ..., d_m)$ is the diagonal matrix of its eigenvalues. Then $V^{-1/2} = ED^{-1/2}E^T$ and the whitening can be done by

$$\boldsymbol{z}(t) = ED^{-1/2}E^T\boldsymbol{x}(t), \tag{3.18}$$

Where $D^{-1/2} = \mathrm{diag}(d_1^{-1/2}, d_2^{-1/2}, ..., d_m^{-1/2})$. Obviously, components of a whiten vector are mutually uncorrelated.

3.3.3 Whitening is Only $\frac{1}{2}$ ICA

Let us assume that the data in the ICA model is whitened by (3.18). Using (3.8) in (3.18), we have

$$\boldsymbol{z}(t) = V^{-1/2}A\boldsymbol{s}(t) = B\boldsymbol{s}(t), \tag{3.19}$$

where, $B = V^{-1/2}A$ is the orthogonal matrix, since

$$\mathrm{E}(\boldsymbol{z}\boldsymbol{z}^T) = B\mathrm{E}(\boldsymbol{s}\boldsymbol{s}^T)B^T = BB^T = \mathbf{I} \tag{3.20}$$

Thus whitening gives the independent components only upto an orthogonal transformation. Instead of having to estimate the m^2 parameters that are the elements of the original matrix A, we only need to estimate an orthogonal mixing matrix B. An orthogonal matrix contains $n(n-1)/2$ degrees of freedom. In larger dimensions, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix. Thus one can say that whitening solves half of the problems of ICA. The remaining half of the parameters has to be estimated by some other methods. It reduces the complexity of the ICA problems (Hyvärinen et al., 2001).

3.4 Principles of ICA Estimation

3.4.1 Non-Gaussian is Independent

The key of estimating ICA model is non-Gaussianity. The estimation is not possible at all without non-Gaussianity, as mentioned in Section 3.2.4. To see how non-Gaussianity leads to

the basic principle of ICA estimation, let us consider the linear ICA model for an observable vector $\boldsymbol{x} = (x_1, x_2, ..., x_m)^T$ of dimension m as

$$\boldsymbol{x} = A\boldsymbol{s} \tag{3.21}$$

where $\boldsymbol{A} \in R^{m \times m}$ is a non-singular mixing matrix and $\boldsymbol{s} = \left(s_1, s_2, ..., s_m\right)^T$ is an unobservable source vector whose components are assumed mutually independent. For simplicity, let us assume in this section that all the independent components have identical distributions. The ICA of a random vector $\boldsymbol{x}$ consists of finding a linear transform

$$\boldsymbol{y} = W\boldsymbol{x} \tag{3.22}$$

so that components of $\boldsymbol{y} = \left(y_1, y_2, ..., y_m\right)^T$ are as mutually independent as possible, where W is the estimate of A^{-1} obtained by ICA algorithm (Comon, 1994; Cardoso et al, 1996; Hyvärinen, 1999c; Lee, 2001; Cichoki et al., 2002). Therefore, to estimate one independent component from the set $\{s_1, s_2, ..., s_m\}$, we consider

$$\begin{aligned} y &= \boldsymbol{w}\boldsymbol{x}, \quad \text{where } \boldsymbol{w} \text{ is a row of } W && (3.23)\\ &= \boldsymbol{w}A\boldsymbol{s}, \quad \text{since } \boldsymbol{x} = A\boldsymbol{s} && (3.24)\\ &= \boldsymbol{b}\boldsymbol{s}, \quad \text{where } \boldsymbol{b} = \boldsymbol{w}A && (3.25)\\ &= \sum_{i=1}^{m} b_i s_i && (3.26) \end{aligned}$$

If $\boldsymbol{w}$ is equal to one of the rows of A^{-1}, then only one element of $\boldsymbol{b}$ is equal to 1 and all other elements are 0. Then, y is equal to one of the independent components from the set $\{s_1, s_2, ..., s_m\}$. In practice, we cannot determine such a $\boldsymbol{w}$ exactly, because mixing matrix A is unknown, however, we can find an estimator by ICA method that gives a good approximation. If $\boldsymbol{w}$ is not equal to one of the rows of A^{-1}, then more than one element of $\boldsymbol{b}$ are non-zero. Then y is obtained by (3.26) using the non-zero elements of $\boldsymbol{b}$. Therefore, by statistical central limit theorem, y in (3.26) is more Gaussian than any one independent component of $\{s_1, s_2, ..., s_m\}$ and becomes least Gaussian when it equals to one component of $\{s_1, s_2, ..., s_m\}$. Therefore, minimizing the Gaussianity or equivalently maximizing the non-Gaussianity of $\boldsymbol{w}\boldsymbol{x}$ with respect to $\boldsymbol{w}$ gives us one of the independent components. Hence non-Gaussian is independent.

3.4.2 Maximization of non-Gaussianity

To use non-Gaussianity in ICA estimation, we must have a quantitative measure of non-Gaussianity of a random variable. One can maximize non-Gaussianity of a random variable by maximizing (i) absolute value of kurtosis or square of kurtosis (ii) Negentropy, and (iii) Approximations of negentropy, (Hyvärinen and Oja, 2000).

Kurtosis

The classical measure of non-Gaussianity is kurtosis or the fourth-order cumulant. The kurtosis of y is classically defined by

$$\begin{aligned} \text{kurt}(y) &= \text{E}\{y^4\} - 3(\text{E}\{y^2\})^2, \quad \text{if } y \text{ is centered.} && (3.27) \\ &= \text{E}\{y^4\} - 3, \quad \text{if } y \text{ is normalized or } \text{E}\{y^2\} = 1 && (3.28) \end{aligned}$$

which implies

$$\begin{aligned} \text{kurt}(y) &= 0, \quad \text{if } y \text{ is Gaussian, (normal curve)} && (3.29) \\ &> 0, \quad \text{if } y \text{ is super-Gaussian, (leptokurtic curve)} && (3.30) \\ &< 0, \quad \text{if } y \text{ is sub-Gaussian, (platykurtic curve)} && (3.31) \end{aligned}$$

Typically non-Gaussianity is measured by the absolute value of kurtosis. The square of kurtosis can also be used. From (3.29 — 3.31), we see that

$$\begin{aligned} |\text{kurt}(y)| &= 0, \quad \text{if } y \text{ is Gaussian} && (3.32) \\ &> 0, \quad \text{if } y \text{ is non-Gaussian} && (3.33) \end{aligned}$$

Therefore, one can maximize non-Gaussianity of a random variable by maximizing absolute value of kurtosis or square of kurtosis. Note that there are few non-Gaussian random variables that have zero kurtosis, but they can be considered to be very rare, (Hyvärinen et al., 2001). The main problem of ICA estimation by kurtosis is that it is very much sensitive to outliers.

Negentropy

A second very important measure of non-Gaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of (differential) entropy. Entropy is the basic

concept of information theory. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy. More rigorously, entropy is closely related to the coding length of the random variable, in fact, under some simplifying assumptions, entropy is the coding length of the random variable (Cover et al, 1991; Papoulis, 1992). The Entropy H is defined for a discrete random variable Y as

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i) \tag{3.34}$$

where the a_i are the possible values of Y. This very well-known definition can be generalized for continuous-valued random variables and vectors, in which case it is often called differential entropy. The differential entropy H of a random vector $\boldsymbol{y} = (y_1, y_2, ..., y_m)^T$ with density $f(\boldsymbol{y})$ is defined as

$$H(\boldsymbol{y}) = -\int f(\boldsymbol{y}) \log f(\boldsymbol{y}) d\boldsymbol{y} = \sum_{i=1}^{m} H(y_i | y_{i-1}, ..., y_1) \tag{3.35}$$

$$= \sum_{i=1}^{m} H(y_i), \quad \text{if components of } \boldsymbol{y} \text{ are mutually independent} \tag{3.36}$$

where

$$H(y_i | y_{i-1}, ..., y_1) = \int p(y_1, y_2, ..., y_i) \log p(y_i | y_{i-1}, ..., y_1) dy_1 ... dy_i \tag{3.37}$$

is the conditional entropy of y_i given $y_1, y_2, ..., y_{i-1}$. A fundamental result of information theory is that a Gaussian variable has the largest entropy among all random variables of equal variance (Cover et al, 1991; Papoulis, 1992). This means that entropy could be used as a measure of non-Gaussianity. To obtain a measure of non-Gaussianity that is zero for a Gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{\text{gauss}}) - H(\boldsymbol{y}) \tag{3.38}$$

where $\boldsymbol{y}_{\text{gauss}}$ is a Gaussian random variable of the same covariance matrix as $\boldsymbol{y}$. Due to the above-mentioned properties, negentropy is always non-negative, and it is zero if and only if $\boldsymbol{y}$ has a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations (Comon, 1994; Hyvärinen et al., 2001). The advantage of using negentropy, or, equivalently, differential entropy, as a measure of non-Gaussianity is that it is well justified by statistical theory. In fact, negentropy is in some sense

3.4.2 Maximization of non-Gaussianity

To use non-Gaussianity in ICA estimation, we must have a quantitative measure of non-Gaussianity of a random variable. One can maximize non-Gaussianity of a random variable by maximizing (i) absolute value of kurtosis or square of kurtosis (ii) Negentropy, and (iii) Approximations of negentropy, (Hyvärinen and Oja, 2000).

Kurtosis

The classical measure of non-Gaussianity is kurtosis or the fourth-order cumulant. The kurtosis of y is classically defined by

$$\begin{aligned} \text{kurt}(y) &= \text{E}\{y^4\} - 3(\text{E}\{y^2\})^2, \quad \text{if } y \text{ is centered.} && (3.27) \\ &= \text{E}\{y^4\} - 3, \quad \text{if } y \text{ is normalized or } \text{E}\{y^2\} = 1 && (3.28) \end{aligned}$$

which implies

$$\begin{aligned} \text{kurt}(y) &= 0, \quad \text{if } y \text{ is Gaussian, (normal curve)} && (3.29) \\ &> 0, \quad \text{if } y \text{ is super-Gaussian, (leptokurtic curve)} && (3.30) \\ &< 0, \quad \text{if } y \text{ is sub-Gaussian, (platykurtic curve)} && (3.31) \end{aligned}$$

Typically non-Gaussianity is measured by the absolute value of kurtosis. The square of kurtosis can also be used. From (3.29 — 3.31), we see that

$$\begin{aligned} |\text{kurt}(y)| &= 0, \quad \text{if } y \text{ is Gaussian} && (3.32) \\ &> 0, \quad \text{if } y \text{ is non-Gaussian} && (3.33) \end{aligned}$$

Therefore, one can maximize non-Gaussianity of a random variable by maximizing absolute value of kurtosis or square of kurtosis. Note that there are few non-Gaussian random variables that have zero kurtosis, but they can be considered to be very rare, (Hyvärinen et al., 2001). The main problem of ICA estimation by kurtosis is that it is very much sensitive to outliers.

Negentropy

A second very important measure of non-Gaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of (differential) entropy. Entropy is the basic

concept of information theory. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy. More rigorously, entropy is closely related to the coding length of the random variable, in fact, under some simplifying assumptions, entropy is the coding length of the random variable (Cover et al, 1991; Papoulis, 1992). The Entropy H is defined for a discrete random variable Y as

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i) \tag{3.34}$$

where the a_i are the possible values of Y. This very well-known definition can be generalized for continuous-valued random variables and vectors, in which case it is often called differential entropy. The differential entropy H of a random vector $\boldsymbol{y} = (y_1, y_2, ..., y_m)^T$ with density $f(\boldsymbol{y})$ is defined as

$$H(\boldsymbol{y}) = -\int f(\boldsymbol{y}) \log f(\boldsymbol{y}) d\boldsymbol{y} = \sum_{i=1}^{m} H(y_i|y_{i-1}, ..., y_1) \tag{3.35}$$

$$= \sum_{i=1}^{m} H(y_i), \quad \text{if components of } \boldsymbol{y} \text{ are mutually independent} \tag{3.36}$$

where

$$H(y_i|y_{i-1}, ..., y_1) = \int p(y_1, y_2, ..., y_i) \log p(y_i|y_{i-1}, ..., y_1) dy_1 ... dy_i \tag{3.37}$$

is the conditional entropy of y_i given $y_1, y_2, ..., y_{i-1}$. A fundamental result of information theory is that a Gaussian variable has the largest entropy among all random variables of equal variance (Cover et al, 1991; Papoulis, 1992). This means that entropy could be used as a measure of non-Gaussianity. To obtain a measure of non-Gaussianity that is zero for a Gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{\text{gauss}}) - H(\boldsymbol{y}) \tag{3.38}$$

where $\boldsymbol{y}_{\text{gauss}}$ is a Gaussian random variable of the same covariance matrix as $\boldsymbol{y}$. Due to the above-mentioned properties, negentropy is always non-negative, and it is zero if and only if $\boldsymbol{y}$ has a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations (Comon, 1994; Hyvärinen et al., 2001). The advantage of using negentropy, or, equivalently, differential entropy, as a measure of non-Gaussianity is that it is well justified by statistical theory. In fact, negentropy is in some sense

the optimal estimator of non-Gaussianity, as far as statistical properties are concerned. The problem in using negentropy is, however, that it is computationally very difficult. Therefore, simpler approximations of negentropy are very useful, as will be discussed next.

Approximations of Negentropy

The estimation of negentropy is difficult, as mentioned above, and therefore this contrast function remains mainly a theoretical one. In practice, some approximation have to be used. An approximation of negentropy is expressed in term of higher-order moments as follows (Jones et al., 1987):

$$J(y) \approx \frac{1}{12}\mathrm{E}\{y^3\}^2 + \frac{1}{48}\mathrm{kurt}(y)^2 \tag{3.39}$$

The random variable y is assumed to be of zero mean and unit variance. However, the validity of such approximations may be rather limited. In particular, these approximations suffer from the non-robustness encountered with kurtosis. To avoid the problems encountered with the preceding approximations of negentropy, new approximations are developed in (Hyvärinen, 1998b). In general we obtain the following approximation:

$$J(y) \approx \sum_{i=1}^{p} k_i \Big[\mathrm{E}\{G_i(y)\} - \mathrm{E}\{G_i(\nu)\}\Big]^2 \tag{3.40}$$

where k_i are some positive constants, and ν is a Gaussian variable of zero mean and unit variance (i.e., standardized). The variable y is assumed to be of zero mean and unit variance, and the functions G_i are some non-quadratic functions (Hyvärinen, 1998b). Note that even in cases where this approximation is not very accurate, (3.40) can be used to construct a measure of non-Gaussianity that is consistent in the sense that it is always non-negative, and equal to zero if y has a Gaussian distribution. In the case where we use only one non-quadratic function G, the approximation becomes

$$J(y) \approx \Big[\mathrm{E}\{G(y)\} - \mathrm{E}\{G(\nu)\}\Big]^2 \tag{3.41}$$

for practically any non-quadratic function G. This is clearly a generalization of the moment-based approximation in (3.40), if y is symmetric. Indeed, taking $G(y) = y^4$, one then obtains exactly (3.40), i.e. a kurtosis-based approximation. But the point here is that by choosing G wisely, one obtains approximations of negentropy that are much better than the one given by (3.40). In particular, choosing G that does not grow too fast, one obtains more robust

estimators. The following choices of G have proved very useful:

$$G_1(u) = \frac{1}{a} \log \cosh a_1 u, \qquad G_2(u) = -\exp(-u^2/2) \tag{3.42}$$

where $1 \leq a \leq 2$ is some suitable constant. Thus we obtain approximations of negentropy that give a very good compromise between the properties of the two classical non-Gaussianity measures given by kurtosis and negentropy. They are conceptually simple, fast to compute, yet have appealing statistical properties, especially robustness. Therefore, we shall use these contrast functions in our ICA methods. Since kurtosis can be expressed in this same framework, it can still be used by our ICA methods.

3.4.3 Minimization of Mutual Information

Another approach for ICA estimation, inspired by information theory, is minimization of mutual information. We will explain this approach here, and show that it leads to the same principle of finding most non-Gaussian directions as was described above. In particular, this approach gives a rigorous justification for the heuristic principles used above.

Mutual Information

Using the concept of differential entropy, we define the mutual information I between m (scalar) random variables, $y_i, i = 1...m$ as follows

$$I(y_1, y_2, ..., y_m) = \sum_{i=1}^{m} H(y_i) - H(y) \tag{3.43}$$

It is always non-negative, that is $I(y_1, y_2, ..., y_m) \geq 0$, equality holds if and only if the variables are statistically independent. Mutual information is a natural measure of the dependence between random variables. It is equivalent to the well-known Kullback-Leibler divergence between the joint density and the product of its marginal densities, that is

$$I(y_1, y_2, ..., y_m) = D_{\mathrm{KL}}\left(p, \prod_{i=1}^{m} p_i\right) = \int p(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{\prod_{i=1}^{m} p_i(y_i)} d\boldsymbol{y} \tag{3.44}$$

Thus, mutual information takes into account the whole dependence structure of the variables, and not only the covariance, like PCA and related methods. An important property of mutual information (Papoulis, 1992; Cover et al, 1991) is that we have for an invertible linear transformation $\boldsymbol{y} = W\boldsymbol{x}$:

$$I(y_1, y_2, ..., y_m) = \sum_{i=1}^{m} H(y_i) - H(\boldsymbol{x}) - \log |\det W| . \tag{3.45}$$

Now, let us consider what happens if we constrain the y_i to be uncorrelated and of unit variance. This means, $\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^T\} = W\mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\}W^T = \mathrm{I}$, which implies,

$$\det \mathbf{I} = 1 = \det\left(W\mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\}W^T\right) = (\det W)\left(\det \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\}\right)(\det W^T) \tag{3.46}$$

and this implies that $\det W$ must be constant. Moreover, for y_i of unit variance, entropy and negentropy differ only by a constant, and the sign. Thus we obtain,

$$I(y_1, y_2, ..., y_m) = c - \sum_{i=1}^{m} J(y_i). \tag{3.47}$$

where c is a constant that does not depend on W. This shows the fundamental relation between negentropy and mutual information.

It is now obvious from (3.47) that finding an invertible transformation W that minimizes the mutual information is roughly equivalent to finding directions in which the negentropy is maximized. Rigorously, speaking, (3.47) shows that ICA estimation by minimization of mutual information is equivalent to maximizing the sum of non-Gussianities of the estimates, when the estimates are constrained to be uncorrelated.

3.4.4 Maximum Likelihood Estimation

The Likelihood

A very popular approach for estimating the ICA model is maximum likelihood estimation, which is closely connected to the infomax principle. Here we discuss this approach, and show that it is essentially equivalent to minimization of mutual information. It is possible to formulate directly the likelihood in the noise-free ICA model, which was done in (Pham et al., 1992), and then estimate the model by a maximum likelihood method. Denoting by $W = (\boldsymbol{w}_1, ..., \boldsymbol{w}_m)^T$ the matrix A^{-1}, the log-likelihood takes the form (Pham et al., 1992).

$$L = \sum_{t=1}^{n}\sum_{i=1}^{m} \log f_i(\boldsymbol{w}_i^T \boldsymbol{x}(t)) + T \log|\det W| \tag{3.48}$$

where the f_i are the density functions of the s_i (here assumed to be known), and the $\boldsymbol{x}(t)$, $t = 1, 2, ..., n$ are the realizations of $\boldsymbol{x}$. The term $\log|\det W|$ in the likelihood comes from the classic rule for (linearly) transforming random variables and their densities (Papoulis, 1992): In general, for any random vector $\boldsymbol{x}$ with density p_x and for any matrix W, the density $\boldsymbol{y} = W\boldsymbol{x}$ is given by $p_x(Wx)|\det|$ of is given by .

The Infomax Principle

Another related contrast function was derived from a neural network viewpoint in (Bell et al., 1995; Nadal et al., 1994). This was based on maximizing the output entropy (or information flow) of a neural network with non-linear outputs. Assume that $\boldsymbol{x}$ is the input to the neural network whose outputs are of the form $g_i(\boldsymbol{w}_i^T\boldsymbol{x})$, where the g_i are some non-linear scalar functions, and the are the weight vectors of the neurons. One then wants to maximize the entropy of the outputs:

$$L_2 = H(g_1(\boldsymbol{w}_1^T\boldsymbol{x}), ..., g_n(\boldsymbol{w}_n^T\boldsymbol{x})) \tag{3.49}$$

If the g_i are well chosen, this framework also enables the estimation of the ICA model. Indeed, several authors, e.g., (Cardoso, 1997; Bell et al., 1995; Comon, 1994), proved the surprising result that the principle of network entropy maximization, or "infomax", is equivalent to maximum likelihood estimation. This equivalence requires that the non-linearities g_i used in the neural network are chosen as the cumulative distribution functions corresponding to the densities f_i, i.e., $g_i'(.) = f_i(.)$.

Connection to Mutual Information

To see the connection between likelihood and mutual information, consider the expectation of the log-likelihood:

$$\frac{1}{T}\mathrm{E}\{L\} = \sum_{i=1}^{n} \mathrm{E}\left\{\log f_i(\boldsymbol{w}_i^T\boldsymbol{x})\right\} + \log|\det W| \tag{3.50}$$

Actually, if the f_i were equal to the actual distributions of $\boldsymbol{w}_i^T\boldsymbol{x}$, the first term would be equal to $-\sum_i H(\boldsymbol{w}_i^T\boldsymbol{x})$. Thus the likelihood would be equal, up to an additive constant, to the negative of mutual information as given in (3.43). Actually, in practice the connection is even stronger. This is because in practice we don't know the distributions of the independent components. A reasonable approach would be to estimate the density of $\boldsymbol{w}_i^T\boldsymbol{x}$ as part of the ML estimation method, and use this as an approximation of the density of si. In this case, likelihood and mutual information are, for all practical purposes, equivalent. Nevertheless, there is a small difference that may be very important in practice. The problem with maximum likelihood estimation is that the densities f_i must be estimated correctly. They need not be estimated with any great precision: in fact it is enough to estimate whether they are sub- or super-Gaussian (Cardoso et al, 1996; Hyvärinen et al., 1998c; Lee et al., 1999).

In many cases, in fact, we have enough prior knowledge on the independent components, and we don't need to estimate their nature from the data. In any case, if the information on the nature of the independent components is not correct, ML estimation will give completely wrong results. Some care must be taken with ML estimation, therefore. In contrast, using reasonable measures of non-Gaussianity, this problem does not usually arise.

3.4.5 ICA and Projection Pursuit

It is interesting to note how our approach to ICA makes explicit the connection between ICA and projection pursuit. Projection pursuit (Friedman et al., 2001; Friedman, 1987; Huber, 1985; Jones et al., 1987) is a technique developed in statistics for finding "interesting" projections of multidimensional data. Such projections can then be used for optimal visualization of the data, and for such purposes as density estimation and regression. In basic (1-D) projection pursuit, we try to find directions such that the projections of the data in those directions have interesting distributions, i.e., display some structure. It has been argued by Huber (1985) and by Jones et al. (1987) that the Gaussian distribution is the least interesting one, and that the most interesting directions are those that show the least Gaussian distribution. This is exactly what we do to estimate the ICA model.

3.5 Some ICA and PCA Algorithms and Their Problems

Many estimation methods for ICA requires prewhitening of observed signals, because it reduces the complexity of the ICA problems (Hyvärinen, Karhunen and Oja, 2001; Cichoki and Amari, 2002). In the case of fixed-point algorithms (Hyvärinen, Karhunen and Oja, 2001), it plays a significant role on the performance of the algorithms. In particular, Hyvärinen (1999) proposed FastICA fixed-point algorithm for robust BSS. However, the performance of this algorithm is not so good sometimes. A main cause of this weak performance may be from the classical prewhitening procedure, which is known to be sensitive to outliers. Thus estimate of independent components under classical prewhitening gives misleading results in presence of outliers or noisy data. There exist some robust prewhitening procedure like batch algorithm based on the subspace approach for ICA (Cichoki and Amari, 2002). However, this type of robust prewhitening may be suffer from the non-robust classical centering, (Hyvärinen et al., 2001, page 154). On the other hand, the performance of classical

prewhitening procedure is better than the performance of robust prewhitening procedure for noiseless data sets, while this performance is completely reverse for noisy data sets. It is also difficult task to know in advance whether a data set is noise free or not. Therefore, existing prewhitening procedures are not always suitable. This thesis discusses a new prewhitening procedure, named β-prewhitening by minimizing β-divergence from the adaptive robustness point of view.

Blind source separation (BSS) by independent component analysis (ICA) has played a significant role in signal processing problems, including speech enhancement, telecommunications, medical signal processing and suchlike. ICA aims to recover the original sources with independent and non-Gaussian structure from the observable mixture data. In the classical ICA model, only one hidden class is considered in the entire data space. We assume that there are several hidden classes of ICA models in the entire data space. Lee, Lewicki and Sejnowski (2000b) proposed a method for extracting all hidden classes of original sources from the entire data space based on the ICA mixture models. However, there exist one problem in their method is that the number c of classes need to know in advance which is very difficult task in practice.

In classical PCA model defined by (5.1) and (5.4), all latent vectors belong to only one source class $\mathscr{S}$, and all input vectors belong to the same class in the entire data space $\mathscr{D}$. However, in practice, these source vectors may originate from several source classes, and the corresponding observed vectors belong to several classes in the entire data space. In this case, the performance of classical PCA may not be so good. Gaussian Mixture Models (GMMs) may be used in this case. However, GMMs suffer from a serious drawbacks as the dimensionality of the problem space increases, the size of each covariance matrix becomes prohibitively large. This can be dealt with by assuming isotropic Gaussians (ie. ignoring the covariance structure) but this greatly reduces the flexibility of the model class. This problem has been solved by Tipping et al. (1999) who replaced each Gaussian with a probabilistic Principal Component Analysis (PCA) model. This allowed the dimensionality of each covariance to be effectively reduced whilst maintaining the richness of the model class. However, one problem encountered when applying this method is that the number of classes c should be known in advance, which is difficult in practice as early discussed also. In this

thesis, an attempt is made to propose iterative algorithm for local PCA and ICA both based on the minimum β-divergence method, where the number c of data clusters in the entire data space need not be known in advance.

3.5.1 Objectives of Our Study

The main objectives of this study are

1. To propose an adaptive robust prewhitening procedure for ICA by minimizing β-divergence.

2. To propose an adaptive algorithm for exploring local PCA structures by minimizing β-divergence using a local kernel function.

3. To extend minimum β-divergence method (Minami and Eguchi, 2002) for exploring local ICA structures.

Chapter 4

Robust Prewhitening for ICA by the Minimum β-Divergence Method

4.1 Defination of β-Divergence

The β-divergence between two pdf's $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ is defined as

$$D_\beta(p,q) = \int \left[\frac{1}{\beta} \left\{ p^\beta(\boldsymbol{x}) - q^\beta(\boldsymbol{x}) \right\} p(\boldsymbol{x}) - \frac{1}{\beta+1} \left\{ p^{\beta+1}(\boldsymbol{x}) - q^{\beta+1}(\boldsymbol{x}) \right\} \right] d\boldsymbol{x}, \quad \text{for } \beta > 0 \tag{4.1}$$

which is non-negative, that is $D_\beta(p(\boldsymbol{x}), q(\boldsymbol{x})) \geq 0$, equality holds iff $\mathrm{p}(\boldsymbol{x}) = \mathrm{q}(\boldsymbol{x})$, (cf. Minami et al. (2002)). We note that β-divergence reduces to Kullback Leibler (KL) divergence when $\beta \longrightarrow 0$, that is

$$\lim_{\beta \downarrow 0} D_\beta(p,q) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \, d\boldsymbol{x} = D_{\mathrm{KL}}(p,q). \tag{4.2}$$

4.2 Classical Prewhitening

Let us consider the linear ICA model for an observable vector $\boldsymbol{x}$ of dimension m as

$$\boldsymbol{x} = A\boldsymbol{s} \tag{4.3}$$

where $\boldsymbol{A} \in R^{m \times m}$ and $\boldsymbol{s}$ is an unobservable source vector whose components are independent and non-Gaussian. A random vector $\boldsymbol{z}$ is said to be white or sphere if $\mathrm{E}(\boldsymbol{z}) = 0$ and $\mathrm{E}(\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}) = \mathrm{I}_m$ (identity matrix). In the prewhitening procedure, vector $\boldsymbol{x}$ is processed by

$$\boldsymbol{z} = V^{-\frac{1}{2}}(\boldsymbol{x} - \boldsymbol{\mu}) \tag{4.4}$$

where $\boldsymbol{\mu}$ and V are the mean vector and covariance matrix of $\boldsymbol{x}$, respectively. It is also called sphering or spatial decorrelation. In the classical prewhitening, the mean vector $\boldsymbol{\mu}$ and the covariance matrix V are estimated by

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \quad \text{and} \quad \hat{V} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T \tag{4.5}$$

in a batch way sampling, respectively. Prewhitening of data is necessary in some adaptive ICA algorithms (Belouchrani et al., 2000; Hyvärinen et al., 2001; Cichoki et al., 2002; Choi et al., 2002) for Blind Source Separation (BSS), because it reduces the complexity of the ICA problems. With this connection, it is considered as the half ICA, see Hyvärinen, Karhunen and Oja (2001) for detailed discussion. In the case of fixed-point algorithms (Hyvärinen, 1999; Hyvärinen and Oja, 1997), it plays a significant role on the performance of the algorithms. In particular, Hyvärinen (1999) proposed FastICA fixed-point algorithm for robust BSS. However, the performance of this algorithm is not so good sometimes. A main cause of this weak performance may be from the classical prewhitening procedure, which is known to be sensitive to outliers. Thus estimate of independent components under standard prewhitening gives misleading results in presence of outliers or noisy data. There exist some robust prewhitening procedures for ICA (Belouchrani et al., 2000; Cichoki et al., 2002; Choi et al., 2002), however, this type of robust prewhitening might be suffer from the non-robust classical centering, (Hyvärinen et al., 2001, page 154). It may be well known that the performance of standard prewhitening procedure would be better than any other prewhitening procedures if data set is not corrupted by noise or outliers. On the other hand, if data set is corrupted by noise or outliers, then any robust prewhitening procedure would be better than standard prewhitening procedure. However, it would be very difficult to know in advance whether a data set is corrupted or not by outliers. In this condition, a researcher or user may feel inconvenience to select an appropriate algorithm for prewhitening. Therefore, existing prewhitening procedures would not always suitable. Based on the situation discussed above, we will propose a new prewhitening procedure in this chapter, named β-prewhitening by minimizing β-divergence from the adaptive robustness point of view. Section 4.3 offers a new prewhitening procedure and its robustness for ICA. We discuss a selection method for the tuning parameter β in section 4.3.2. In section 4.4, a measure of performance index is proposed for assessing prewhitening procedures. Section 4.5 presents numerical examples and section 4.6 contains the conclusions.

4.3 New Algorithm for Robust Prewhitening by Minimizing β-Divergence

Let us consider the space of unnormalized density functions rather than that of probability ones. Accordingly in this context the space of Gaussian density functions should be extended to $\kappa\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})$, where κ is a positive scalar and $\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})$ is the density function $N(\boldsymbol{\mu},\ V)$. The minimization of the β-divergence over the extended space may be lead as follows.

THEOREM 1: Let $p(\boldsymbol{x})$ be a probability density function of a random vector $\boldsymbol{X}$ and let

$$(\kappa, \boldsymbol{\mu}, V) = \mathrm{argmin}_{\kappa',\boldsymbol{\mu}',V'} D_\beta\left(p(\boldsymbol{x}),\ \kappa'\varphi_{\boldsymbol{\mu}',V'}(\boldsymbol{x})\right) \tag{4.6}$$

Then we have that

$$\kappa = \frac{\mathrm{E}_{p_{\boldsymbol{X}}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta\right\}}{\mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta\right\}}, \tag{4.7}$$

$$\boldsymbol{\mu} = \frac{\mathrm{E}_{p_{\boldsymbol{X}}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta \boldsymbol{X}\right\}}{\mathrm{E}_{p_{\boldsymbol{X}}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta\right\}}, \tag{4.8}$$

and

$$V = (\beta+1)\frac{\mathrm{E}_{p_{\boldsymbol{X}}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T\right\}}{\mathrm{E}_{p_{\boldsymbol{X}}}\left\{(\varphi_{\boldsymbol{\mu},V})^\beta\right\}}, \tag{4.9}$$

where the notations $_{p_{\boldsymbol{X}}}$ and $\varphi_{\boldsymbol{\mu},V}$ represent the p.d.f.'s $p(\boldsymbol{x})$ and $\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})$, respectively. The notation "E" means statistical expectation.

Proof: The gradients of D_β are

$$\frac{\partial}{\partial\kappa} D_\beta(p_{\boldsymbol{X}}, \kappa\varphi_{\boldsymbol{\mu},V}) = \kappa^{\beta-1}\int\left\{\kappa\varphi_{\boldsymbol{\mu},V}^{\beta+1} - p_{\boldsymbol{X}}\varphi_{\boldsymbol{\mu},V}^{\beta}\right\}d\boldsymbol{x},$$

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{\mu}} D_\beta(p_{\boldsymbol{X}}, \kappa\varphi_{\boldsymbol{\mu},V}) &= \kappa^{\beta}\int\left\{\kappa\varphi_{\boldsymbol{\mu},V}^{\beta+1} - p_{\boldsymbol{X}}\varphi_{\boldsymbol{\mu},V}^{\beta}\right\}V^{-1}(\boldsymbol{x}-\boldsymbol{\mu})d\boldsymbol{x} \\ &= -\mathrm{E}_{p_{\boldsymbol{X}}}\{(\kappa\varphi_{\boldsymbol{\mu},V})^\beta V^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\}\end{aligned}$$

and

$$\frac{\partial}{\partial V} D_\beta \left(p_{\boldsymbol{X}}, \kappa\varphi_{\boldsymbol{\mu},V}\right) = \frac{1}{2}V^{-1}\int \left\{(\kappa\varphi_{\boldsymbol{\mu},V})^{\beta+1} - p_{\boldsymbol{X}}(\kappa\varphi_{\boldsymbol{\mu},V})^{\beta}\right\} \times \left\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T - V\right\} V^{-1} d\boldsymbol{x},$$

since

$$\frac{\partial}{\partial \boldsymbol{\mu}} \log \varphi_{\boldsymbol{\mu}V}(\boldsymbol{x}) = V^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$$

and

$$\frac{\partial}{\partial V} \log \varphi_{\boldsymbol{\mu}V}(\boldsymbol{x}) = \frac{1}{2}V^{-1}\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T - V\}V^{-1}.$$

Equations (4.7) and (4.8) are easily obtained by solving $\frac{\partial D_\beta}{\partial \kappa} = 0$ and $\frac{\partial D_\beta}{\partial \boldsymbol{\mu}} = \mathbf{0}$, respectively. We now show (5.15). In fact $\frac{\partial D_\beta}{\partial V} = O$ leads to

$$\kappa \mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\left[\varphi^{\beta}_{\boldsymbol{\mu},V}\{(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T - V\}\right] = \mathrm{E}_{p_{\boldsymbol{X}}}\left[\varphi^{\beta}_{\boldsymbol{\mu},V}\{(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T - V\}\right],$$

which is, by substitution of (4.7),

$$\frac{\mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\{\varphi^{\beta}_{\boldsymbol{\mu},V}(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T\}}{\mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\{\varphi^{\beta}_{\boldsymbol{\mu},V}\}} = \frac{\mathrm{E}_{p_{\boldsymbol{X}}}\{\varphi^{\beta}_{\boldsymbol{\mu},V}(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T\}}{\mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\{\varphi^{\beta}_{\boldsymbol{\mu},V}\}} \tag{4.10}$$

In this way we observe that the left hand side of (4.10) is $(\beta+1)^{-1}V$, since $\frac{\varphi^{\beta+1}_{\boldsymbol{\mu},V}}{\mathrm{E}_{\varphi_{\boldsymbol{\mu},V}}\{\varphi^{\beta}_{\boldsymbol{\mu},V}\}}$ is just a Gaussian density with mean vector $\boldsymbol{\mu}$ and variance matrix $(\beta+1)^{-1}V$. This concludes (5.15).

Reweighted Moment Algorithm:

We will give an unsupervised leaning algorithm based on a random sample $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ from a probability density function $p(\boldsymbol{x})$. In general, for any integrable function $A(\boldsymbol{x})$, it holds that

$$\mathrm{E}_p\{A(\boldsymbol{X})\} \approx \frac{1}{n}\sum_{i=1}^{n} A(\boldsymbol{x}_i), \tag{4.11}$$

of which approximation can be probabilistically described by giving an explicit assumption for the data set. Hence we directly find the empirical form of (4.7), (4.8) and (5.15)

$$\kappa = \frac{1}{n}\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}_i)\}^{\beta}\{\det(2\pi V)\}^{\frac{\beta}{2}}(\beta+1)^{\frac{m}{2}}, \tag{4.12}$$

$$\frac{1}{n}\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}_i)\}^{\beta}(\boldsymbol{x}_i-\boldsymbol{\mu}) = \mathbf{0} \tag{4.13}$$

and

$$\frac{1}{n}\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}_i)\}^{\beta}\{(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^T - V\} = O. \tag{4.14}$$

From the empirical representation (4.12), (4.13) and (4.14), we can give a heuristic algorithm for solving jointly (4.7), (4.8) and (5.15) as follows:

$$\kappa_{t+1} = \frac{1}{n}\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu}_t,V_t}(\boldsymbol{x}_i)\}^{\beta}\{\det(2\pi V_t)\}^{\frac{\beta}{2}}(\beta+1)^{\frac{m}{2}}, \tag{4.15}$$

$$\boldsymbol{\mu}_{t+1} = \frac{\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu}_t,V_t}(\boldsymbol{x}_i)\}^{\beta}\boldsymbol{x}_i}{\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu}_t,V_t}(\boldsymbol{x}_i)\}^{\beta}}, \tag{4.16}$$

and

$$V_{t+1} = (\beta+1)\frac{\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu}_t,V_t}(\boldsymbol{x}_i)\}^{\beta}(\boldsymbol{x}_i-\boldsymbol{\mu}_t)(\boldsymbol{x}_i-\boldsymbol{\mu}_t)^T}{\sum_{i=1}^{n}\{\varphi_{\boldsymbol{\mu}_t,V_t}(\boldsymbol{x}_i)\}^{\beta}}. \tag{4.17}$$

In the prewhitening procedure (4.4), if we estimate mean vector $\boldsymbol{\mu}$ and covariance matrix V by (4.16) and (4.17), respectively, then it is said to be β-prewhitening.

In equations (4.16) and (4.17), the scaling factor $\{\varphi_{\mu,V}(\boldsymbol{x}_i)\}^{\beta}$ is considered as the weight of each data point $\boldsymbol{x}_i$, $i = 1, 2, ..., n$. For convenience of presentation, let us define a weight function, ϕ, using the scaling factor of this algorithm as

$$\phi(\boldsymbol{x}_i|\boldsymbol{\mu}, V) = \{\varphi_{\mu,V}(\boldsymbol{x}_i)\}^{\beta}, \quad i = 1, 2, ..., n. \tag{4.18}$$

Obviously, we see that this weight function is a function of Gaussian density for $\beta > 0$. Therefore, it significantly weights main population data points and insignificantly weights

outlier data points, because of outlier data points are usually far from the center of the main population. Thus, weight function 4.18 plays the key role for robust prewhitening. We note that if $\beta \longrightarrow 0$, then (4.15), (4.16) and (4.17) leads to the classical non-iteratative estimates as follows:

$$\hat{\kappa} = 1, \hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \hat{V} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T. \tag{4.19}$$

Then, β-prewhitening reduces to the standard prewhitening.

4.3.1 Robustness

We will investigate the robustness of our estimators by the influence function (IF). The influence function for the estimator $\boldsymbol{T}$ at $\boldsymbol{x}$ under the distribution F is defined as

$$\mathrm{IF}(\boldsymbol{x}; \boldsymbol{T}, F) = \lim_{t \downarrow 0} \frac{\boldsymbol{T}[(1-t)F + t\triangle_{\boldsymbol{x}}] - \boldsymbol{T}(F)}{t}, \tag{4.20}$$

where $\triangle_{\boldsymbol{x}}$ is the probability measure that puts mass 1 at the point $\boldsymbol{x}$. If the gross error sensitivity(GES), that is,

$$\sup_{\boldsymbol{x}} \mid \mathrm{IF}(\boldsymbol{x}; \boldsymbol{T}, F) \mid$$

is finite, then the estimator $\boldsymbol{T}$ is said to be B-robust under the distribution F. In our context the influence function for the estimator of the mean vector $\boldsymbol{\mu}$ and variance matrix V are obtained as

$$\mathrm{IF}(\boldsymbol{x}; \boldsymbol{T}_1, F) = -\boldsymbol{T}_1(F) + \frac{\left(\varphi_{\boldsymbol{\mu},V}\right)^{\beta} \boldsymbol{x}}{\mathrm{E}_{p\boldsymbol{x}}\left\{(\varphi_{\boldsymbol{\mu},V})^{\beta}\right\}} \tag{4.21}$$

and

$$\mathrm{IF}(\boldsymbol{x}; \boldsymbol{T}_2, F) = -\boldsymbol{T}_2(F) \ + (\beta+1)\frac{\left(\varphi_{\boldsymbol{\mu},V}\right)^{\beta} (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T}{\mathrm{E}_{p\boldsymbol{x}}\left\{(\varphi_{\boldsymbol{\mu},V})^{\beta}\right\}} \tag{4.22}$$

respectively, where $\boldsymbol{T}_1(F) = \boldsymbol{\mu}$ and $\boldsymbol{T}_2(F) = V$ satisfy equations (4.8) and (5.15). In both equations (4.21) and (4.22), $(\varphi_{\boldsymbol{\mu},V})^{\beta} = \{\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})\}^{\beta}$ is the weight function as described by equation (4.18) and $\mathrm{E}_{p\boldsymbol{x}}\left\{(\varphi_{\boldsymbol{\mu},V})^{\beta}\right\}$ is constant for each data point. Therefore, the GES for the influence functions (4.21) and (4.22) are bounded for $\beta > 0$, while the GES for both influence functions are unbounded for $\beta = 0$. Thus our estimators are B-robust. An empirical version of IF can be obtained by Tukey's (1970) Sensitivity Curve (SC) defined as

$$\mathrm{SC}_n(\boldsymbol{x}) = n\left[\boldsymbol{T}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n-1}, \boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n-1})\right] \tag{4.23}$$

Let $\boldsymbol{T}(F_n) = \boldsymbol{T}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ represent our estimators, where F_n is the empirical distribution of $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$, then sensitivity curve reduces to

$$\mathrm{SC}_n(\boldsymbol{x}) = \left[\boldsymbol{T}\left((1-\frac{1}{n})F_{n-1} + \frac{1}{n}\triangle_x\right) - \boldsymbol{T}(F_{n-1})\right]\frac{1}{n} \tag{4.24}$$

which converges to $\mathrm{IF}(\boldsymbol{x}; \boldsymbol{T}, F)$ for $n \longrightarrow \infty$ Hampel et al. (1986). To investigate the perfor-

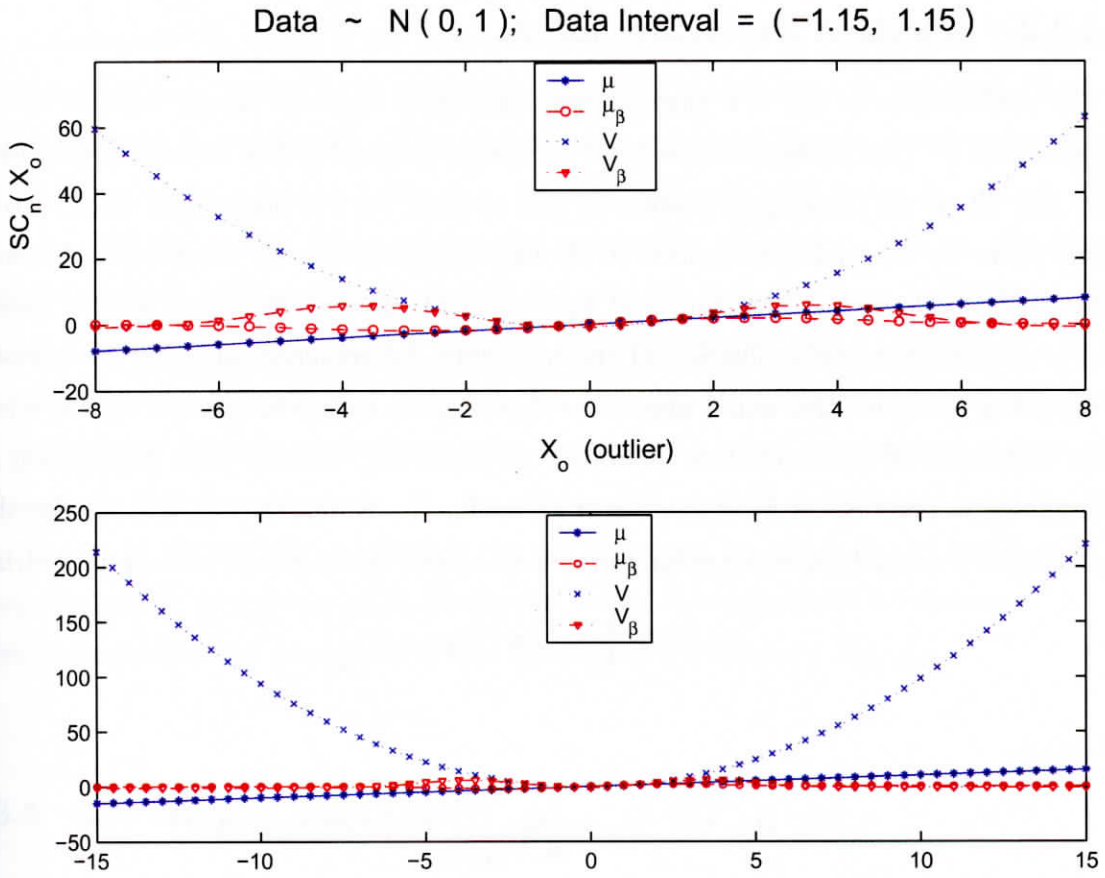

Figure 4.1: Tukey's sensitivity curve for classical estimators of mean and variance, and our proposed estimators of mean and variance based on β-divergence

mance of the proposed estimators by the Tukey's sensitivity curve, we consider a sample of data from N(0,1) to estimate mean and variance including outliers or noise, where original data points lies between -1.5 to +1.5. Figure 4.1 shows the Tukey's sensitivity curve for classical estimators of mean $(\boldsymbol{\mu})$ and variance (V), and our proposed estimators of mean $(\boldsymbol{\mu}_\beta)$ and variance (V_β) based on β-divergence. The marker style '*' and '$\times$' represents the curves for classical estimators of mean and variance, respectively, while the marker style 'o'

and '∇' represents the curves for our proposed estimators of mean and variance, respectively. From figure 4.1 clearly we see that GES for classical estimators are unbounded, while the GES for our proposed estimators are bounded and reduce to almost zero for larger outliers. Thus our proposed estimators for mean and variance are B-robust.

4.3.2 Selection Procedure for β

The performance of our new prewhitening procedure depends on the value of the tuning parameter β. This performance is good for a wide range of β. Let us define this wide range by R_β. To obtain better performance by this method, we will propose an adaptive selection procedure for the tuning parameter β. To find an appropriate β, we evaluate the estimates by various values of β. Minami and Eguchi (2003) used β-divergence with a fixed value of β as a measure for evaluation of the minimum β-divergence estimator for robust BSS. Following them, we also would like to use β-divergence with a fixed value of β as a measure for evaluation of our estimators for robust prewhitening, because data distribution $p(\boldsymbol{x})$ is unknown in practice. A fixed value of β for evaluation is denoted by $\beta_0 \in R_\beta$. We define a measure for evaluation of our estimators for the mean vector $\boldsymbol{\mu}$ and covariance matrix V as:

$$D_{\beta_0}(\beta) = \mathrm{E}\left\{ D_{\beta_0}\left(p_{\boldsymbol{X}, \hat{\kappa}_\beta \varphi_{\widehat{\boldsymbol{\mu}}_\beta, \widehat{V}_\beta}}(\boldsymbol{X}) \right) \right\} \tag{4.25}$$

where

$$\begin{aligned} (\widehat{\kappa}_\beta,\ \widehat{\boldsymbol{\mu}}_\beta,\ \widehat{V}_\beta) &= \underset{\kappa,\ \boldsymbol{\mu},\ V\ (\boldsymbol{x}\in\mathscr{D})}{\operatorname{argmin}} D_\beta\big(p(\boldsymbol{x}), \kappa\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})\big) \\ &= \underset{\kappa,\ \boldsymbol{\mu},\ V\ (\boldsymbol{x}\in\mathscr{D})}{\operatorname{argmax}} L_\beta(\boldsymbol{x}; \kappa, \boldsymbol{\mu}, V). \end{aligned}$$

Here

$$\begin{aligned} L_\beta(\boldsymbol{x}; \kappa, \boldsymbol{\mu}, V) &= \frac{1}{n\beta} \sum_{\boldsymbol{x}\in\mathscr{D}} \left\{ \kappa\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}) \right\}^\beta \\ &\quad - \frac{\kappa^{\beta+1}}{(\beta+1)^{(m+2)/2} \left\{ \det(2\pi V) \right\}^{\beta/2}}, \end{aligned}$$

and $\mathscr{D}$ is the data set. Then (5.52) can be simplified as

$$D_{\beta_0}(\beta) = -\mathrm{E}\left\{ L_{\beta_0}(\boldsymbol{X}; \hat{\kappa}_\beta, \widehat{\boldsymbol{\mu}}_\beta, \widehat{V}_\beta) \right\} \tag{4.26}$$

where

$$L_{\beta_0}(\boldsymbol{x};\hat{\kappa}_\beta,\widehat{\boldsymbol{\mu}}_\beta,\widehat{V}_\beta) = \frac{1}{n\beta_0}\sum_{\boldsymbol{x}\in\mathscr{D}}\left\{\hat{\kappa}_\beta\varphi_{\widehat{\boldsymbol{\mu}}_\beta,\widehat{V}_\beta}(\boldsymbol{x})\right\}^{\beta_0} - \frac{(\hat{\kappa}_\beta)^{\beta_0+1}}{(\beta_0+1)^{(m+2)/2}\left\{\det(2\pi\widehat{V}_\beta)\right\}^{\beta_0/2}}$$

The measurement $D_{\beta_0}(\beta)$ is of the generalization performance of a estimator. The generalization performance relates to its prediction capability on independent test data. If we use the same dataset to evaluate $D_{\beta_0}(\beta)$ as to estimate a recovering matrix, it will underestimate $D_{\beta_0}(\beta)$. If we are in a data-rich situation, the best approach is to divide the dataset into a few parts, and use one set for estimation and another for evaluation. In other situations, a simple and widely used method by sample re-use is the **K-fold Cross Validation (CV)** method (Hastie et al., 2001). The K-fold CV method uses part of the available data to find the estimate and a different part to test it. For the current problem, we employ the K-fold CV method as a generalization scheme. We split the data into K approximately equal-sized and similarly distributed sections. For the k th section, we find the estimate using the other $K-1$ parts of the data, and calculate the β_0-divergence for the k th section of the data. Then we combine the calculated β_0-divergence values to obtain the CV estimate.
Table 1 summarizes the procedure to find the K-fold CV estimate $\widehat{D}_{\beta_0}(\beta)$.

4.3.3 Deciding β Adaptively

We compute

$$\mathrm{SD}_{\beta_0}(\beta) = \text{the standard error of } \frac{1}{|\mathscr{D}(k)|}\mathrm{CV}_{(k)},$$

as a measure for the variation of $\widehat{D}_{\beta_0}(\beta)$, where $|\mathscr{D}(k)|$ denotes the number of elements in the k-th part of data $\mathscr{D}(k)$. Plots of $\widehat{D}_{\beta_0}(\beta)$ for β with the auxiliary boundary curves $\widehat{D}_{\beta_0}(\beta)\pm\mathrm{SD}_{\beta_0}(\beta)$ will help us to select an optimum β. We denote this optimum β by β_{opt}. We often have to employ the upper auxiliary boundary curve (UABC) with the $\widehat{D}_{\beta_0}(\beta)$ curve to choose β_{opt}. If the curve of $\widehat{D}_{\beta_0}(\beta)$ is flat for a wide range of β, then $\beta_{\mathrm{opt}}=0$. When more than one data class or outliers exist in the entire data space, typical shapes of curves of $\widehat{D}_{\beta_0}(\beta)$ that enables us to chose an appropriate value β are elbow and dipper shapes. So, if the curve does not have these shapes, we increase the value of β_0. If these shapes do not

Table 4.1: K-fold Cross Validation procedure

Split the data set $\mathscr{D}$ into K subsets; $\mathscr{D}(1), \cdots, \mathscr{D}(K)$.

Let $\mathscr{D}^{-k} = \{\boldsymbol{x} | \boldsymbol{x} \notin \mathscr{D}(k)\}$.

For $k = 1, \cdots, K$

- Estimate κ, $\boldsymbol{\mu}$ and V by minimizing $D_\beta\big(p(\boldsymbol{x}), \kappa\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})\big)$ using $\mathscr{D}^{-k}$,
  $(\widehat{\kappa}_\beta,\ \widehat{\boldsymbol{\mu}}_\beta,\ \widehat{V}_\beta) = \underset{\kappa,\ \boldsymbol{\mu},\ V\ (\boldsymbol{x}\in\mathscr{D}^{-k})}{\operatorname{argmax}} L_\beta(\boldsymbol{x}; \kappa, \boldsymbol{\mu}, V)$
- Compute $\mathrm{CV}_{(k)}$ using $\mathscr{D}(k)$,
  $\mathrm{CV}_{(k)} = -L_{\beta_0}(\boldsymbol{x};\ \widehat{\kappa}_\beta,\ \widehat{\boldsymbol{\mu}}_\beta,\ \widehat{V}_\beta)$

End

Then, $\widehat{D}_{\beta_0}(\beta) = \dfrac{1}{n}\sum_{k=1}^{K} \mathrm{CV}_{(k)}$

$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}}\ \widehat{D}_{\beta_0}(\beta)$

---

appear for any β_0, then $\beta_{\mathrm{opt}} = 0$, (Mollah, Minami and Eguchi, 2005). If the curve of $\widehat{D}_{\beta_0}(\beta)$ has an elbow or dipper shape, we choose the smaller one instead of the smallest β as the β_{opt} whose evaluated value $\widehat{D}_{\beta_0}(\beta_{\mathrm{opt}})$ is not larger than the value of UABC that corresponds to the smallest value of $\widehat{D}_{\beta_0}(\beta)$. However, there is no theoretical justification for this rule, which is known as the one-standard error rule (Hastie et al., 2001). Note that fixed β_0 should be larger than optimum β.

4.4 Performance Index

Let us now discuss a measure of performance index (PI) for assessing prewhitening procedures. The purpose of pre-whitening is to find a prewhitening matrix $W = V^{-1/2}$ such that the global mixing matrix $H = WA$ satisfis

$$H^T H = kI, \tag{4.27}$$

where $k > 0$ and I is the $m \times m$ identity matrix Hyvärinen et al. (2001). If $k = 1$, then H is said to be perfectly orthogonal and W is said to be standard prewhitening matrix. If data set is corrupted by noise or outliers and W is estimated by standard prewhitening

procedure, then global mixing matrix H is deviated from orthogonality and does not satisfy (4.27). These deviations could be a good measure of performance of prewhitening. If H satisfies (4.27), then absolute value of each eigen value $\lambda_i, (i = 1, 2, ..., m)$ of H is k, that is $|\lambda_i| = k$ for all i. Therefore, a general performance index (PI) may be defined by

$$\text{PI} = \frac{(\sum_{i=1}^{m} |\lambda_i|)^2}{m \sum_{i=1}^{m} |\lambda_i|^2}, \tag{4.28}$$

Obviously, PI = 1, if $|\lambda_i| = k$ for all i; otherwise $0 < \text{PI} < 1$. Therefore, performance of a prewhitening procedure is ideal if PI = 1.

4.5 Numerical Examples

We investigate the performance of β-prewhitening in a comparison of the standard prewhitening by a performance index proposed in (4.28) as well as by FastICA using both synthetic and real data sets.

4.5.1 Simulation With Randomly Generated Synthetic Data

Two-dimensional 1000 random samples were drawn from uniform distribution with zero mean vector such that components of each source vector are independent of each other. Then we mixed this source data set by a mixing matrix

$$A = \begin{pmatrix} 0.13 & 0.38 \\ 0.48 & -0.72 \end{pmatrix}.$$

Figure 4.2a shows the scatter plot of mixed signals. Then, we added two-dimensional 50, 100, 200, 500 and 1000 sizes of outliers in the data set that is shown in Figure 4.2a. Figures 4.2b-4.2f represent scatter plots of mixed data points (.) in presence of 50, 100, 200, 500 and 1000 sizes of outliers (+), respectively. To investigate the performance of β-prewhitening in a comparison of standard prewhitening, we estimated prewhitening matrix $W = V^{-1/2}$ and global mixing matrix $H = WA$ for each data set described in Figures 4.2a-4.2f by both procedures. Then we computed performance indexes (PI) by (4.28) for both procedures. Figure 4.2g shows PI for each data set described in Figures 4.2a-4.2f, respectively. The solid line with marker style (*) represent the performance of standard prewhitening and the dashed

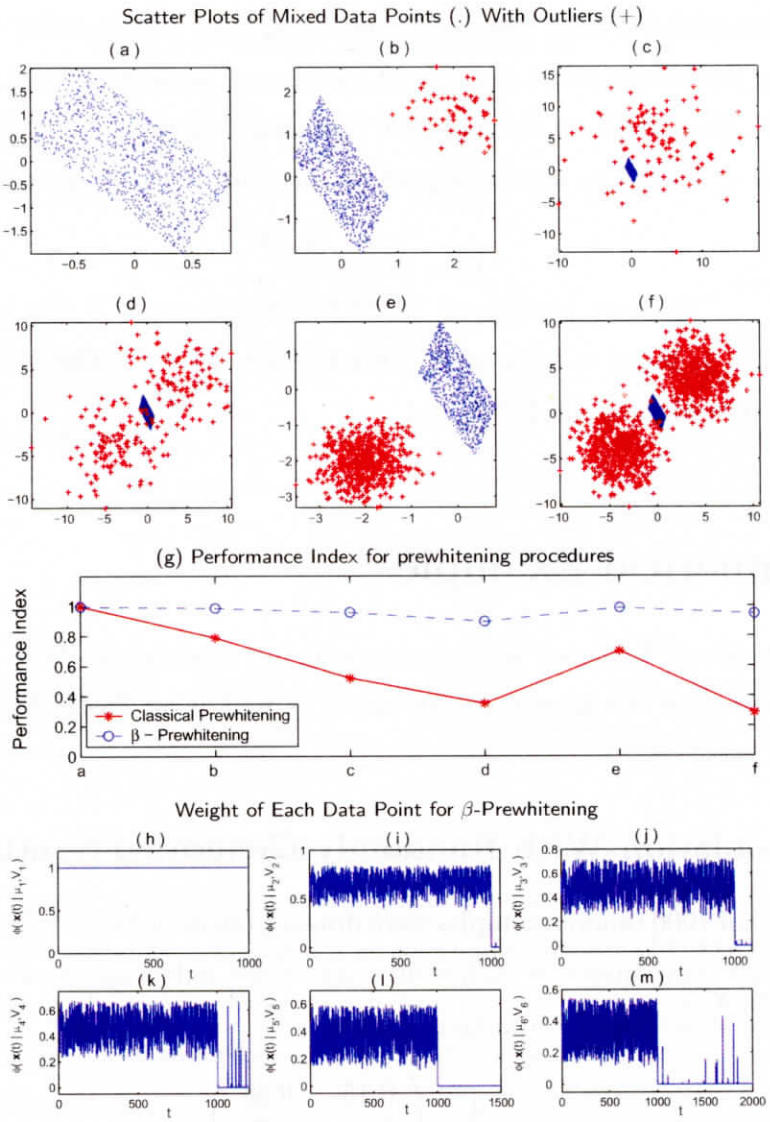

Figure 4.2: (a-f) Scatter plots of mixed data points (.) in presence of 0, 50, 100, 200, 500 and 1000 outliers (+), respectively. (g) Performance index for prewhitening procedures with data sets (a-f), respectively. (h-m) Weight of each data point for β-prewhitening with data sets (a-f), respectively.

line with marker style (o) represent the performance of β-prewhitening. We see that PI is 1 by both methods only for noise or outlier free data set that is described in Figure 4.2*a*. For other data sets those are described in Figures 4.2b-4.2f, PI is far from 1 for standard prewhitening, however, PI is almost close to 1 for β-prewhitening. Therefore, performance of both methods are same if data set is not corrupted by noise or outliers, however, if data set is corrupted by noise or outliers, then performance of β-prewhitening is better than

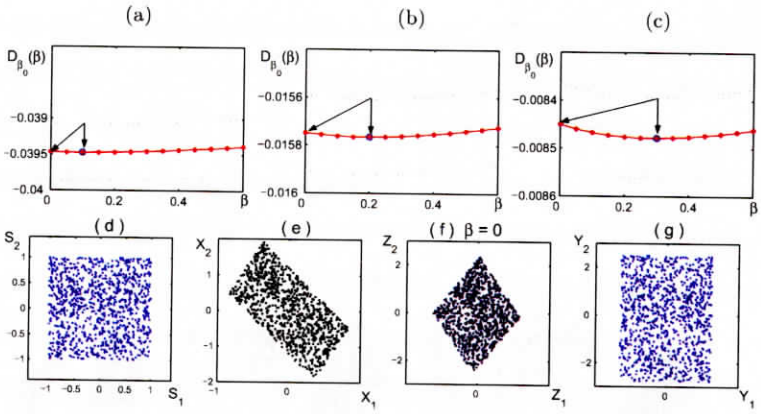

Figure 4.3: (a-c) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.1$, 0.2 and 0.3 for different values of β. (d) Scatter plot of source signals. (e) Scatter plot of mixed signals. (f) Scatter plot of whitened signals (g) Scatter plot of recovered signals by FastICA

standard prewhitening. Figures 4.2h-4.2m represent the weight of each data point obtained by (4.18) for data sets those are described in Figures 4.2a-4.2f, respectively. We see that weight of each data point for data set (Figure 4.2b) is exactly 1, this means β-prewhitening reduces to standard prewhitening for this data set. For other data sets, weight of each mixed data point is significantly larger, while weight of each outlier data point is almost close to zero. This means that outlier data points have no influence in the estimation by the proposed method. In Figures 4.2k and 4.2m, we see that weights corresponding to some outlier data points are larger, however, these outlier data points are overlapped or very close to mixed data points, so estimate is not affected so much by those data points. Note that first 1000 mixed data points for each data set are same and the rest are Gaussian noise or outliers.

In order to investigate the performance of β-prewhitening in a comparison of standard prewhitening by FastICA, we consider the data set that is already described in Figure 4.2a. For convenience of presentation, we display this mixed data set again in Figure 6.1e. To obtain whiten data from this mixed data set by the proposed method, we selected the values of the tuning parameter β by K-fold CV ($K = 10$). We computed $\widehat{D}_{\beta_0}(\beta)$ for several values of β with a fixed value of β_0 using the algorithm given in table 1. We computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 0.6 by 0.05 with $\beta_0 = 0.1, 0.2$ and 0.3 using the algorithm given in table 1. Figures 6.1a-6.1c show the plots of $\widehat{D}_{\beta_0}(\beta)$. In each plot, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are

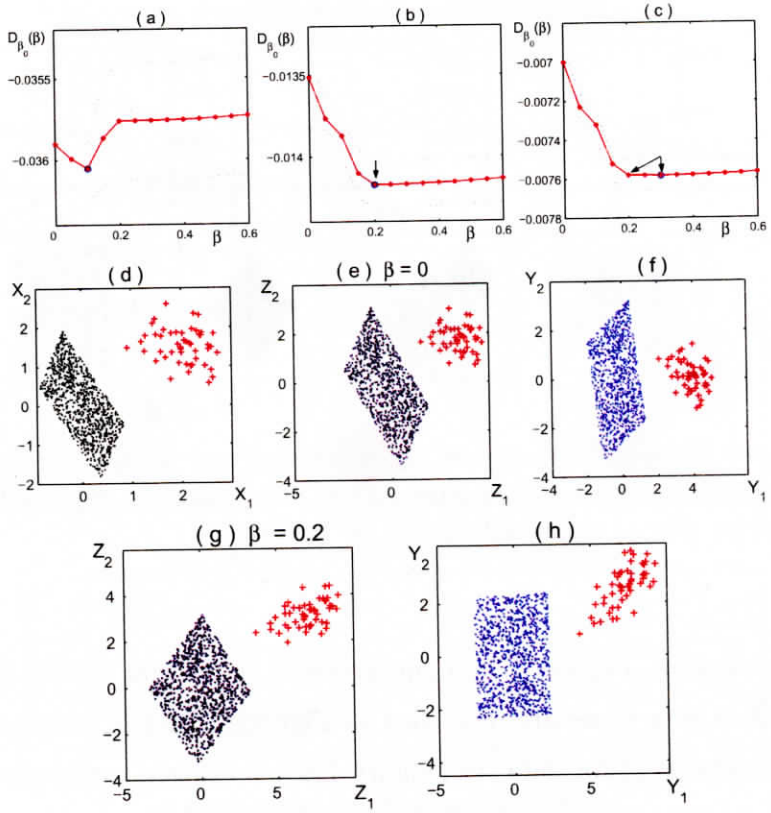

Figure 4.4: (a-c) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.1$, 0.2 and 0.3 for different values of β. (d) Scatter plot of mixed signals in presence outliers (+). (e) Scatter plot of whitened signals under standard prewhitening (f) Scatter plot of recovered signals by FastICA under standard prewhitening (e) Scatter plot of whitened signals under β-prewhitening with $\beta = 0.2$. (f) Scatter plot of recovered signals by FastICA under β-prewhitening with $\beta = 0.2$.

$\widehat{D}_{\beta_0}(\beta) \pm \mathrm{SD}_{\beta_0}(\beta)$. Plots of $\widehat{D}_{\beta_0}(\beta)$ shown in figures 6.1a-6.1c suggest $\beta = 0$ for each β_0 by 'one standard error' rule for β-prewhitening, which is equivalent to standard prewhitening. Thus adaptive selection procedure for β suggest standard prewhitening if data set is not corrupted by noise or outliers. Figure 6.1f shows the scatter plot of whitened signals under standard prewhitening. Figure 6.1g shows the scatter plot of recovered signals by FastICA under standard prewhitening. Comparing figures 6.1d and 6.1g, we see that recovered signals are independent with each other with non-Gaussian structure.

To investigate the performance of β-prewhitening on robustness, we added 50 outliers (+) from Gaussian distribution that is already described in 4.2b. For convenience of discussion, we display this mixed data set again in Figure 4.4d. To obtain the whiten data by

β-prewhitening, we selected the values of the tuning parameter β by K-fold CV (K=10) as before. We computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 0.6 by 0.05 with $\beta_0 = 0.1, 0.2$ and 0.3. Figures 4.4a-4.4c show the plots of $\widehat{D}_{\beta_0}(\beta)$. In each plot, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are $\widehat{D}_{\beta_0}(\beta) \pm \mathrm{SD}_{\beta_0}(\beta)$. For $\beta_0 = 0.2$ and 0.3, plots of $\widehat{D}_{\beta_0}(\beta)$ shown in Figures 4.4b-4.4c have elbow and dipper shapes, and suggest β=0.2 by the 'one standard error' rule. So, we choose β=0.2 for β-prewhitening, which is different from the standard prewhitening. To investigate the performance of β-prewhitening in a comparison of standard prewhitening by FastICA, first we computed whiten data set by classical method. We recovered original signals by FastICA using classical prewhitened data set. Figures 4.4e-4.4f show the scatter plots of whitened signals and recovered signals under standard prewhitening, respectively. Then we computed whiten data set by β-prewhitening. We recovered original signals by FastICA using β-prewhitened data set. Figures 4.4g-4.4h show the scatter plots of whitened signals and recovered signals under β-prewhitening with β=0.2, respectively. Comparing Figures 6.1f, 4.4e and 4.4g, we observe that whitened signals in absence of outliers by standard prewhitening are not similar to the whitened signals in presence of outliers by standard prewhitening, while whitened signals in absence of outliers by classical prewhitening are almost similar to the whitened signals in presence of outliers by β-prewhitening with β=0.2 . Similarly, comparing Figures 6.1g, 4.4f and 4.4h, we see that recovered signals in absence of outliers by FastICA under standard prewhitening are not similar to the recovered signals in presence of outliers by FastICA under standard prewhitening, while recovered signals in absence of outliers by FastICA under standard prewhitening are almost similar to the recovered signals in presence of outliers by FastICA under β-prewhitening with β=0.2. Therefore, β-prewhitening is much better than standard prewhitening when outliers exist; otherwise, it keeps equal performance.

To demonstrate the validity of the proposed method for high dimensional data analysis, we generated the following data sets:

Dataset-I: 100 times, 20-dimensional 1000 random vectors were drawn from **uniform distribution** with zero mean vector such that components of each vector are independent of each other. Each time identity matrix (I) was used as the mixing matrix A and 20-dimensional 100 Gaussian random vectors were included as outliers or noises.

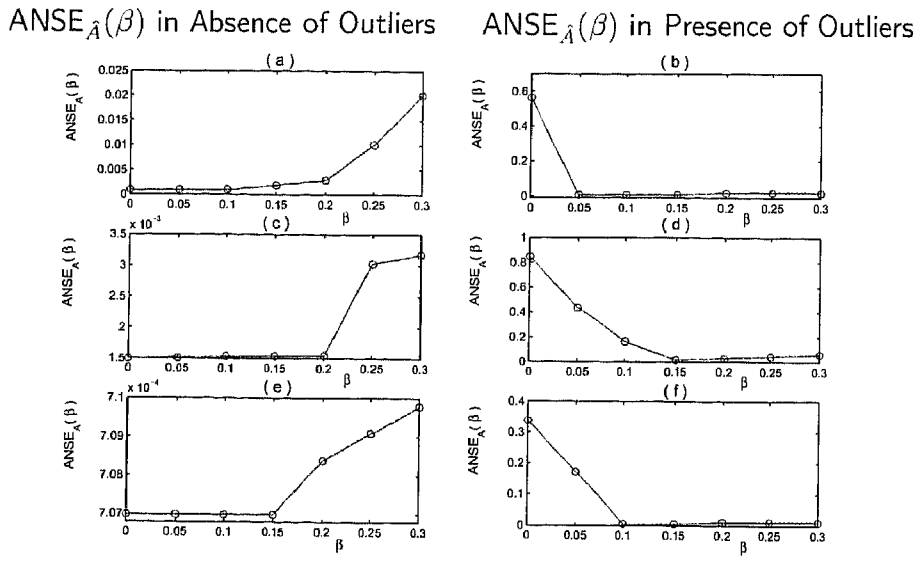

Figure 4.5: (a) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset I without noise. (b) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset I with noise. (c) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset II without noise. (d) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset II with noise. (e) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset III without noise. (f) Plots of ANSE$_{\hat{A}}(\beta)$ for dataset III with noise.

Dataset-II: 100 times, 20-dimensional 1000 random vectors were drawn from **Laplace distribution** with zero mean vector such that components of each vector are independent of each other. Each time identity matrix (I) was used as the mixing matrix A and 20-dimensional 100 Gaussian random vectors were included as outliers or noises as before.

Dataset-III: 100 times, 20-dimensional 500 random vectors were drawn from each of **uniform distribution** and **Laplace distribution** with zero mean vector such that components of each vector are independent of each other. Each time identity matrix (I) was used as the mixing matrix A and 20-dimensional 100 Gaussian random vectors were included as outliers or noises as before.

In order to investigate the performance of β-prewhitening on FastICA in-depth, we computed the average of norm square error (ANSE) for estimating the mixing matrix $A = [A_1, A_2, ..., A_m] = I_m$ (identity matrix) by a measure defined by

$$\text{ANSE}_{\hat{A}}(\beta) = \frac{1}{rm}\sum_{i=1}^{m}\sum_{j=1}^{r}\|A_i - \hat{A}_{ij}(\beta)\|^2$$

where $\hat{A}_{ij}(\beta)$ is the estimate of i-th column of A for j-th data set ($j = 1, 2, ..., r$) by Fas-

tICA under β-prewhitening. Obviously $\text{ANSE}_{\hat{A}}(\beta) \geq 0$, equality hold iff $A_i = \hat{A}_{ij}(\beta)$. We computed $\text{ANSE}_{\hat{A}}(\beta)$ for several values of β varying from 0 to 0.3 by 0.05 with m=20 and r=100. Figures 4.5a, 4.5c and 4.5e show the results of $\text{ANSE}_{\hat{A}}(\beta)$ for data sets I-III in absence of outliers, respectively. We see that performance of β-prewhitening with $\beta = 0$ or standard prewhitening on FastICA is better than β-prewhitening with $\beta > 0$ if data set is not corrupted by noises or outliers. On the other hand, Figures 4.5b, 4.5d and 4.5f show the results of $\text{ANSE}_{\hat{A}}(\beta)$ for data sets I-III in presence of outliers, respectively. Then we see that performance of β-prewhitening with $\beta > 0$ on FastICA is much better than standard prewhitening if data set is corrupted by noises or outliers.

4.5.2 Simulation With Synthetic Signals

For a practical example of our method, we have taken 4 independent signals (sinusoid, funny curve, saw-tooth and impulsive noise) from the Hyvärinen's FastICA project Hyvärinen (A.) shown in figure 4.7(a). Note that sinusoid and saw-tooth signals are sub-Gaussian signals and funny curve and impulsive noise are super Gaussian signals. In the previous examples, outliers were considered far from the data center, where β-prewhitening with $\beta > 0$ was batter than classical prewhitening for ICA by FastICA. Now we would like to investigate the case, where outliers occur close to the data center. For this, we consider independent signals, where one is sinusoid and the other one is impulsive noise shown in figure 4.6(a). We mixed them linearly by a non-singular mixing matrix and add 40 outliers (+) at the end of mixed data points. Figure 4.6(b) represent the mixed signals and their scatter plot (right). To obtain whiten data by the proposed method, we selected the value of the tuning parameter β by K-fold CV (K=10) same as above. Using the plots of $\widehat{D}_{\beta_0}(\beta)$ shown in Figures 4.6(c), we choose $\beta = 0$ by the 'one standard error' rule, which is equivalent to the classical prewhitening. First two signal in Figures 4.6(d) are the recovered signals by FastICA under classical prewhitening, and the right plot represents the scatter plot of recovered signals. We see that recovered signals are almost similar to the original signals. Also K-fold CV plot suggests that one can get the similar result for a wide range of β.

To investigate the performance with the multidimensional data set, we mixed 4 independent

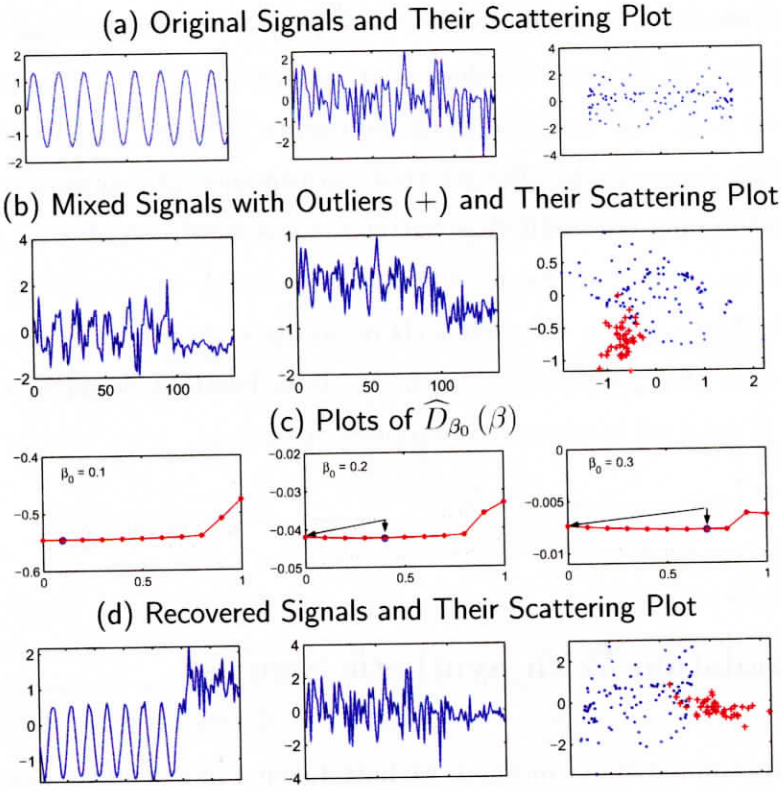

Figure 4.6: Plots for simulation results with the mixture of two synthetic signals. (a) Original signals (left & middle) and their scattering plots (right). (b) Mixed signals (left & middle) and their scattering plots (right). (c) Plots of $\widehat{D}_{\beta_0}(\beta)$ to select appropriate values of β for β-prewhitening. (d) Recovered signals under β-prewhitening with $\beta = 0$ (left & middle) and their scattering plots (right).

signals (sinusoid, funny curve, saw-tooth and impulsive noise) by a mixing matrix

$$A = \begin{pmatrix} 0.9507 & 0.1054 & 0.8403 & 0.6823 \\ -0.5303 & 0.2808 & 0.1429 & 0.0518 \\ 0.7445 & 0.0552 & 0.9581 & -0.6252 \\ 0.0234 & -0.9187 & 0.7388 & 0.2918 \end{pmatrix}$$

and added outliers (+) from N(10, 1) and N(-10, 1) with probability of occurrence 0.05 at the end of each mixed signal. Figures 4.7(b) shows the mixed signals and Figure 4.7(c) represent the scatter plot of mixed signal. For whitening the data set by the proposed method, we selected the values of the tuning parameter β by K-fold CV (K=10) as in the previous example. we computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 0.3 by 0.05 with $\beta_0 = 0.1, 0.2$ and 0.3 using the algorithm given in table 1. Figure 4.8(a) show the plots of $\widehat{D}_{\beta_0}(\beta)$. In each plot, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk.

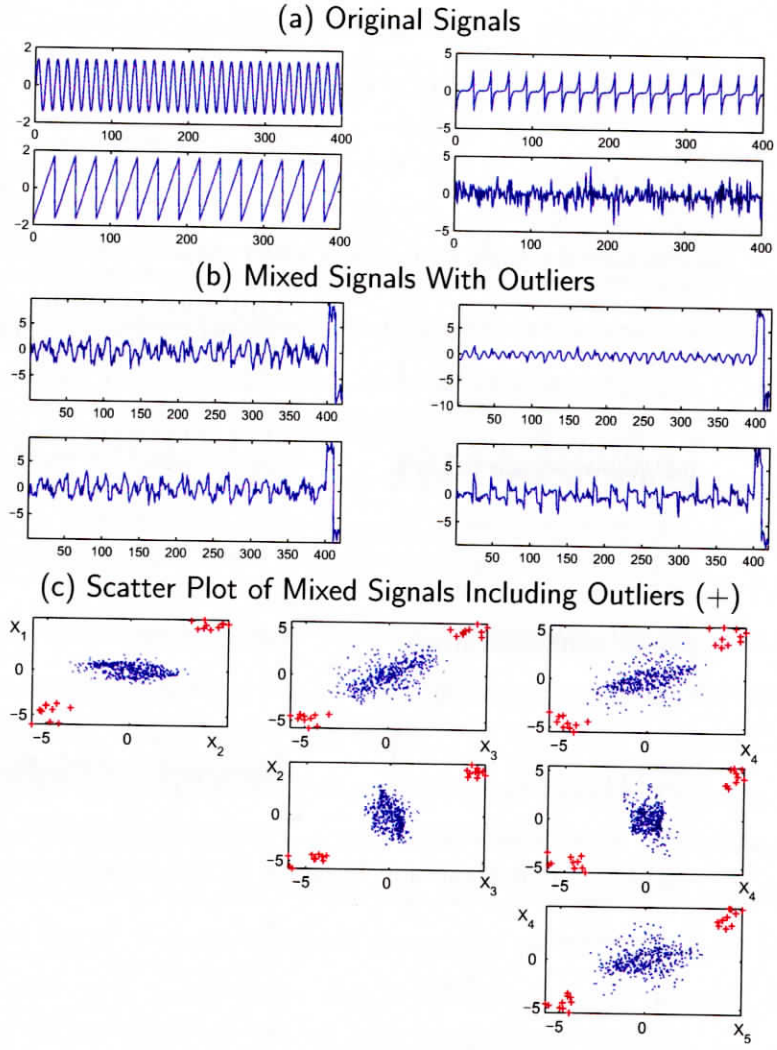

Figure 4.7: Plots for simulation results with the mixture of 4 synthetic signals. (a) Original signals (b) Mixed signals (c) Scatter plot of mixed signals including outliers (+).

Dotted lines are $\widehat{D}_{\beta_0}(\beta) \pm 2\widehat{\mathrm{D}}_{\beta_0}(\beta)$. For β_0=0.1 and 0.2, plots of $\widehat{D}_{\beta_0}(\beta)$ shown in figure 4.8(a) look like an elbow shape and suggest β=0.05 by the 'one standard error' rule, while for β_0=0.3 plot of $\widehat{D}_{\beta_0}(\beta)$ shown in figure 4.8(c) suggest β=0.3 by the same rule. Clearly β=0.05 is more stable than β=0.3 for wide range of β_0. From figure 4.8(d), we see that estimated mean and variance for $\beta = 0.05$ to 0.25 are almost consistent, but for other values of β estimates are drastically changed. Therefore we decided β=0.05 for β-prewhitening. Figure 4.8(b) show the recovered signals under classical prewhitening or β-prewhitening with β=0, while figure 4.8(c) shows the recovered signals under β-prewhitening with β=0.05. Comparing figures 4.7(a), 4.8(b) and 4.8(c), we see that recovered signals by FastICA under classical

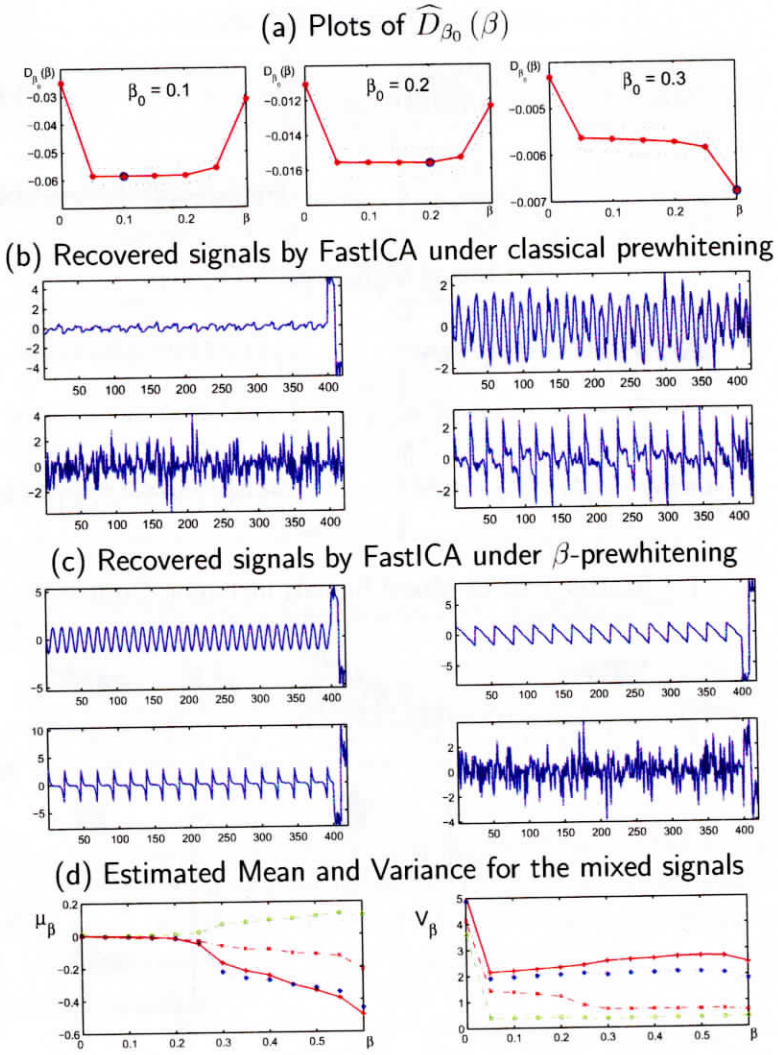

Figure 4.8: Plots for simulation results with the mixture of synthetic data set. (a) Noisy mixed signals (b) Plots of $\widehat{D}_{\beta_0}(\beta)$ to select appropriate values of β for β-prewhitening. (c) Recovered signals by FastICA under classical prewhitening. (d) Recovered signals under β-prewhitening with β=0.05. (e) Plots of the estimated mean ($\hat{\mu}_\beta$) and variance ($\hat{V}_\beta$) for the mixed signals by the proposed method.

prewhitening are not good, while the recovered signals under β-prewhitening with β=0.05 are good and almost similar to the original signals.

4.5.3 Simulation With Real Audio Signals

We have taken 3 independent audio signals shown in figure 4.9(i) as original signals for a practical example with real data. In figure 4.9(i), first two are speech signals and last one is

music signal. We mixed this source signals by a mixing matrix

$$A = \begin{pmatrix} 0.8678 & 0.4315 & -0.9657 \\ 0.3293 & -0.2605 & 0.9485 \\ -0.4880 & 0.9830 & 0.4517 \end{pmatrix}$$

Figure 4.9(ii) shows the mixed signals (data set 7). For whitening the mixture of audio signals without noise by our method, we selected the values of the tuning parameter β by K-fold CV (K=25) as in the previous example. We computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 0.3 by 0.05 with $\beta_0 = 0.1, 0.2$ and 0.3. Figures 4.9(iii(a-c)) show the plots of $\widehat{D}_{\beta_0}(\beta)$. In each plot, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are $\widehat{D}_{\beta_0}(\beta) \pm \mathrm{SD}_{\beta_0}(\beta)$. For β_0=0.1 and 0.2, plot of $\widehat{D}_{\beta_0}(\beta)$ shown in figures 4.9(iii(a-b)) suggest β=0 by the 'one standard error' rule, while for β_0=0.3, plot of $\widehat{D}_{\beta_0}(\beta)$ shown in figure 4.9(iii(c)) suggest β=0.15 by the same rule. Obviously β=0 is more stable than β=0.15 for wide range of β_0. Therefore we choose β=0 for β-prewhitening. Figures 4.9(iv) shows the recovered signals under β-prewhitening with β=0. Comparing figures 4.9(i) and 4.9(iv), we see that recovered audio signals by FastICA under β-prewhitening with β=0 are almost similar to the original audio signals. To investigate the performance on robustness with the mixture of audio signals, we added Gaussian noise from N(1.3, 0.2) and N(-1.3, 0.2) with probability of occurrence 0.05 at the end of each mixed signal. Figures 4.10(a-b) represents the mixed signals (data set 7^+) and their pairwise scatter plot respectively. For β-prewhitening, we selected the values of the tuning parameter β by K-fold CV (K=25) as in the previous example. we computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 0.3 by 0.05 with $\beta_0 = 0.1, 0.2$ and 0.3 using the algorithm given in table 1. Figure 4.10(c) shows the plots of $\widehat{D}_{\beta_0}(\beta)$ for $\beta_0 = 0.1, 0.2$ and 0.3, respectively. In each plot, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are $\widehat{D}_{\beta_0}(\beta) \pm \mathrm{SD}_{\beta_0}(\beta)$. For β_0=0.1 0.2 and 0.3, plot of $\widehat{D}_{\beta_0}(\beta)$ look like an elbow shape and suggest β=0.05 by the 'one standard error' rule. Therefore we choose β=0.05 for β-prewhitening. Figure 4.11(a-b) shows the recovered signals under classical prewhitening or β-prewhitening with β=0 and their scattering plot, respectively. Figure 4.11(c-d) shows the recovered signals under β-prewhitening with β=0.05 and their scattering plot, respectively. Comparing figures 4.11(a-b) and 4.11(c-d) with 4.9(i), Evidently we see that recovered signals by FastICA under classical prewhitening are not good, while the recovered signals under β-prewhitening with β=0.05 are good and almost similar to the original signals.

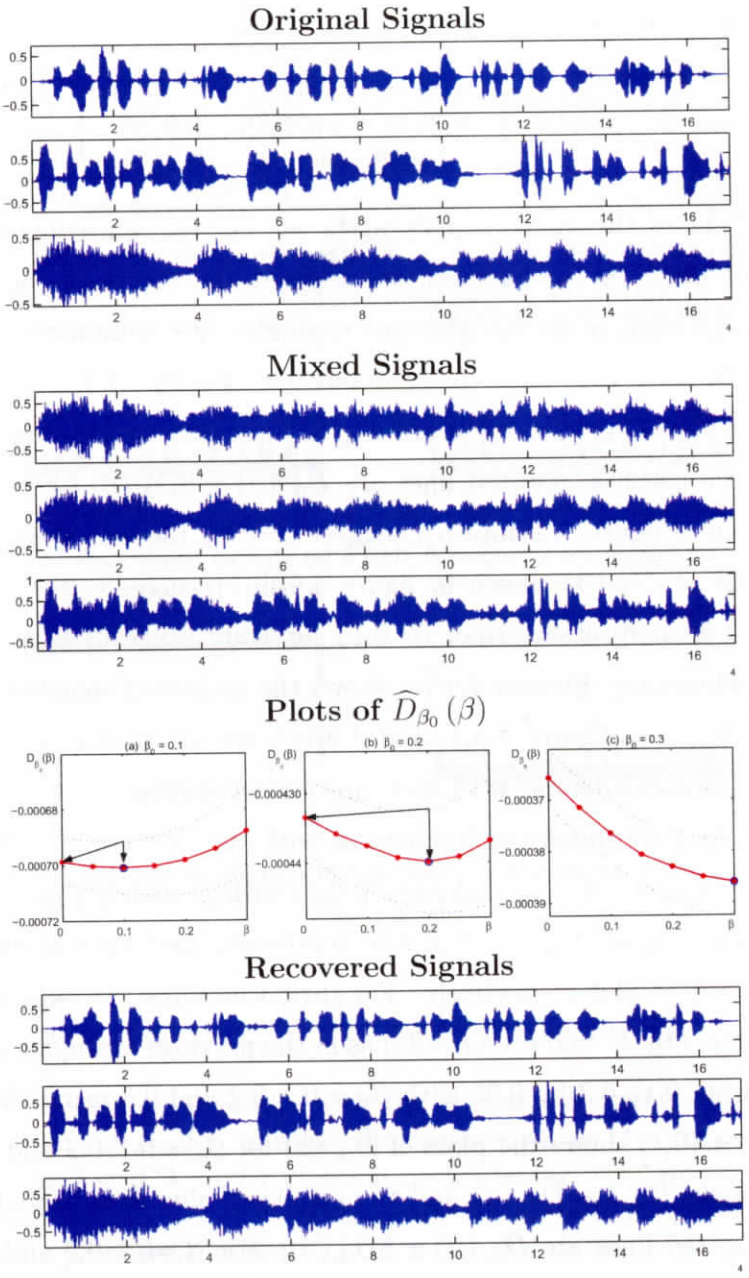

Figure 4.9: Plots for simulation results with the mixture of real audio signals. (i) Original signals (ii) mixed signals (iii) Plots of $\widehat{D}_{\beta_0}(\beta)$ to select appropriate values of β for β-prewhitening. (iv) Recovered signals under β-prewhitening with $\beta = 0$.

4.6 Conclusions

In this chapter, we proposed β-prewhitening as an adaptive robust pre-whitening procedure for ICA instead of existing prewhitening procedure. The performance of this new prewhiten-

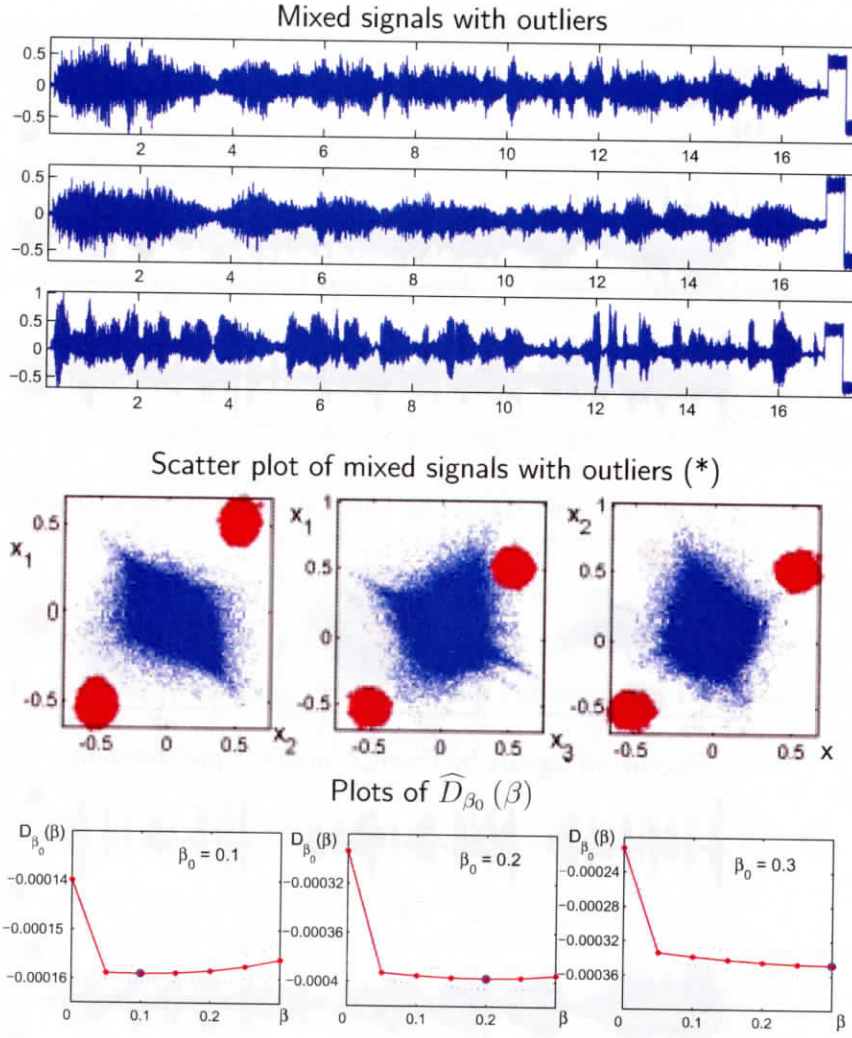

Figure 4.10: Plots for simulation results with the mixture of real audio signals. (a) Mixed signals with outliers. (b) Scatter plot of mixed signals with outliers (*). (c) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.1$, 0.2 and 0.3 to select appropriate β for β-prewhitening.

ing procedure is equivalent to the standard prewhitening if data set is not corrupted by noise or outliers. If data set is corrupted by noise or outliers, then β-prewhitening is much better than standard prewhitening.

The tuning parameter β plays the key role on the performance of β-prewhitening. Therefore, we proposed an adaptive selection procedure for the tuning parameter β based on cross validation. This adaptive selection procedure suggests β-prewhitening with β=0 if data set is not corrupted by noise or outliers. If data set is corrupted by noise or outliers, then adap-

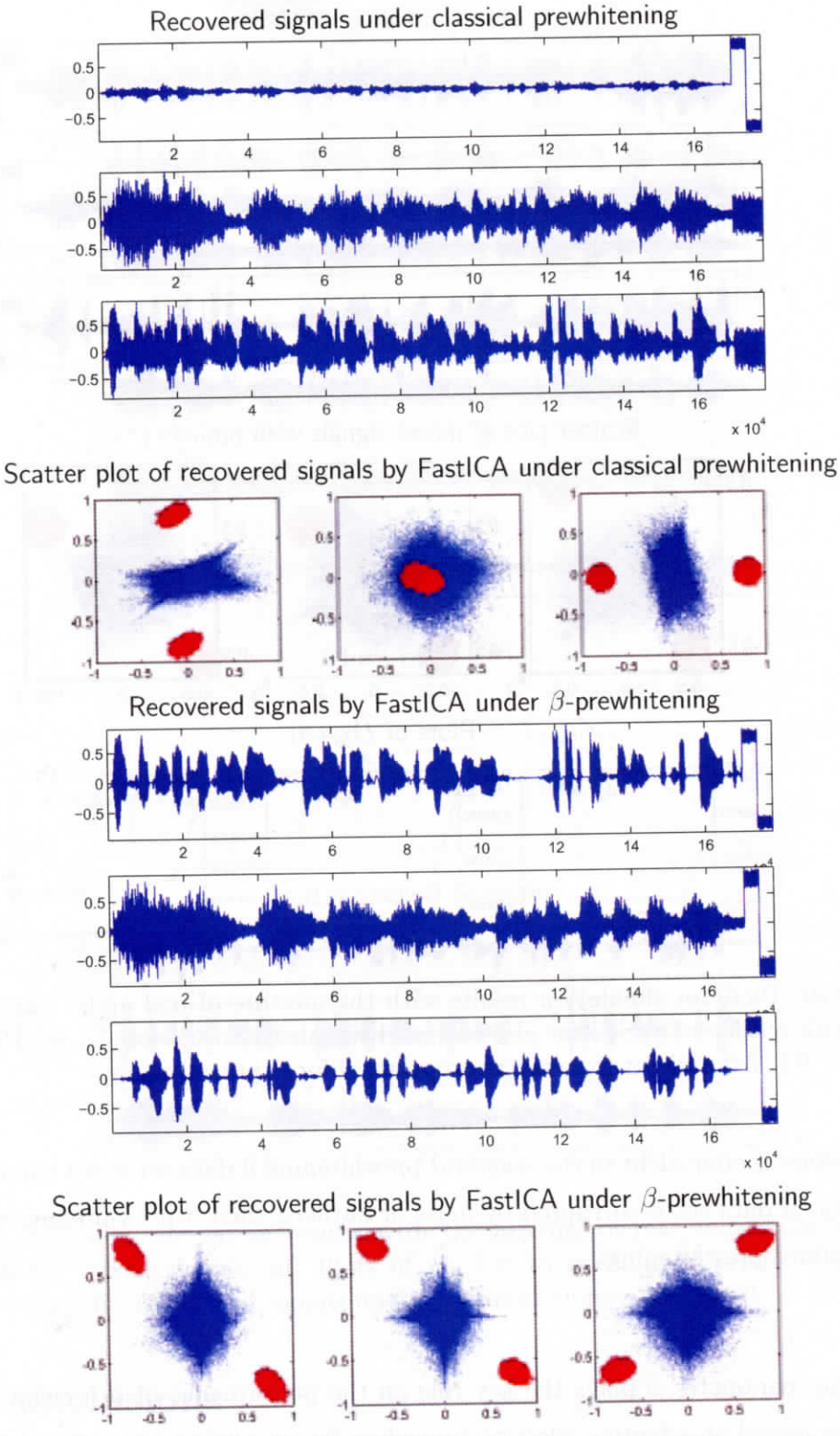

Figure 4.11: Plots for simulation results with the mixture of real audio signals. (a) Recovered signals by FastICA under classical prewhitening. (b) Scatter plot of recovered signals under classical prewhitening (c) Recovered signals by FastICA under β-prewhitening with β=0.05. (d) Scatter plot of recovered signals under β-prewhitening

tive selection procedure suggest β-prewhitening with $\beta > 0$. Note that β-prewhitening with $\beta = 0$ is equivalent to the standard prewhitening.

It is well known that standard prewhitening procedure is better than any other prewhitening procedures if data set is not corrupted by noise or outliers. On the other hand, if data set is corrupted by noise or outliers, then any robust prewhitening procedure is better than standard prewhitening procedure. However, it is very difficult to know in advance whether a data set is corrupted or not by outliers. Therefore, a researcher or user may feel inconvenience to select an appropriate algorithm for prewhitening. In this situation, β-prewhitening is suitable than any other prewhitening procedures.

At last, we proposed a measure of performance index for prewhitening procedures based on the eigenvalues of the global mixing matrix. In the simulation study, we investigated the performance of β-prewhitening procedure in a comparison of the standard prewhitening procedure using the proposed measure of performance index.

Chapter 5

Exploring Local PCA Structures by the Minimum β-Divergence Method

5.1 The Problem of PCA Mixture Models for Exploring Local Structures

Principal component analysis (PCA) is one of the most popular techniques for processing, compressing and visualizing multivariate data. It is widely used for dimensionality reduction of multivariate data (Jolliffe, 2002). In general, PCA aims to extract the most informative q-dimensional output vector $\boldsymbol{y}(t)$ from an input vector $\boldsymbol{x}(t) = \big(x_1(t), x_2(t), ..., x_m(t)\big)^T$ of dimension $m \geq q$ whose components are assumed to be Gaussian and linearly correlated of each other. This is achieved by learning the $m \times q$ orthogonal matrix Γ or $\Gamma^T\Gamma = I_q$ (q-identity matrix) which connects $\boldsymbol{x}(t)$ to $\boldsymbol{y}(t)$ by

$$\boldsymbol{y}(t) = \Gamma^T\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big), \quad (t = 1, 2, \ldots, n) \tag{5.1}$$

such that components of $\boldsymbol{y}_t = \big(y_1(t), y_2(t), ..., y_q(t)\big)^T$ are mutually uncorrelated satisfying $\mathrm{Var}(Y_1) > \mathrm{Var}(Y_2) > ... > \mathrm{Var}(Y_q) > 0$, where $\boldsymbol{\mu}$ is the mean vector of the input data. In neural networks, Γ is interpreted as the matrix of coefficients connecting m neurons to q neurons, where a learning process works by renewing Γ according to a batch of inputs in an off-line manner or sequential input vectors in an on-line manner (Oja, 1982, 1989; Haykin, 1999). The input vector $\boldsymbol{x}(t)$ is represented by the m-dimensional latent vector $\boldsymbol{s}(t) = \big(s_1(t), s_2(t), ..., s_m(t)\big)^T$ as

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t) + \boldsymbol{b}, \quad (t = 1, 2, \ldots, n) \tag{5.2}$$

where A is $m \times m$ non-singular coefficient matrix and $\boldsymbol{b}$ is bias vactor. The components of the latent vector $\boldsymbol{s}(t)$ are assumed to be mutually independent and Gaussian with unit

variance, that is, $\boldsymbol{s}(t) \sim N(0, I)$. The latent variable model (5.2) is considered as the data generating model. It offers more economical explanation of the linear dependencies among the input observations (Tipping and Bishop, 1997, 1999).

In classical PCA model defined by (5.1) and (5.2), all latent vectors belong to only one source class $\mathscr{S}$, and all input vectors belong to the same class in the entire data space $\mathscr{D}$. However, in practice, these source vectors may originate from several source classes, and the corresponding observed vectors belong to several classes in the entire data space. In this case, the performance of classical PCA may not be so good. Therefore, Tipping et al. (1999) proposed a PCA mixture models by modeling the observed data as a mixture of several mutually exclusive classes, each of which is described by linear combinations of independent and Gaussian densities. However, one problem encountered when applying this method is that the number of classes, c, should be known in advance, which is difficult in practice.

We assume that source vectors come from c source classes $\{\mathscr{S}_1, \mathscr{S}_2, ..., \mathscr{S}_c\}$ and that the corresponding observed vectors belong to c different data classes $\{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_c\}$ in the entire data space $\mathscr{D}$, where the number c is unknown. In addition, we assume that data class $\mathscr{D}_k$ occurs in the entire data space $\mathscr{D}$ due to the source class $\mathscr{S}_k$, $(k = 1, 2, \ldots, c)$. In practice, the occurrence order of an observed vector in the entire data space $\mathscr{D}$ from a source class is unknown. However, we can assume that an observed vector $\boldsymbol{z}_k(j) \in \mathscr{D}_k = \{\boldsymbol{z}_k(j);\ j = 1, 2, \ldots, n_k\}$, $(k = 1, 2, \ldots, c;\ \sum_{k=1}^{c} n_k = n)$ whose occurrence order is unobserved, follows a PCA data generating model as

$$\boldsymbol{z}_k(j) = A_k \boldsymbol{s}_k(j) + \boldsymbol{b}_k, \tag{5.3}$$

where A_k is an $m \times m$ non-singular coefficient matrix, $\boldsymbol{b}_k$ is the bias vector and $\boldsymbol{s}_k(j) \in \mathscr{S}_k = \{\boldsymbol{s}_k(j);\ j = 1, 2, \ldots, n_k\}$, $(k = 1, 2, \ldots, c)$ is the j-th random vector in the source class k with zero mean vector, the components of which are assumed to be independent and Gaussian. In a practical situation, an observable vector $\boldsymbol{x}_t \in \mathscr{D} = \{\boldsymbol{x}(t);\ t = 1, 2, \ldots, n\}$ is obtained as one vector of $\bigcup_{k=1}^{c} \mathscr{D}_k = \{\boldsymbol{z}_k(j);\ j = 1, 2, \ldots, n_k,\ k = 1, 2, \ldots, c;\ \sum_{k=1}^{c} n_k = n\}$ such that $\mathscr{D} = \bigcup_{k=1}^{c} \mathscr{D}_k$. If the permutation of $\{\boldsymbol{z}_1(1), \boldsymbol{z}_1(2), \ldots, \boldsymbol{z}_k(j), \ldots, \boldsymbol{z}_c(n_c)\}$ into $\{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(n)\}$ is purely random, then (5.3) reduces to the probabilistic PCA mixture models. In the probabilistic PCA mixture models, the observed data in each class are considered to be a linear combination of independent and Gaussian sources. (See Tipping

and Bishop (1997, 1999) for a detailed discussion). When the data in each class are modeled as multivariate non-Gaussian, it is known as a ICA mixture model (Mollah, Minami and Eguchi, 2006).

Based on the situation discussed above, our proposed method is sequential application of the minimum β-divergence method with explicitly including a local kernel function to extract all local structures sequentially for PCA based on a rule of step-by-step change of the shifting parameter. Later, we will propose a stopping rule for repeated application of the minimum β-divergence method based on the cumulative weight. In order to explore k-th local structure, we estimate an $m \times q_k$ orthogonal matrix Γ_k and a shifting parameter $\boldsymbol{\mu}_k$, based on the minimum β-divergence method with explicitly including a local kernel parameter vector $\boldsymbol{x} \in \mathscr{D}$, initializing both $\boldsymbol{x}$ and $\boldsymbol{\mu}_k$ by the same vector $\boldsymbol{x}_0 \in \mathscr{D}_k$ that transforms the input vector $\boldsymbol{x}(t) \in \mathscr{D}$ into an output vector $\boldsymbol{y}(t)$ by

$$\boldsymbol{y}(t) = \Gamma_k^T\big(\boldsymbol{x}(t) - \boldsymbol{\mu}_k\big), \quad (t = 1, 2, \ldots, n) \tag{5.4}$$

such that

$$\begin{aligned} \boldsymbol{y}(t) \quad &\in \quad \mathscr{D}_k^o = \big\{\boldsymbol{y}_k(j);\ j = 1, 2, \ldots, n_k\big\}, \qquad \text{if} \quad \boldsymbol{x}_t \in \mathscr{D}_k \\ &\in \quad \mathscr{D}^*, \qquad \text{otherwise,} \end{aligned}$$

where components of $\boldsymbol{y}_k = (y_{1k}, y_{2k}, ..., y_{q_k k})^T$ are mutually uncorrelated satisfying

$$\mathrm{Var}(Y_{1k}) > \mathrm{Var}(Y_{2k}) > ... > \mathrm{Var}(Y_{q_k k}) > 0. \tag{5.5}$$

Here $\mathscr{D}_k^o$ is the estimated orthogonal class for data class $\mathscr{D}_k$ and $\mathscr{D}^*$ is the set of output vectors corresponding to the other data classes. The values of the tuning parameter β and kernel parameter ν plays a key rule on the performance of the proposed method. Therefore, an adaptive selection procedure is proposed for both β and ν.

Section (5.2) reviews the existing methods for PCA, section (5.3) discusses the local PCA based on Gaussian mixture distribution, sections (5.4 - 5.4.3) describe the proposed minimum β-divergence method for local PCA. Finally, section (5.5) presents numerical examples, and Section (5.6) presents the conclusions of this study.

5.2 A Review on Existing PCA Methods

Let us present a concise review of the classical PCA for detecting the principal q-subspace. Let

$$z\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big) = \frac{1}{2}\left\{\|\boldsymbol{x}(t) - \boldsymbol{\mu}\|^2 - \big\|\Gamma^T\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big)\big\|^2\right\} \tag{5.6}$$

be half the square of the residual distance of $(\boldsymbol{x} - \boldsymbol{\mu})$ from the subspace spanned by the columns of Γ. We note that $z(\boldsymbol{x}, \boldsymbol{\mu}, \Gamma) = \frac{1}{2}\min_{\lambda \in R^k} \|\boldsymbol{x}(t) - \boldsymbol{\mu} - \Gamma\lambda\|^2$, see Hotelling (1933) for the original derivation. Classical PCA is simply characterized by minimizing

$$\frac{1}{n}\sum_{t=1}^{n} z\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big) \tag{5.7}$$

with respect to $\boldsymbol{\mu}$ and Γ, which reduces to solving q dominant eigenvectors of the sample covariance matrix

$$S = \frac{1}{n}\sum_{t=1}^{n}\big(\boldsymbol{x}(t) - \widehat{\boldsymbol{\mu}}\big)\big(\boldsymbol{x}(t) - \widehat{\boldsymbol{\mu}}\big)^T, \tag{5.8}$$

where the centralized vector $\widehat{\boldsymbol{\mu}} = \sum_{t=1}^{n} \boldsymbol{x}(t)/n$. Then, we obtain a solution Γ by stacking the q dominant eigenvectors of S, which we write in the form

$$\Gamma = \mathrm{eigen}(S) \tag{5.9}$$

Higuchi and Eguchi (2004) proposed a variant of this classical procedure for robust PCA by minimizing the objective function

$$L(\boldsymbol{\mu}, \Gamma) = \frac{1}{n}\sum_{t=1}^{n} \Psi\big(z\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\big) \tag{5.10}$$

where $\Psi(z)$ is assumed to be a monotonically increasing function of $z > 0$. Various Ψs yield various procedures for PCA. As typical examples, the identity function $\Psi_0(z) = z$ reduces to the classical PCA and

$$\Psi_1(z) = \log \frac{1}{1 + \exp\{-\lambda(z - \eta)\}} \tag{5.11}$$

defines Xu and Yuille's self-organizing rule, where λ and η are tuning parameters, referred to as the inverse temperature and saturation value, respectively (Xu and Yuille, 1995). In general, Ψ is interpreted as the generic function which gives the total function L. The minimization of L in equation 5.10 is referred as the " minimum psi principle generated by

Ψ". Based on an argument similar to that of the classical PCA, Higuchi and Eguchi (2004) found that the minimizer $(\tilde{\boldsymbol{\mu}}, \tilde{\Gamma})$ of $L(\boldsymbol{\mu}, \Gamma)$ satisfies the stationary equations

$$\tilde{\boldsymbol{\mu}} = \sum_{t=1}^{n} p_t(\tilde{\boldsymbol{\mu}}, \tilde{\Gamma}) \boldsymbol{x}(t), \tag{5.12}$$

and

$$\tilde{\Gamma} = \text{eigen}\left(S(\tilde{\boldsymbol{\mu}}, \tilde{\Gamma}\right) \tag{5.13}$$

where

$$p_t(\boldsymbol{\mu}, \Gamma) = \frac{\psi\Big(z\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\Big)}{\sum_{t=1}^{n} \psi\Big(z\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\Big)} \tag{5.14}$$

and

$$S\left(\boldsymbol{\mu}, \Gamma\right) = \sum_{t=1}^{n} p_t\left(\boldsymbol{\mu}, \Gamma\right) \big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big)\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big)^T \tag{5.15}$$

with $\psi(z) = (\partial/\partial z)\Psi(z)$. The equilibrium point $(\tilde{\boldsymbol{\mu}}, \tilde{\Gamma})$ is expressed by the weighted mean and the covariance matrix, where the weight function p_t depends upon $\tilde{\boldsymbol{\mu}}$ and $\tilde{\Gamma}$, except for the case of $\psi(z) = 1$, which yields the classical PCA.

In the statistical literature (Croux et al., 2000; Campbell, 1980; Caussinus, 1990), another type of PCA method has been proposed in which

$$\frac{1}{n} \sum_{t=1}^{n} \Psi\Big(d\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\Big)$$

is minimized with respect to $(\boldsymbol{\mu}, V)$, where d is the Mahalanobis squared distance, that is

$$d\big(\boldsymbol{x}(t), \boldsymbol{\mu}, V\big) = \frac{1}{2}\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big) V^{-1} \big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big).$$

5.3 Local PCA Based on Gaussian Mixture (GM) Distribution

The Gaussian mixture distribution for a random vector $\boldsymbol{x}_t$ is given by

$$p(\boldsymbol{x}_t \mid \Theta) = \sum_{k=1}^{r} p(C_k) \varphi(\boldsymbol{x}_t \mid \theta_k, C_k), \tag{5.16}$$

where C_k denotes the k-th Gaussian class and $\theta_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$ are the unknown parameters for the density

$$\varphi(\boldsymbol{x} \mid \theta_k, C_k) = |\det(2\pi\Sigma_k)|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right\}, \tag{5.17}$$

corresponding to the k-th Gaussian class. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ be a random sample drawn from (5.17), then the log-likelihood of the data for the unknown parameter $\Theta = \{\theta_1, \theta_2, ...\theta_c\}$ is given by

$$L = \sum_{t=1}^{n} \log p(\boldsymbol{x}_t \mid \Theta). \tag{5.18}$$

The gradients of the parameters for class k is given by

$$\frac{\partial L}{\partial \theta_k} = \sum_{t=1}^{n} \frac{1}{p(\boldsymbol{x}_t \mid \Theta)} \frac{\partial}{\partial \theta_k} p(\boldsymbol{x}_t \mid \Theta) \tag{5.19}$$

$$= \sum_{t=1}^{n} \frac{\frac{\partial}{\partial \theta_k} p(C_k)\varphi(\boldsymbol{x}_t \mid \theta_k, C_k)}{p(\boldsymbol{x}_t \mid \Theta)} \tag{5.20}$$

Using the Bayes relation, the class probability for a given data vector $\boldsymbol{x}_t$ is

$$p(C_k \mid \boldsymbol{x}_t, \Theta) = \frac{p(C_k)\varphi(\boldsymbol{x}_t \mid \theta_k, C_k)}{\sum_{k=1}^{c} p(C_k)\varphi(\boldsymbol{x}_t \mid \theta_k, C_k)} \tag{5.21}$$

Substituting $(A.21)$ in $(A.20)$ leads to

$$\frac{\partial L}{\partial \theta_k} = \sum_{t=1}^{n} \frac{p(C_k \mid \boldsymbol{x}_t, \Theta)}{p(C_k)\varphi(\boldsymbol{x}_t \mid \theta_k, C_k)} \frac{\partial}{\partial \theta_k} p(C_k)\varphi(\boldsymbol{x}_t \mid \theta_k, C_k) \tag{5.22}$$

$$= \sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta) \frac{\partial}{\partial \theta_k} \log \varphi(\boldsymbol{x}_t \mid \theta_k, C_k) \tag{5.23}$$

Now,

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \log \varphi(\boldsymbol{x}_t \mid \theta_k, C_k) = \Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)$$

and

$$\frac{\partial}{\partial \Sigma_k} \log \varphi(\boldsymbol{x}_t \mid \theta_k, C_k) = \frac{1}{2}\Sigma_k^{-1}\{(\boldsymbol{x}_t - \boldsymbol{\mu}_k)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^T - \Sigma_k\}\Sigma_k^{-1}$$

Therefore, $\frac{\partial L}{\partial \boldsymbol{\mu}_k} = 0$ implies

$$\boldsymbol{\mu}_k{}^* = \frac{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)\boldsymbol{x}_t}{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)} \tag{5.24}$$

and $\frac{\partial L}{\partial \Sigma_k} = 0$ implies

$$\Sigma_k{}^* = \frac{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^T}{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)} \tag{5.25}$$

Note that prior probability $p(C_k)$ can be updated by

$$p(C_k)^* = \frac{1}{n}\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta) \tag{5.26}$$

The notations ${\mu_k}^*$, ${\Sigma_k}^*$ and $p(C_k)^*$ are the update of μ_k, Σ_k and $p(C_k)$ respectively, where Σ_k should be initialized by identity matrix and other parameters can be initialized randomly. Then, the orthogonal matrix for extracting k-th local PCA structure is obtained as

$$\widehat{\Gamma}_k = \text{eigen}\left(\widehat{\Sigma}_k\right) \tag{5.27}$$

If c=1, then local PCA based on GM distribution reduces to the standard PCA as discussed around equations 5.8 and 5.9. For local PCA, we transform the input data vector $\boldsymbol{x}_t$ into output vector $\boldsymbol{y}_t$ by

$$\boldsymbol{y}_t = \widehat{\Gamma}_{(k)}^T\left(\boldsymbol{x}_t - \widehat{\boldsymbol{\mu}}_{(k)}\right), \quad t = 1, 2, ..., n; \; k = 1, 2, ..., c \tag{5.28}$$

where

$$\widehat{\Gamma}_{(k)} \in \{\widehat{\Gamma}_1, \widehat{\Gamma}_2, ..., \widehat{\Gamma}_c\} \quad \text{and} \quad \widehat{\boldsymbol{\mu}}_{(k)} \in \{\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_2, ..., \widehat{\boldsymbol{\mu}}_c\}, \quad k = 1, 2, ..., c$$

Then, (k)-th local PCA structure is defined by those output vectors $\boldsymbol{y}_t$ whose input vectors $\boldsymbol{x}_t$ belong to the data class

$$\mathscr{D}_{(k)} = \left\{\boldsymbol{x}_t \in \mathscr{D} : \; p(C_{(k)} \mid \boldsymbol{x}_t, \Theta) \geq 0.5\right\}, \tag{5.29}$$

5.4 New Estimator for PCA by Minimizing β-Divergence

The β-divergence between two pdf's $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ is defined as

$$D_\beta(p, q) = \int \left[\frac{1}{\beta}\left\{p^\beta(\boldsymbol{x}) - q^\beta(\boldsymbol{x})\right\}p(\boldsymbol{x}) - \frac{1}{\beta+1}\left\{p^{\beta+1}(\boldsymbol{x}) - q^{\beta+1}(\boldsymbol{x})\right\}\right] d\boldsymbol{x}, \quad \text{for } \beta > 0 \tag{5.30}$$

which is non-negative, that is $D_\beta(p(\boldsymbol{x}), q(\boldsymbol{x})) \geq 0$, equality holds iff $\mathrm{p}(\boldsymbol{x}) = \mathrm{q}(\boldsymbol{x})$, (cf. Minami et al. (2002)). We note that β-divergence reduces to Kullback Leibler (KL) divergence when $\beta \longrightarrow 0$, that is

$$\lim_{\beta \downarrow 0} D_\beta(p, q) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\, d\boldsymbol{x} = D_{\mathrm{KL}}(p, q). \tag{5.31}$$

Let $p(\boldsymbol{x})$ is the empirical distribution (data distribution) of $\boldsymbol{x}$ and $\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})$ is the Gaussian density $N(\boldsymbol{\mu}, V)$. Then the minimum β-divergence estimators for $\boldsymbol{\mu}$ and V are obtained by

$$\underset{\boldsymbol{\mu},V}{\operatorname{argmin}}\ D_\beta\Big(p(\boldsymbol{x}),\ \varphi_{\boldsymbol{\mu},V}(\boldsymbol{x})\Big) \ = \ \underset{\boldsymbol{\mu},V}{\operatorname{argmin}}\ L_\beta\left(\boldsymbol{\mu}, V\right), \tag{5.32}$$

where,

$$L_\beta\left(\boldsymbol{\mu}, V\right) = \begin{cases} \dfrac{1}{\beta+1}\displaystyle\int \left\{\varphi^{\beta+1}_{\boldsymbol{\mu},V}(\boldsymbol{x})\right\} d\boldsymbol{x} - \dfrac{1}{\beta}\int \left\{p(\boldsymbol{x})\varphi^{\beta}_{\boldsymbol{\mu},V}(\boldsymbol{x})\right\} d\boldsymbol{x}, & \text{for} \quad \beta > 0 \\ -\int p(\boldsymbol{x}) \log \varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}) d\boldsymbol{x}, & \text{for} \quad \beta = 0 \end{cases} \tag{5.33}$$

or

$$L_\beta\left(\boldsymbol{\mu}, V\right) = \begin{cases} c_1 - \dfrac{1}{\beta}\cdot\dfrac{1}{n}\displaystyle\sum_{t=1}^{n}\left\{\varphi^{\beta}_{\boldsymbol{\mu},V}(\boldsymbol{x}(t))\right\}, & \text{for} \quad \beta > 0 \\ -\dfrac{1}{n}\displaystyle\sum_{t=1}^{n}\log \varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}(t)), & \text{for} \quad \beta = 0 \end{cases} \tag{5.34}$$

or

$$L_\beta\left(\boldsymbol{\mu}, V\right) = \begin{cases} c_1 - \dfrac{c_2}{n\beta}\displaystyle\sum_{t=1}^{n}\exp\left\{-\frac{\beta}{2}\Big(\boldsymbol{x}(t)-\boldsymbol{\mu}\Big)^T V^{-}\Big(\boldsymbol{x}(t)-\boldsymbol{\mu}\Big)\right\}, & \text{for} \quad \beta > 0 \\ c_3 + \dfrac{1}{n}\displaystyle\sum_{t=1}^{n}\left\{\frac{1}{2}\Big(\boldsymbol{x}(t)-\boldsymbol{\mu}\Big)^T V^{-}\Big(\boldsymbol{x}(t)-\boldsymbol{\mu}\Big)\right\}, & \text{for} \quad \beta = 0 \end{cases} \tag{5.35}$$

By the neural network property for PCA, there is an orthogonal matrix Γ satisfying

$$V^{-} = (I_m - \Gamma\Gamma^T) \approx \begin{pmatrix} I_q & O \\ O & O \end{pmatrix}$$

where I_q is the identity matrix of order $q \times q$ and O's are the zero matrices of appropriate order. Here c_1, c_2 and c_3 are constant by the above relation. Then (5.35) reduces to

$$L_\beta\left(\boldsymbol{\mu}, \Gamma\right) = \begin{cases} c_1 - \dfrac{c_2}{n\beta}\displaystyle\sum_{t=1}^{n}\exp\left\{-\frac{\beta}{2}\left(\|\boldsymbol{x}(t)-\boldsymbol{\mu}\|^2 - \left\|\Gamma^T(\boldsymbol{x}(t)-\boldsymbol{\mu})\right\|^2\right)\right\}, & \text{for} \quad \beta > 0 \\ c_3 + \dfrac{1}{n}\displaystyle\sum_{t=1}^{n}\left\{\frac{1}{2}\left(\|\boldsymbol{x}(t)-\boldsymbol{\mu}\|^2 - \left\|\Gamma^T(\boldsymbol{x}(t)-\boldsymbol{\mu})\right\|^2\right)\right\}, & \text{for} \quad \beta = 0 \end{cases} \tag{5.36}$$

Minimization of (5.36) with respect to $\boldsymbol{\mu}$ and Γ is equivalent to the minimization of

$$L_\beta\left(\boldsymbol{\mu}, \Gamma\right) = \begin{cases} \dfrac{1}{\beta}\left[1 - \dfrac{1}{n}\displaystyle\sum_{t=1}^{n}\exp\left\{-\beta z\Big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\Big)\right\}\right], & \text{for} \quad \beta > 0 \\ \dfrac{1}{n}\displaystyle\sum_{t=1}^{n} z\Big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\Big), & \text{for} \quad \beta = 0 \end{cases} \tag{5.37}$$

with respect to $\boldsymbol{\mu}$ and Γ. For convenience of presentation of our new proposal based on equation 5.37, let us denote the generic function Ψ in equation 5.10 as

$$\Psi_\beta(z) = \begin{cases} \frac{1}{\beta}\left\{1 - \exp(-\beta z)\right\}, & \text{for} \quad \beta > 0 \\ z, & \text{for} \quad \beta = 0 \end{cases} \tag{5.38}$$

Note that minimization of $L_\beta(\boldsymbol{\mu}, \Gamma)$ for $\beta = 0$ with respect to $\boldsymbol{\mu}$ and Γ offers classical PCA. In equation (5.38), $\Psi_\beta(z)$ is the monotonic increasing function of $z > 0$. Therefore, based on an argument similar to that of the robust PCA (Higuchi and Eguchi, 2004), we obtain the minimizer of $L_\beta(\boldsymbol{\mu}, \Gamma)$ for $\beta > 0$ by

$$\boldsymbol{\mu}^* = \frac{\sum_{t=1}^n \psi_\beta\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\boldsymbol{x}(t)}{\sum_{t=1}^n \psi_\beta\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)}, \tag{5.39}$$

$$\Gamma^* = \text{eigen}(V^*) \tag{5.40}$$

where

$$V^* = \frac{\sum_{t=1}^n \psi_\beta\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big)\big(\boldsymbol{x}(t) - \boldsymbol{\mu}\big)^T}{\sum_{t=1}^n \psi_\beta\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big)}, \tag{5.41}$$

and

$$\psi_\beta\big(\boldsymbol{x}(t), \boldsymbol{\mu}, \Gamma\big) = \exp\left\{-\frac{\beta}{2}\left(\|\boldsymbol{x}(t) - \boldsymbol{\mu}\|^2 - \left\|\Gamma^T(\boldsymbol{x}(t) - \boldsymbol{\mu})\right\|^2\right)\right\} \tag{5.42}$$

with $\psi_\beta(z) = (\partial/\partial z)\Psi_\beta(z)$. The notations $\boldsymbol{\mu}^*$, V^* and Γ^* represent the update of $\boldsymbol{\mu}$, V and Γ, respectively. Here $\psi_\beta(\boldsymbol{x}(t);\ \boldsymbol{\mu}, \Gamma)$ is considered to be a weight function. It provides a weight to each data point for robust PCA. For $\beta \to 0$, (5.39), (5.40) and (5.41) reduce to the classical non-iterative estimates as discussed around equations 5.8 and 5.9.

5.4.1 Exploring Local PCA Structures by the Minimum β-Divergence Method Using Gaussian Kernel Function

For local PCA, we modify the objective function (equation 5.37) for $\beta > 0$ without loss of generality in the minimization of the objective function. We post-multiply the exponential part by a local kernel function

$$\exp\left\{-\frac{\nu}{2}\|\boldsymbol{x}(t) - \boldsymbol{x}\|^2\right\}$$

to impose more weight to the data points that belong to the local cluster, where $\boldsymbol{x}$ is the center of the kernel and ν is the inverse of the bandwidth. Then the objective function (equation 5.37) is extended as

$$L_{\beta,\nu}(\boldsymbol{\mu},\Gamma) = \begin{cases} \frac{1}{\beta}\left[1 - \frac{1}{n}\sum_{t=1}^{n}\exp\left(-\beta\left\{z\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma\big) + \frac{\nu}{2}\left\|\boldsymbol{x}(t)-\boldsymbol{x}\right\|^2\right\}\right)\right], & \text{for} \quad \beta > 0 \\ \frac{1}{n}\sum_{t=1}^{n}\left\{z\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma\big) + \frac{\nu}{2}\left\|\boldsymbol{x}(t)-\boldsymbol{x}\right\|^2\right\}, & \text{for} \quad \beta = 0. \end{cases} \tag{5.43}$$

Note that minimization of $L_{\beta,\nu}(\boldsymbol{\mu},\Gamma)$ for $(\beta,\ \nu) = (0,0)$ with respect to μ and Γ offers classical PCA. We can employ the reweighting learning algorithm to obtain the minimizer of $L_{\beta,\nu}(\boldsymbol{\mu},\Gamma)$ for $\beta > 0$ by

$$\boldsymbol{\mu}^* = \frac{\sum_{t=1}^{n}\psi_{\beta,\nu}\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma,\boldsymbol{x}\big)\boldsymbol{x}(t)}{\sum_{t=1}^{n}\psi_{\beta,\nu}\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma,\boldsymbol{x}\big)} \tag{5.44}$$

$$\Gamma^* = \text{eigen}\,(V^*) \tag{5.45}$$

where

$$V^* = \frac{\sum_{t=1}^{n}\psi_{\beta,\nu}\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma,\boldsymbol{x}\big)\big(\boldsymbol{x}(t)-\boldsymbol{\mu}\big)\big(\boldsymbol{x}(t)-\boldsymbol{\mu}\big)^T}{\sum_{t=1}^{n}\psi\big(\boldsymbol{x}(t),\boldsymbol{\mu},\Gamma,\boldsymbol{x}\big)}, \tag{5.46}$$

$$\psi_{\beta,\nu}\big(\boldsymbol{x}(t);\ \boldsymbol{\mu},\Gamma,\boldsymbol{x}\big) = \exp\left\{-\frac{\beta}{2}\left(\left\|\boldsymbol{x}(t)-\boldsymbol{\mu}\right\|^2 - \left\|\Gamma^T\left(\boldsymbol{x}(t)-\boldsymbol{\mu}\right)\right\|^2 + \nu\left\|\boldsymbol{x}(t)-\boldsymbol{x}\right\|^2\right)\right\} \tag{5.47}$$

with $\psi_{\beta,\nu}(z) = (\partial/\partial z)\Psi_{\beta,\nu}(z)$. The notations $\boldsymbol{\mu}^*$, V^* and Γ^* represent the update of $\boldsymbol{\mu}$, V and Γ, respectively. Here $\psi_{\beta,\nu}(\boldsymbol{x}(t);\ \boldsymbol{\mu},\Gamma,\boldsymbol{x})$ is a weight function, which significantly weights each data point that belongs to the local cluster and insignificantly weights data points otherwise. For $(\beta,\nu) \to (0,0)$, equations 5.44, 5.45 and 5.46 reduce to the classical non-iterative estimates as discussed around equations 5.8 and 5.9.

5.4.2 A Sequential Procedure to Explore Local PCA Structures

Tipping and Bishop (1999) proposed mixtures of PPCA algorithm for extracting all local PCA structures simultaneously by maximizing the likelihood function using EM algorithm. In this section, we are proposing a new iterative algorithm based on the minimum β-divergence method using a local kernel function for extracting all local PCA structures sequentially. The proposed method explores an orthogonal matrix to extract a local PCA

structure based on the initial condition of the shifting parameter $\boldsymbol{\mu}$ and the local kernel vector $\boldsymbol{x}$. If the initial values of $\boldsymbol{\mu}$ and $\boldsymbol{x}$ both belong to the data class $\mathscr{D}_k$, then the estimates of orthogonal matrix Γ_k and shifting parameter $\boldsymbol{\mu}_k$ can suggest for k-th local PCA structure by considering the data in other classes as outliers. Thus, we can learn $\{(\Gamma_k, \boldsymbol{\mu}_k);\ k = 1, 2, ..., c\}$ by the repeated application of the minimum β-divergence method to extract all local PCA structures sequentially based on a rule for the step-by-step change of the initial values for $\boldsymbol{\mu}$ and $\boldsymbol{x}$ both by the same vector. Note that initial kernel vector $\boldsymbol{x}$ will be fixed for each iteration in a step. Our proposed learning algorithm for sequential estimation of Γ_k and $\boldsymbol{\mu}_k$, (k=1, 2, ..., c) is given below:

Step 1: Randomly fix $\boldsymbol{x}_{(1)} \in \mathscr{D}$ and let $\boldsymbol{x} = \boldsymbol{x}_{(1)}$. Find the minimizer $\left(\widehat{\Gamma}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ of the loss function $L_{\beta,\nu}(\Gamma, \boldsymbol{\mu})$ applying the reweighted algorithm defined by equations 5.44 and 5.45. The initial setting in the algorithm is that $\boldsymbol{\mu} = \boldsymbol{x}_{(1)}$ and V is the identity matrix.
Let us suppose that $(k-1)$ pairs of estimates

$$\left\{\left(\widehat{\Gamma}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right), \left(\widehat{\Gamma}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right), ..., \left(\widehat{\Gamma}_{(k-1)}, \widehat{\boldsymbol{\mu}}_{(k-1)}\right)\right\}.$$

are obtained sequentially in steps 1 to $(k-1)$.

Step k: Learn $\boldsymbol{\mu}$, Γ and V using equations 5.44, 5.45 and 5.46, iteratively, changing initial value for $\boldsymbol{\mu}$ and the local kernel vector $\boldsymbol{x}$ both by $\boldsymbol{x}_{(k)} \in \mathscr{D}$, which is obtained such that $\phi_{k-1}\left(\boldsymbol{x}_{(k)}\right)$ is the nearest value of α-th percentile of cumulative weights

$$\phi_{k-1}\big(\boldsymbol{x}(t)\big) = \sum_{j=1}^{k-1} \psi_{\beta,\nu}\left(\boldsymbol{x}(t);\ \widehat{\boldsymbol{\mu}}_{(j)}, \widehat{\Gamma}_{(j)}, \boldsymbol{x}_{(j)}\right) \tag{5.48}$$

for $t = 1, 2, ..., n$. Then, let $\widehat{\boldsymbol{\mu}} = \widehat{\boldsymbol{\mu}}_{(k)}$, $\widehat{V} = \widehat{V}_{(k)}$ and $\widehat{\Gamma} = \widehat{\Gamma}_{(k)}$.
If $\boldsymbol{\mu}$ and $\boldsymbol{x}$ are initialized by an outlier data point, then the proposed method might provide misleading results. Therefore, a fixed integer α with $(2 \leq \alpha \leq 5)$ should be used to obtain reasonable result. Latter we will discuss an adaptive selection for β and ν for each step $k = 1, 2, ..., c$. Accordingly, our desired estimates are

$$\left\{\left(\widehat{\Gamma}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right), \left(\widehat{\Gamma}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right), ..., \left(\widehat{\Gamma}_{(c)}, \widehat{\boldsymbol{\mu}}_{(c)}\right)\right\}.$$

For local PCA, we transform the input data vector $\boldsymbol{x}(t)$ into output vector $\boldsymbol{y}(t)$ by

$$\boldsymbol{y}(t) = \widehat{\Gamma}_{(k)}^{T}\big(\boldsymbol{x}(t) - \widehat{\boldsymbol{\mu}}_k\big), \quad (t = 1, 2, ..., n;\ k = 1, 2, ..., c) \tag{5.49}$$

Then, k-th local PCA structure is defined by those output vectors $\boldsymbol{y}(t)$ whose input vectors $\boldsymbol{x}(t)$ belong to the data class

$$\mathscr{D}_{(k)} = \left\{ \boldsymbol{x}(t) \in \mathscr{D} : \ \psi_{\beta,\nu}\left(\boldsymbol{x}(t); \widehat{\Gamma}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}, \boldsymbol{x}_{(k)}\right) \geq \epsilon_k \right\}, \tag{5.50}$$

where we chose the value of ϵ_k by

$$\epsilon_k = (1-\eta) \min_{\boldsymbol{x}(t)\in\mathscr{D}} \psi_{\beta,\nu}\left(\boldsymbol{x}(t); \widehat{\Gamma}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}, \boldsymbol{x}_{(k)}\right) + \eta \max_{\boldsymbol{x}_t\in\mathscr{D}} \psi_{\beta,\nu}\left(\boldsymbol{x}(t); \widehat{\Gamma}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}, \boldsymbol{x}_{(k)}\right),$$

with heuristically $0.01 \leq \eta \leq 0.05$. Components of each output vector that corresponds to the k-th local PCA structure are mutually uncorrelated satisfying (equation 5.5). Note that weight of each input vector that does not correspond to the k-th local PCA structure is almost close to zero.

The cumulative weighting plot represents the weight of each data point whether it corresponds to any one local PCA structure or not. Thus, sequential estimation can be continued until the remaining data points correspond to a local PCA structure by monitoring the cumulative weighting plot and the value of the termination index, TI $= \frac{|J|}{n} \leq 1$, after each step, where $|J|$ is the number of elements in the set

$$J = \left\{ t : \sum_{k=1}^{c} \psi_{\beta,\nu}\left(\boldsymbol{x}(t); \widehat{\Gamma}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}, \boldsymbol{x}_{(k)}\right) \geq \epsilon \right\} \tag{5.51}$$

where $\epsilon = \sum_{k=1}^{c} \epsilon_k$.

If $\boldsymbol{\mu}$ and $\boldsymbol{x}$ are initialized at step 1 by the outlier data vector, then the proposed method might provide misleading results and TI might be very small. Therefore, if TI ≤ 0.1, we should restart the procedure by changing the initial value using other data vector at step 1. If initialization occurs from the overlapping section of two or more data clusters at step 1, then the proposed method might be given misleading results also and TI might be very large. Therefore, if TI ≥ 0.8, we should restart the proposed procedure by changing the initial value using other data vector at step 1. If there is only one data cluster in the entire data space, then TI might be greater than 0.90 for any initialization. The value TI $= a \leq 1$ suggests around 100a% input data vectors are transformed to the output data vectors corresponding to several local PCA structures and the rest of the data points, 100(1-a)%, remain untransformed as outlier or overlapping data points. The transformation procedure is terminated when the value of the termination index TI exceeds a certain value. In our simulation

study, we terminated the procedure when TI exceeds 0.90. It should be noted here that the performance of the proposed method depends on the value of the tuning parameters β and ν, where β controls weight for robust PCA and ν controls weight for localization based on local kernel vector $\boldsymbol{x}$. In the following section, we will introduce an adaptive selection procedure for β and ν.

5.4.3 Adaptive Selection for Tuning Parameters β and ν

Let us discuss a selection procedure for tuning parameters based on a given data set. We observe that the performance of the proposed method depends on the value of the tuning parameter β and ν, where ν plays the key role for local PCA. To obtain better performance by this method, we will propose an adaptive selection procedure for ν by fixing β as β_0 everywhere. To find an appropriate ν, we evaluate the estimates by various values of ν. Minami and Eguchi (2003) used β-divergence with a fixed value of β as a measure for evaluation of the minimum β-divergence estimator for robust ICA. Following them, we are proposing a modified loss function defined by equation 5.43 with a fixed value of ν denoted by ν_0. We define a measure for evaluating our estimators for the mean vector $\boldsymbol{\mu}$ and orthogonal matrix Γ as

$$D_{\beta_0,\nu_0}(\nu) = \mathrm{E}\left\{L_{\beta_0,\nu_0}\left(\widehat{\boldsymbol{\mu}}_{\beta_0,\nu}, \widehat{\Gamma}_{\beta_0,\nu}\right)\right\} \tag{5.52}$$

where

$$(\widehat{\boldsymbol{\mu}}_{\beta_0,\nu},\ \widehat{\Gamma}_{\beta_0,\nu}) = \underset{\boldsymbol{\mu},\ \Gamma}{\operatorname{argmin}}\ L_{\beta_0,\nu}(\boldsymbol{\mu}, \Gamma) \tag{5.53}$$

The measurement $D_{\beta_0,\nu_0}(\nu)$ is of the generalization performance of a estimator. The generalization performance relates to its prediction capability on independent test data. If we use the same dataset to evaluate $D_{\beta_0,\nu_0}(\nu)$ as to estimate a recovering matrix, it will underestimate $D_{\beta_0,\nu_0}(\nu)$. If we are in a data-rich situation, the best approach is to divide the dataset into a few parts, and use one set for estimation and another for evaluation. In other situations, a simple and widely used method by sample re-use is the **K-fold Cross Validation (CV)** method (Hastie et al., 2001). The K-fold CV method uses part of the available data to find the estimate and a different part to test it. For the current problem,

we employ the K-fold CV method as a generalization scheme. We split the data into K approximately equal-sized and similarly distributed sections. For the k th section, we find the estimate using the other $K-1$ parts of the data, and calculate the β_0-divergence for the k th section of the data. Then we combine the calculated β_0-divergence values to obtain the CV estimate.

Table 5.1: K-Fold Cross Validation Procedure

Split the data set $\mathscr{D}$ into K subsets; $\mathscr{P}(1), \cdots, \mathscr{P}(K)$.

Let $\mathscr{P}^{-k} = \{\boldsymbol{x}(t) | \boldsymbol{x}(t) \notin \mathscr{P}(k)\}$.

For $k = 1, \cdots, K$

- Estimate $\boldsymbol{\mu}$, V and Γ by minimizing $D_\beta\Big(p(\boldsymbol{x}(t)), \kappa\varphi_{\boldsymbol{\mu},V}(\boldsymbol{x}(t))\Big)$ using $\mathscr{P}^{-k}$, that is

$$(\widehat{\boldsymbol{\mu}}_{\beta_0,\nu},\ \widehat{\Gamma}_{\beta_0,\nu}) = \underset{\boldsymbol{\mu},\ \Gamma}{\operatorname{argmin}} \sum_{\boldsymbol{x}(t)\in\mathscr{P}^{-k}} \Psi_{\beta_0,\nu}\Big(\boldsymbol{x}(t); \boldsymbol{\mu}, \Gamma, \boldsymbol{x}\Big).$$

- Compute $\mathrm{CV}_{(k)}$ using $\mathscr{P}(k)$,

$$\mathrm{CV}_{(k)} = \sum_{\boldsymbol{x}(l)\in\mathscr{P}(k)} \Psi_{\beta_0,\nu_0}\Big(\boldsymbol{x}(t); \widehat{\boldsymbol{\mu}}_{\beta_0,\nu},\ \widehat{\Gamma}_{\beta_0,\nu},\ \boldsymbol{x}\Big).$$

End

Then, $\widehat{D}_{\beta_0,\nu_0}(\nu) = \dfrac{1}{n}\displaystyle\sum_{k=1}^{K} \mathrm{CV}_{(k)}$

Table 1 summarizes the procedure to find the K-fold CV estimate $\widehat{D}_{\beta_0,\nu_0}(\nu)$.

5.4.4 How to Decide ν

As a measure for the variation of $\widehat{D}_{\beta_0,\nu_0}(\nu)$, we compute

$$\mathrm{SD}_{\beta_0,\nu_0}(\nu) = \text{the standard error of } \frac{1}{|\mathscr{D}(k)|}\mathrm{CV}_{(k)},$$

where $|\mathscr{D}(k)|$ denotes the number of elements in the k-th part of data $\mathscr{D}(k)$. Plots of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ for ν with the auxiliary boundary curves $\widehat{D}_{\beta_0,\nu_0}(\nu) \pm \mathrm{SD}_{\beta_0,\nu_0}(\nu)$ will help us to judge an optimum ν. We denote this optimum ν by ν_{opt}. Often we have to employ the

upper auxiliary boundary curve (UABC) with the curve of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ in order to choose ν_{opt}. If the curve of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ is flat for a wide range of ν, then $\beta_{\text{opt}} = 0$. When more than one data class or outliers exist in the entire data space, typical shapes of curves of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ that enables us to chose an appropriate value ν are elbow and dipper shapes. So, if the curve does not have these shapes, we increase the value of ν_0. If these shapes do not appear for any β_0, then $\beta_{\text{opt}} = 0$, (Minami and Eguchi, 2003). If the curve of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ looks elbow and dipper shapes, we choose the smaller one instead of the smallest ν as the β_{opt} whose evaluated value $\widehat{D}_{\beta_0,\nu_0}(\nu_{\text{opt}})$ is not larger than the value of UABC that corresponds to the smallest value of $\widehat{D}_{\beta_0,\nu_0}(\nu)$. However, there is no theoretical justification for this rule, which is known as the one-standard error rule (Hastie et al., 2001). Note that fixed ν_0 should be larger than optimum ν.

5.5 Simulation and Discussion

The performance of the proposed method depends on the value of the tuning parameter β and ν, where the last one is fixed as ν_0 by inverse of the half of diameter of data heuristically. The fixed value ν_0 is used for fast localization to a data cluster by the shifting parameter $\boldsymbol{\mu}$ and local kernel vector $\boldsymbol{x}$. The optimum value of the tuning parameter β plays the key role to provide significant weight to the local data points and insignificant weight to the other data points. The data point corresponding to the insignificant weight has no influence on the estimation. To demonstrate the performance of the proposed algorithm, we generated the following data sets by formula (5.3) using different coefficient matrices A_k and bias vectors $\boldsymbol{b}_k$.

Dataset 1 : Two-dimensional, two-class mixtures (Figure 5.1(a)) generated with Gaussian random numbers. 500 samples were drawn from each class to make 1000 samples in total.

Dataset 2 : Five-dimensional, two-class mixture generated with Gaussian random numbers. Plots of two observed signals are shown in Figure 5.4 using the combination rule. 500 samples were generated from each class to make 1050 samples in total.

Dataset 3 : Two-dimensional, five-class mixture generated with Gaussian random numbers. 100 samples were generated from each class. Twenty (20) outliers (*) were added to make 520 samples in total.

5.5.1 Simulation With Randomly Generated Synthetic Data

Datasets 1, 2 and 3 are randomly generated synthetic datasets. Figures (5.1 - 5.11) represent the simulation results for these datasets. Datasets 1 and 2 consists of 2 data clusters, where one cluster is represented by the symbol "." and the other one by the symbol "o". Dataset 4 consist of 5 clusters that will be explained later. Let us start the simulation with the data set 1 which consist of two clusters (Figure 5.1(a)), where each cluster represent the linear relationship between two variables. To estimate principal components (PCs), first we apply classical method. Figure 5.1(b) shows the scatter plot between two PCs. We see that transformed data set also consist of two clusters, where each cluster shows that estimated components are highly correlated of each other, which contradicts with the uncorrelatedness properties of PCA. Therefore, classical method is not so good to estimate local PCs from data set 1. Then, we compute local PCs shown in figures 5.1(c-d) by the maximum likelihood estimator (MLE) of Gaussian mixture (GM) distribution. We see that output components shown in figure 5.1(c) consist of two clusters, where one cluster 'o' represent that estimated components are highly correlated of each other, while the the cluster '.' satisfies the uncorrelatedness properties of PCA. The last one also satisfy the another important property that first PCs has the largest variance. Similarly, output components shown in figure 5.1(d) also consist of two clusters, where one cluster '.' represent that estimated components are highly correlated of each other, while the other one 'o' satisfies the uncorrelatedness and variance properties of PCA. Therefore, local PCA based on GM distribution is good for data set 1. Figures 5.3(a-b) shows the class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point. Then we apply minimum β-divergence method using a local kernel function for the same purpose. For this, to select an optimum kernel parameter ν, we computed $\widehat{D}_{\beta_0,\nu_0}(\nu)$ with $(\beta_0 = 0.2, \nu_0 = 0.4)$ for ν varying from 0 to 0.5 by 0.05 using 10-fold CV algorithm given in table 1. In the plots of $\widehat{D}_{\beta_0,\nu_0}(\nu)$, asterisks (*) are $\widehat{D}_{\beta_0,\nu_0}(\nu)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are $\widehat{D}_{\beta_0,\nu_0}(\nu) \pm SD_{\beta_0,\nu_0}(\nu)$. By the 'one-standard error rule', we chose $\nu = 0.15$ using Figure 5.1(g) for step 1. Figure 5.1(e) shows the scatter plot between two PCs at step 1. We see that transformed data set also consist of two clusters, where one cluster '.' satisfies both uncorrelatedness and variance properties of PCA like local PCA based GM distribution. After step 1, the value of termination index TI is 0.48, so computation by the minimum β-divergence method not yet finished. At step 2, we used $\beta_0 = 0.2, \nu_0 = 0.4$ again for selection of optimum ν. By the

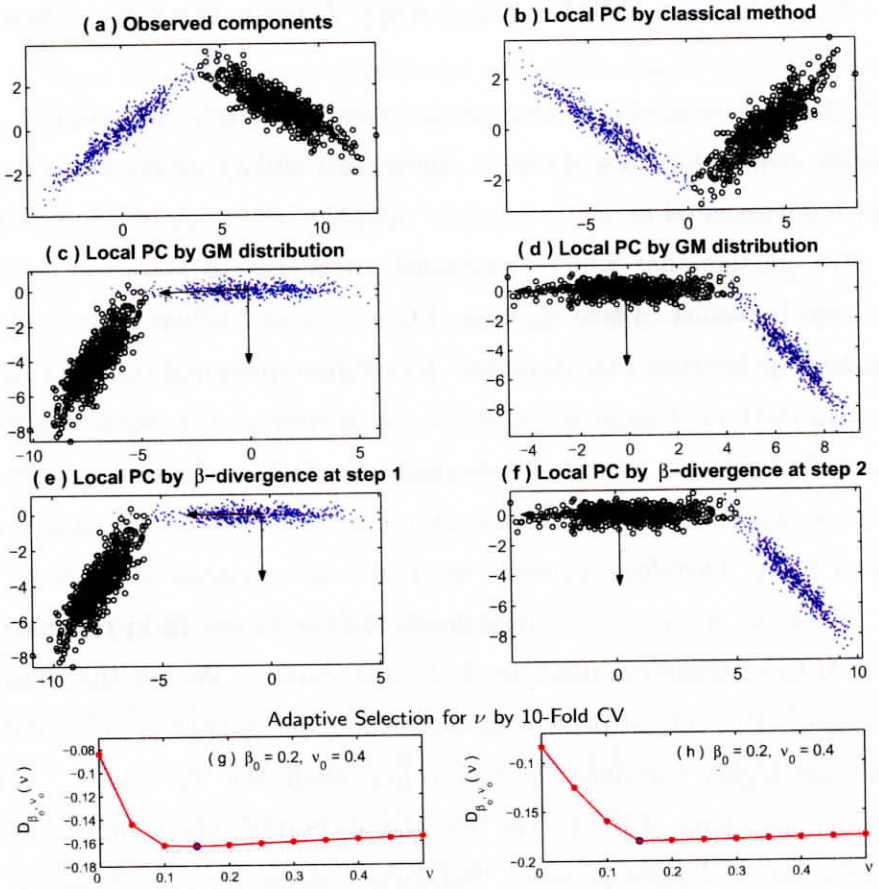

Figure 5.1: For dataset 1, (a) Observed components. (b) Local principal component (PC) based classical Method. (c-d) Local PC based on Gaussian mixture (GM) distribution. (e-f) Local PC based on β-divergence at step 1 and 2, respectively. (g-h) Plots of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ with $\beta_0 = 0.2, \nu_0 = 0.4$ by K-fold CV at step 1 and 2, respectively.

'one-standard error rule', we chose $\nu = 0.2$ using Figure 5.1(h). Then we apply the minimum β-divergence method again changing the initial value of the shifting parameter vector μ and the local kernel vector $\boldsymbol{x}$ both by (5.48). Figure 5.1(f) shows the scatter plot between two PCs at step 2. We see that transformed data set consist of two clusters as previous, where one cluster 'o' satisfies both uncorrelatedness and variance properties of PCA like the cluster '.' at step 1. After step 2, TI=0.95. So sequential estimation by the minimum β-divergence method is terminated. Figures 5.3(c-d) show the weight of each data point corresponding to the estimates at step 1 and 2, respectively. One can see that at each step, one class of data were used and the other class of data totally were ignored by the weight function (5.47) for estimating Γ and $\boldsymbol{\mu}$. The arrows in Figures 5.1(c-f) represent the center of local PCs.

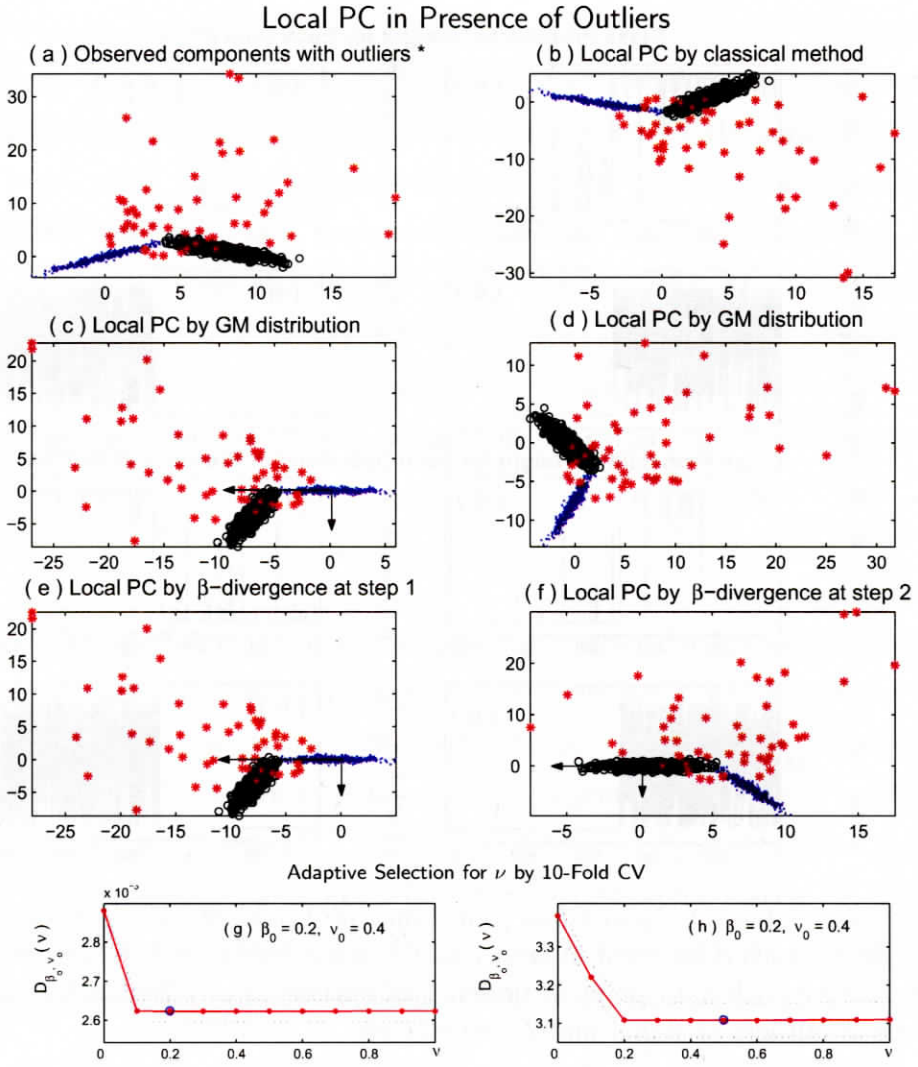

Figure 5.2: For dataset 1 with outliers (*), (a) Observed components with outliers. (b) Local principal component (PC) based classical Method. (c-d) Local PC based on Gaussian mixture (GM) distribution. (e-f) Local PC based on β-divergence at step 1 and 2, respectively. (g-h) Plots of $\widehat{D}_{\beta_0,\nu_0}(\nu)$ with $\beta_0 = 0.2, \nu_0 = 0.4$ by K-fold CV at step 1 and 2, respectively.

respectively. Comparing figures 5.1(e-f) with 5.1(c-d), we see that performance of minimum β-divergence method for local PCA is almost equivalent to local PCA based on GM distribution for data set 1.

Two investigate robustness of the proposed method, we added 50 outliers (*) from the exponential distribution to make 1050 samples points in data set 1 shown in figure 5.2(a). To estimate principal components (PCs), first we apply classical method. Figure 5.2(b) shows

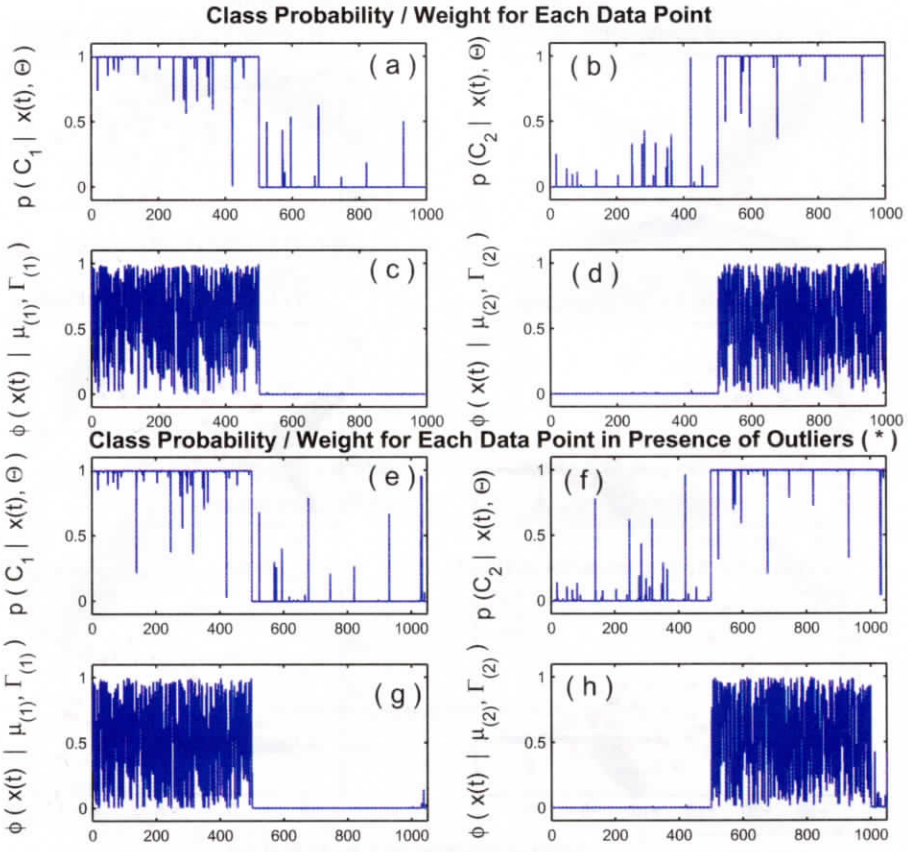

Figure 5.3: For dataset 1, (a-b) Class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point. (c-d) Weight for each data point at step 1 and 2, respectively. (e-f) Class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point in presence of outliers. (g-h) Weight for each data point in presence of outliers at step 1 and 2, respectively.

the scatter plot between two PCs. We see that result is not so good under the previous discussion for data set 1 with outliers. Then, we compute local PCs shown in figures 5.2(c-d) by the maximum likelihood estimator (MLE) of Gaussian mixture (GM) distribution. We see that local PCs belonging to the cluster '.' in figure 5.2(c) are good as previous, however, local PCs represented by figure 5.2(d) are not so good by the previous discussion. Therefore, local PCA based on GM distribution is not so good for data set 1 in presence of outliers. Figures 5.3(e-f) shows the class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point. Then we apply minimum β-divergence method using a local kernel function for the same purpose. To select an optimum kernel parameter ν, we computed $\widehat{D}_{\beta_0,\nu_0}(\nu)$ with $(\beta_0 = 0.2, \nu_0 = 0.4)$ for ν varying from 0 to 1 by 0.1 using 10-fold CV algorithm given in table 1. By the 'one-standard error rule', we chose $\nu = 0.15$ and 0.25 using Figure 5.2(g-h) for step 1 and 2, respectively.

Figure 5.2(e-f) shows the scatter plot between two PCs at step 1 and 2, respectively . We see that local PCA in both step is good by the uncorrelatedness and variance properties of PCA. After step 2, TI=0.92. So sequential estimation by the minimum β-divergence method is terminated. Figures 5.3(g-h) show the weight of each data point corresponding to the estimates at step 1 and 2, respectively. One can see that at each step, one class of data were used and the other class of data totally were ignored by the weight function (5.47) for estimating Γ and μ. The arrows in Figures 5.2(c-f) represent the center of local PCs, respectively. Comparing figures 5.2(e-f) with 5.2(c-d), we see that performance of minimum β-divergence method for local PCA is batter than local PCA based on GM distribution for data set 1 with outliers.

To investigate the performance of the proposed procedure for high-dimensional data, we considered five-dimensional two-class mixture data as dataset 2, which contains 1000 sample points in total. With projection of observed data onto two-dimensional coordinates, two classes are overlapped as shown in figure 5.4(a). To generate data set 2 by (5.3), we used coefficient matrices $A_1 = \text{diag}(2.5, 2.4, 0.90, 0.70, 0.40)$ and $A_2 = \text{diag}(1.0, 0.9, 0.8, 0.4, 0.3)$. Therefore, to estimate principal components, the true orthogonal matrix (Γ) will be identity matrix (**I**) for both clusters, that is $\Gamma = (\gamma_1, \gamma, ..., \gamma_q) = \mathbf{I}$. An estimate $\widehat{\Gamma} = (\widehat{\gamma}_1, \widehat{\gamma}, ..., \widehat{\gamma}_q)$ will be good for Γ if the inner product between γ_i and $\hat{\gamma}_j$ satisfy

$$\gamma_i^T \hat{\gamma}_j = 1, \quad \text{for } i = j \tag{5.54}$$

$$= 0, \quad \text{for } i \neq j \tag{5.55}$$

Also we estimate the estimating error of each $\hat{\gamma}_i$ by

$$EE(i) = \|\gamma_i - \hat{\gamma}_i\|^2 \geq 0, \quad (\ i = 1, 2, ..., q\ ) \tag{5.56}$$

equality hold iff $\gamma_i = \hat{\gamma}_i$. To apply minimum β-divergence method in dataset 2, we used $\beta_0 = 0.2, \nu_0 = 0.3$ for selection of optimum ν for steps 1 and 2 both as previous. By the 'one-standard error rule', we chose optimum $\nu = 0.1$ and 0.15 for steps 1 and 2, respectively using figures 5.5(a-b). The sequential estimating procedure was terminated after step 2 with termination index $TI = 0.97$. Figures 5.5(c-d) show the inner products (IP) between true vector γ_i and its estimates $\hat{\gamma}_i$, $(i = 1, 2, .., 5)$ at step 1 and 2, respectively. Note that in each plot, dash-dot line without marker style means that estimation based on classical method,

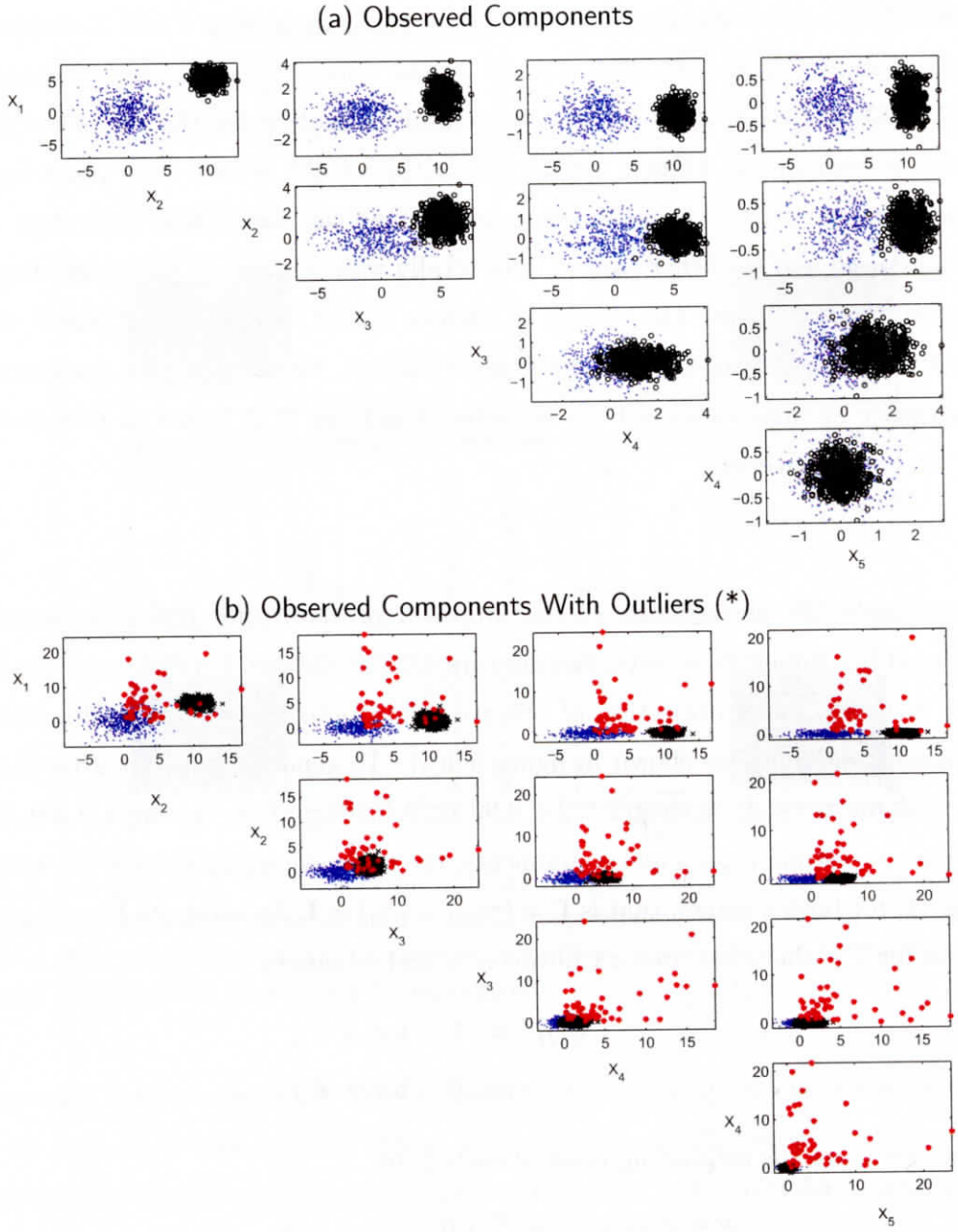

Figure 5.4: For dataset 2, (a) Scatter plot of observed components. (b) Scatter plot of observed components with outliers (*).

solid line with marker style (o) indicates estimation based on GM distribution and dashed line means the simulation based on the minimum β-divergence method. Clearly, we see that classical estimator does not satisfy $\gamma_i^T \hat{\gamma}_i = 1$ for $i = 1, 2$, while the estimators obtained by other two methods satisfy it for all $i = 1, 2, ..., 5$. Also from Figures 5.5(e-f), we see that

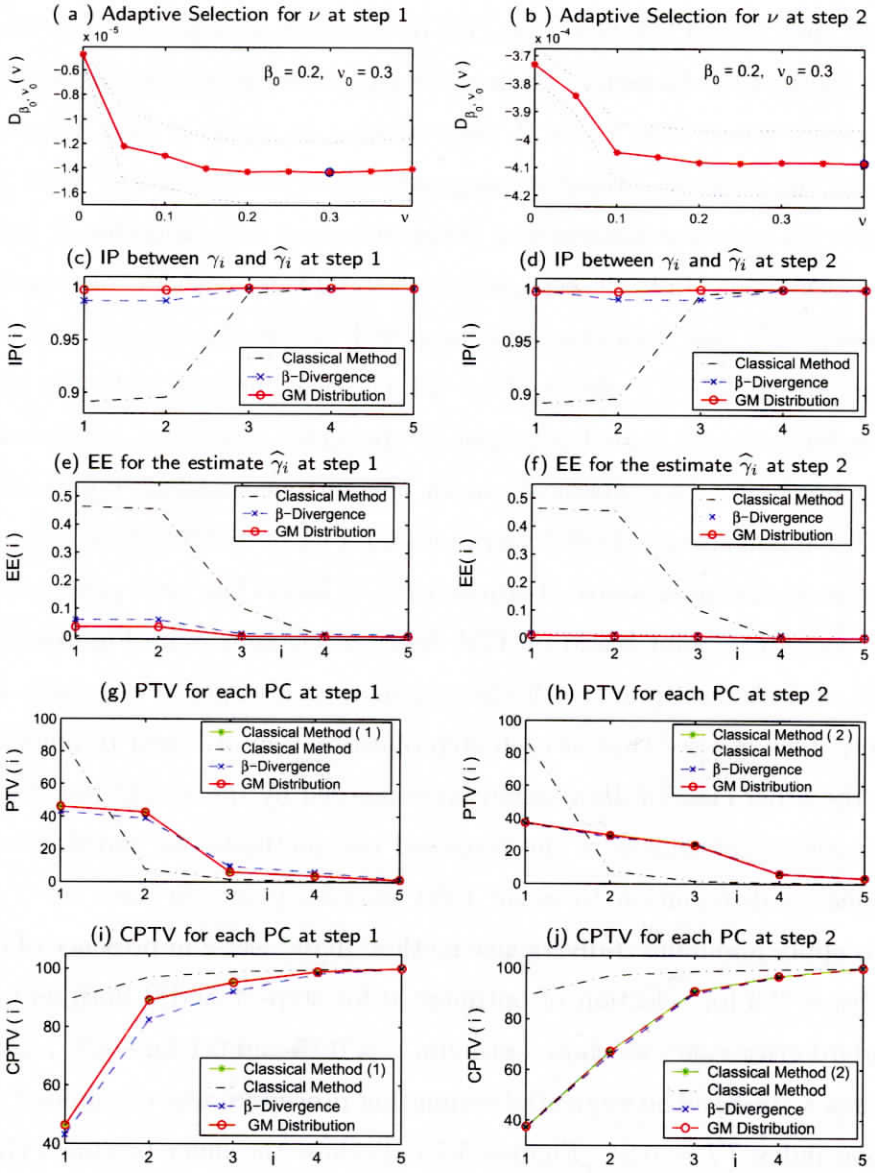

Figure 5.5: For dataset 2, (a-b) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.2, \nu_0 = 0.3$ for ν by 10-fold CV at step 1 and 2, respectively. (d-e) Inner product (IP) between true column vector and the estimated column vector of Γ at step 1 and 2 . (f-g) Estimating error (EE) for $\hat{\gamma}_i$, (i=1,2,...,5). (c-d) Percentage of total variation (PTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively. (e-f) Cumulative percentage of total variation (CPTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively.

estimating error (EE) with classical method is high for $i = 1, 2$, while EE with other two methods are almost close to zero for all $i = 1, 2, ..., 5$. Therefore, local PCA based on GM distribution and minimum β-divergence method both are better than classical method. It

is also seen that local PCA based on GM distribution is slightly better than minimum β-divergence method for dataset 2. Figures 5.5(g-h) represent the percentage of total variation (PTV) for each PC at step 1 and 2, respectively. At steps 1, solid line with marker style (*) represent the classical estimates using only data class 1, while at steps 2, solid line with marker style (*) represent the classical estimates using only data class 2 and other lines are described as previous. In both steps, we see that PTV for each PC obtained by the classical method using only one data class and proposed two methods including all data class are almost similar, while PTV obtained by the classical method including all data class are not similar for first and second principal components. Therefore, proposed two local PCA algorithms are better than classical one in our current context. Figures 5.5(i-j) represent the cumulative percentage of total variation (CPTV) by the principal components obtained by the methods discussed above. Figures 5.7(a-b) shows the class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point based on GM distribution approach. Figures 5.7(c-d) represent the weight of each data point with the minimum β-divergence estimator at step 1 and 2, respectively. One can see that at each step of estimation of Γ and $\boldsymbol{\mu}$, one class of data were used and the other class of data totally were ignored by the weight function (5.47) .

Two investigate robustness of the proposed two methods, we added 50 outliers (*) from the exponential distribution to make 1050 samples points in data set 2 shown in figure 5.4(b). To apply minimum β-divergence method in dataset 2 in presence of outliers, we used $\beta_0 = 0.2, \nu_0 = 0.3$ for selection of optimum ν for steps 1 and 2 both as previous. By the 'one-standard error rule', we chose optimum $\nu = 0.15$ and 0.1 for steps 1 and 2, respectively using figures 5.7(a-b). The sequential estimating procedure was terminated after step 2 with termination index $TI = 0.94$. Figures 5.7(c-d) show the inner products (IP) between true vector γ_i and its estimates $\hat{\gamma}_i$, $(i = 1, 2, .., 5)$ at step 1 and 2, respectively. Note that in each plot, dash-dot line without marker style means that estimation based on classical method, solid line with marker style (o) indicates estimation based on GM distribution and dashed line means the simulation based on the minimum β-divergence method. Clearly, we see that estimators based on classical method or GM distribution do not satisfy the condition $\gamma_i^T \hat{\gamma}_i = 1$ for $i = 1, 2$, while the estimators obtained by minimum β-divergence method satisfy this condition for all $i = 1, 2, ..., 5$. Also from figures 5.7(e-f), we see that estimating error (EE) with classical method and GM distribution both, is large for $i = 1, 2$, while EE with minimum β-divergence methods is almost close to zero for all $i = 1, 2, ..., 5$. Therefore,

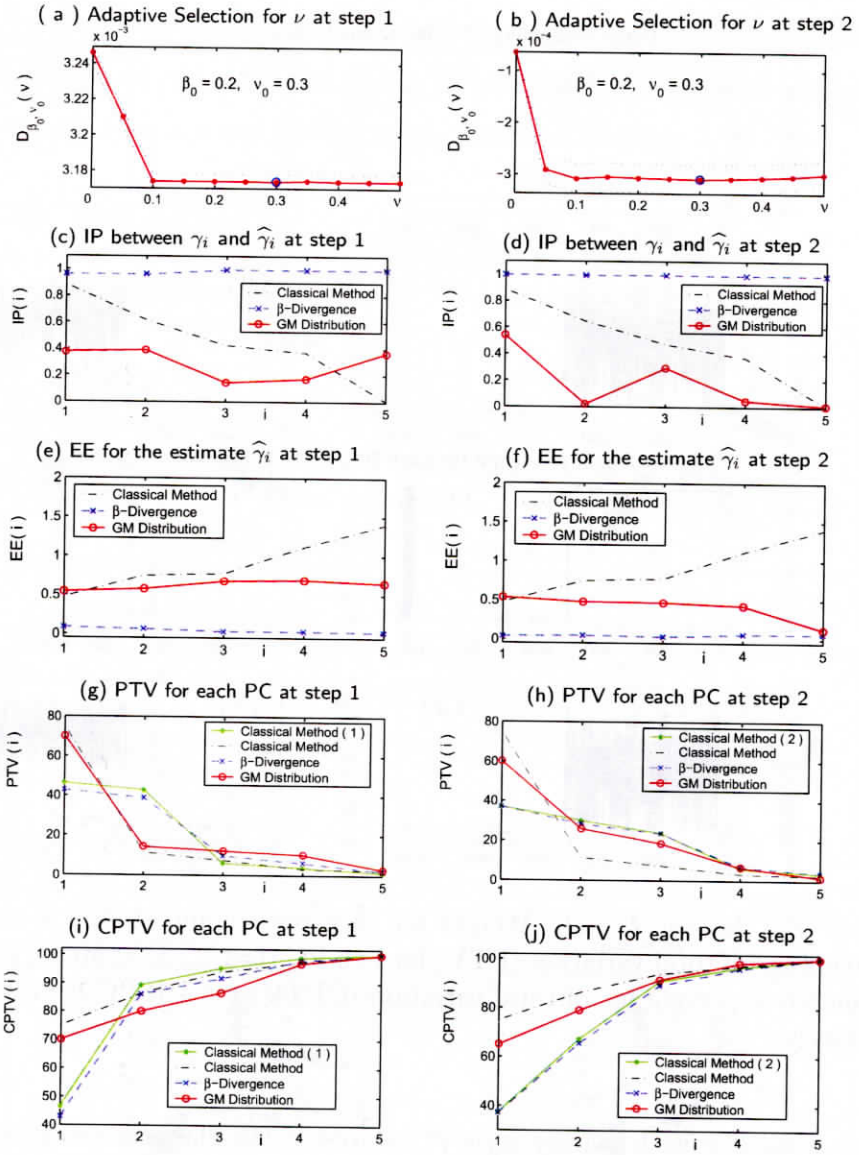

Figure 5.6: For dataset 2 with outliers, (a-b) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.2, \nu_0 = 0.3$ for ν by 10-fold CV at step 1 and 2, respectively. (d-e) Inner product (IP) between true column vector and the estimated column vector of Γ at step 1 and 2 . (f-g) Estimating error (EE) for $\hat{\gamma}_i$, (i=1,2,...,5). (c-d) Percentage of total variation (PTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively. (e-f) Cumulative percentage of total variation (CPTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively.

local PCA based on minimum β-divergence approach is better than both classical and GM distribution approaches for dataset 2 in presence of outliers. Figures 5.7(g-h) represent the percentage of total variation (PTV) for each PC at step 1 and 2, respectively. At steps 1, solid line with marker style (*) represent the classical estimates using only data class 1, while

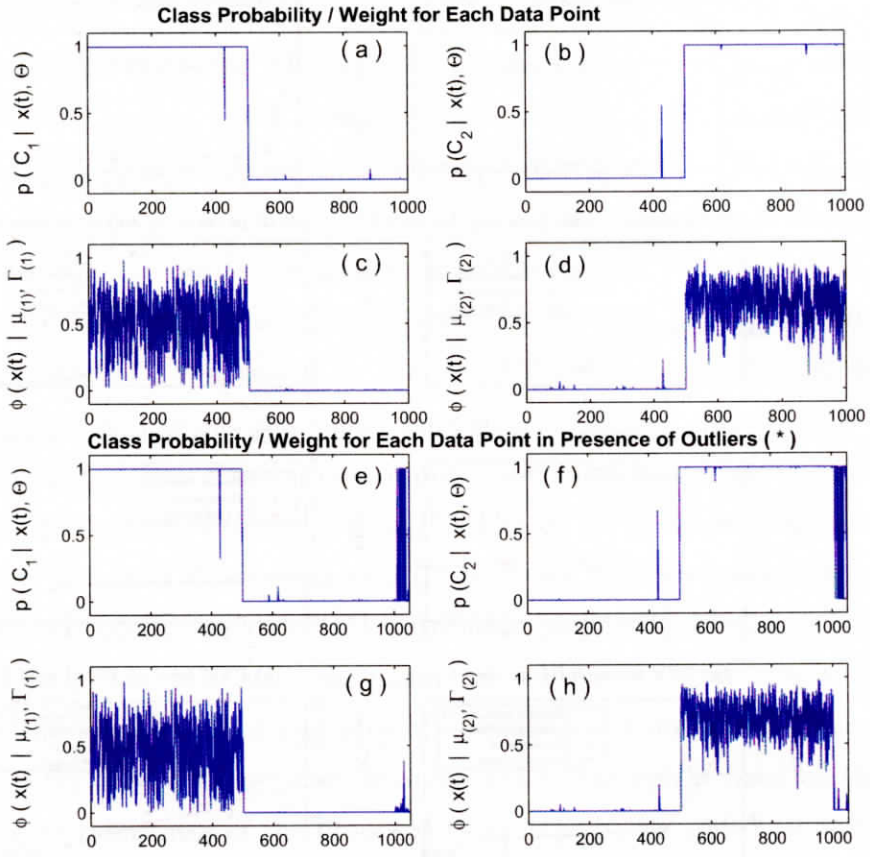

Figure 5.7: For dataset 2, (a-b) Weight for each data point at step 1 and 2, respectively. (c-d) Percentage of total variation (PTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively. (e-f) Cumulative percentage of total variation (CPTV) for i-th PC (i=1,2,..5) at step 1 and 2, respectively.

at steps 2, solid line with marker style (*) represent the classical estimates using only data class 2 and other lines are described as previous. In both steps, we see that PTV for each PC obtained by the classical method using only one data class and minimum β-divergence method using all data classes including outliers are almost similar, while PTV obtained by the classical approach or GM distribution approach using all data classes including outliers are not similar for first and second principal components. Therefore, minimum β-divergence method for local PCA is better than the other two method in our current context for dataset 2 in presence of outliers. Figures 5.7(i-j) represent the cumulative percentage of total variation (CPTV) by the principal components obtained by the methods discussed above. Figures 5.7(e-f) shows the class probability $p(C_k \mid \boldsymbol{x}_t, \Theta)$, k=1,2 for each data point based on GM distribution approach. Figures 5.7(g-h) represent the weight of each data point with the

minimum β-divergence estimator at step 1 and 2, respectively. Same as previous, one can see that at each step of estimation of Γ and $\boldsymbol{\mu}$, one class of data were used and the other class of data totally were ignored by the weight function (5.47) .

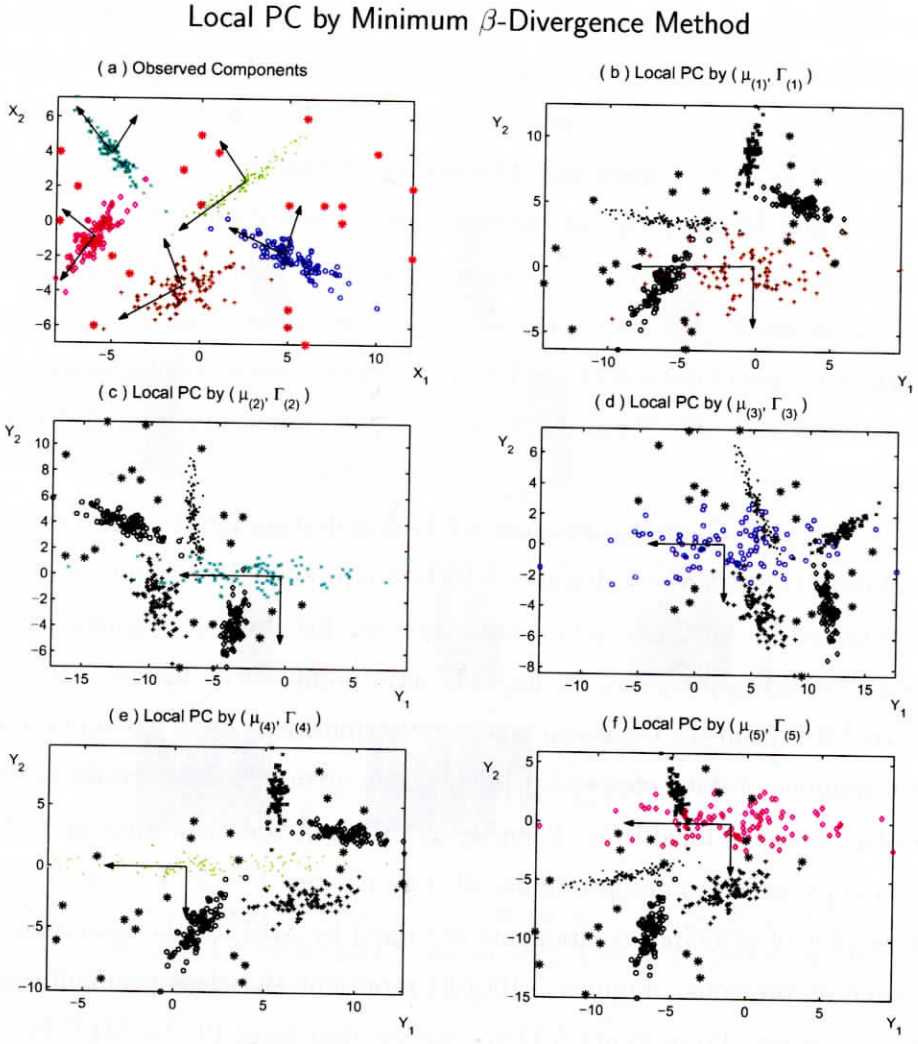

Figure 5.8: For dataset 3, (a) Scatter plot of observed data. (b-f) Local PC estimated by $(\boldsymbol{\mu}_{(k)}, \Gamma_{(k)})$, $k = 1, 2, ...5$, respectively.

To demonstrate the validity of the proposed methods for mixtures of several classes, we considered two-dimensional, five class mixture of synthetic data shown in Figure 5.8(a), where each class represent the linear relationship between two variables. To estimate principal components (PCs) by the proposed method, we chose optimum $\nu = 0.2, 0.25, 0.15, 0.25\&0.2$

with $\beta_0 = 0.5, \nu_0 = 0.5$ at step 1 to 5, respectively, by the K-fold CV plots (K=10) shown in Figures 5.9(g-k), respectively. Figures 5.8(b-f) show the scatter plot between first PC (Y_1) and second PC (Y_2) estimated by $(\boldsymbol{\mu}_{(k)}, \Gamma_{(k)})$ at step $k = 1, 2, ...5$, respectively. We see that transformed data set also consist of five classes in each plot, where one class in each plot represent that estimated components are uncorrelated of each other and the first principal component has the largest variance, while the components of other classes do not satisfy the properties of PCA. After step 5, termination index TI=0.96. So sequential estimation by the proposed method is terminated. The arrows in Figure 5.8(a) represent the orthogonal direction obtained by the proposed method, while the arrows in Figures 5.8(b-f) represent the center of the local PCs. Figures 5.9(a-e) show the weight of each data point corresponding to the estimates at step 1 to 5, respectively. One can see that at each step of estimation of Γ and $\boldsymbol{\mu}$, one class of data were used and the other classes of data totally were ignored by the weight function (5.47). Figures 5.9(f-j) show the cumulative weight after each step from 1 to 5, respectively.

Then we use MLE of Gaussian mixture (GM) distribution for the same purpose with exact number of data clusters $c = 5$. Figures 5.10(b-f) represent the scatter plot of principal components obtained by MLE of the Gaussian mixture distribution. Figures 5.10(g-k) represent the class probabilities $p(C_k \mid \boldsymbol{x}_t, \Theta)$ for each data point. From figures 5.10(b-f), we see that local PC by MLE from the Gaussian mixture distribution is good like minimum β-divergence method if number of data cluster (c) is known in advance. The arrows in figures 5.10(b-f) indicates the center of local PCs. Then we use MLE of Gaussian mixture (GM) distribution for the same purpose assuming number of data clusters $c = 3$. Figures 5.11(b-d) represent the scatter plot of principal components obtained by MLE of the Gaussian mixture distribution same as previous. Figures 5.10(g-k) represent the class probabilities $p(C_k \mid \boldsymbol{x}_t, \Theta)$ for each data point. From figure 5.11(b), we see that local PC by MLE from the Gaussian mixture distribution is good like minimum β-divergence method if number of data cluster (c) is unknown in advance, however, other figures 5.11(c-d) show the misleading results. The arrows in figures 5.11(b-d) indicates the center of local PCs. Thus we may conclude that local PCA based on Gaussian mixture distribution is good only when number of data clusters (c) is known in advance.

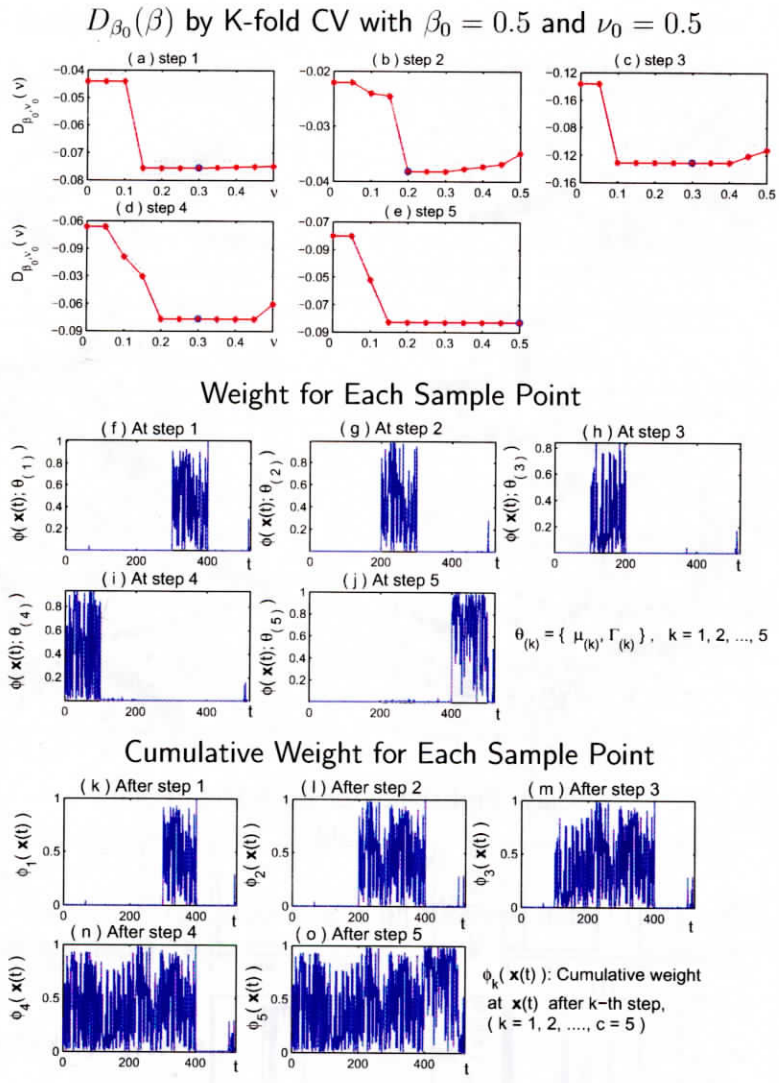

Figure 5.9: For dataset 3, (g-k) Plots of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3$ by K-fold CV at step 1 to 5, respectively. (a-e) Weight for each data point at step 1 to 5, respectively. (f-j) Cumulative weight at step 1 to 5, respectively

5.6 Conclusions

We proposed a method for exploring local structure of PCA mixture model for dimensionality reduction based on the minimum β-divergence method using a local kernel function. The proposed procedure searches the orthogonal matrix of each local class for PCA based on the initial conditions of the shifting parameter and a local kernel vector. If the initial value of the shifting parameter vector $\boldsymbol{\mu}$ and the local kernel vector $\boldsymbol{x}_o$ belongs to a data class, then the minimum β-divergence estimator finds the estimates of the orthogonal matrix and

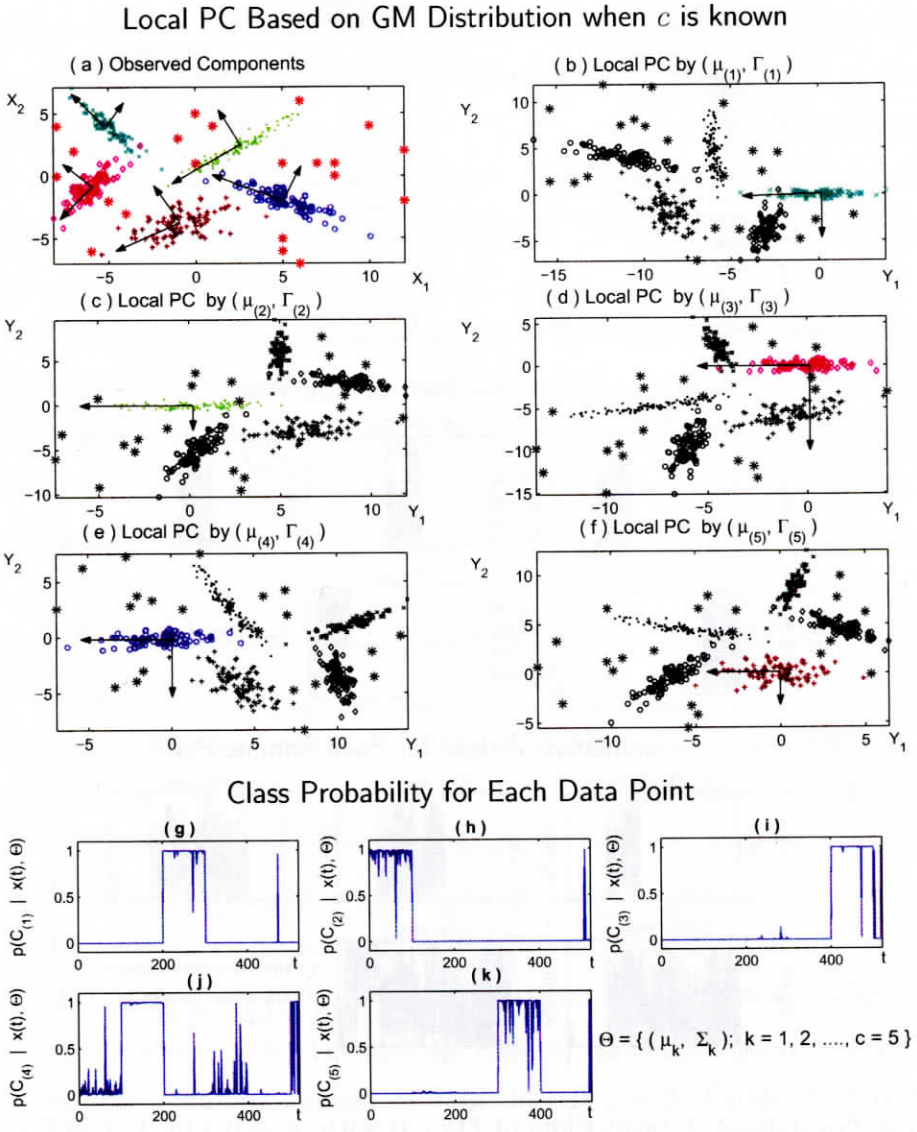

Figure 5.10: For dataset 3, (a) Observed components. (b-f) Local PC estimated by $(\boldsymbol{\mu}_{(k)}, \Gamma_{(k)})$, $k = 1, 2, ...5$, respectively.

shifting parameter for this class. In order to obtain estimates of the recovering matrix and the shifting parameter for other data classes, the initial value of the shifting parameter is changed according to the observed vector having the α-percentile cumulative weight. Using the proposed method, all local structures can be explored sequentially from the entire data space. We suggested a termination index for the proposed method based on the cumulative weight. On the basis of our simulation results, the value of the termination index (TI) should be greater than 0.90 to terminate the classification procedure.

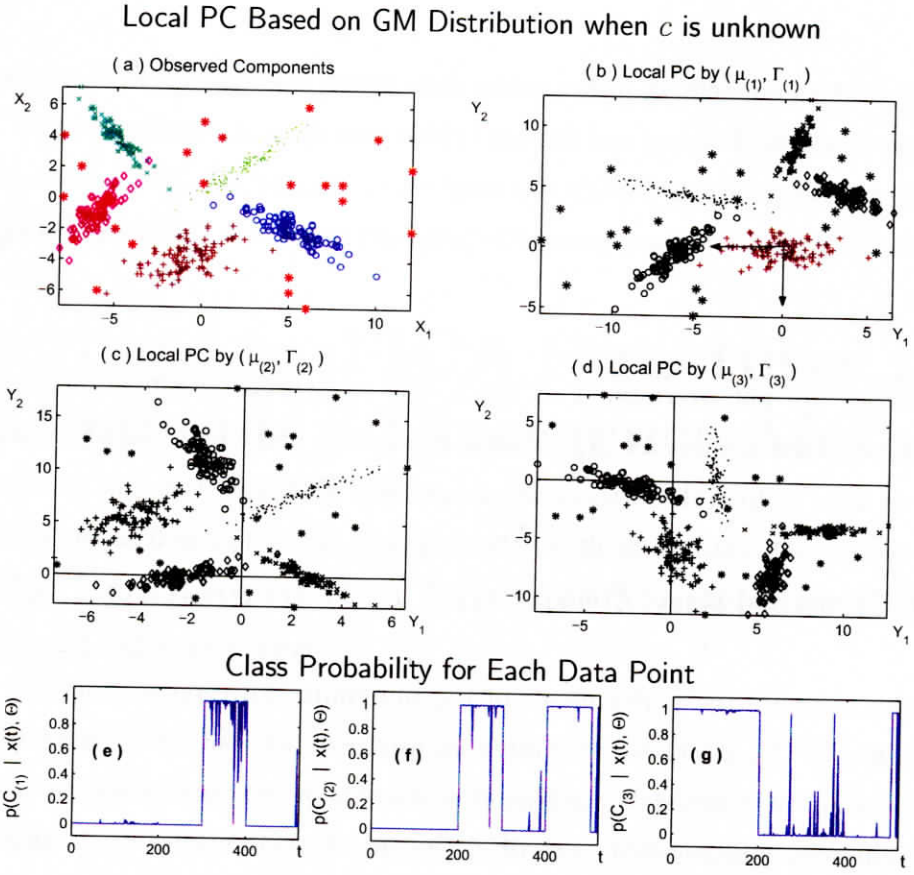

Figure 5.11: For dataset 3, (a) Scatter plot of observed data. (b-f) Local PC estimated by $(\boldsymbol{\mu}_{(k)}, \Gamma_{(k)})$, $k = 1, 2, 3$, respectively.

The performance of the proposed method depends on the value of the tuning parameter β and ν, where ν plays the key rule for local PCA. We used an adaptive selection procedure for ν keeping fixed β as β_0 everywhere in the simulation study by the cross validation. The modified loss function defined by (5.52) with fixed value β_0, ν_0 is used as a measure for evaluation of the proposed estimators by different values of ν. $D_{\beta_0,\nu_0}(\nu)$ for different values of ν were estimated by K-fold cross-validation summarized in Table 1.

The main purpose of the proposed method is similar to the conventional PCA mixture models proposed by Tipping and Bishop (1999)). The procedure proposed by Tipping and Bishop (1999) finds all local PCA structures simultaneously, whereas the method proposed herein finds all local PCA structures sequentially.

If number of data clusters, c, in the entire data space $\mathscr{D}$ is unknown, then the conventional method proposed by Tipping and Bishop (1999) may gives misleading results. However, our proposed method does not require the number, c, in advance. Finally, our method is able to estimate c. Another advantages of the proposed method is that it robust against outliers (e.g.5.2).

When classes are not overlapped so much, the sequential classification methods and the Bayes rule will give similar results. If some classes are overlapped lightly, then the proposed method is able to find the orthogonal directions. However, the case in which classes are heavily overlapped is still difficult for the proposed method as well as the model-based local PCA by Tipping and Bishop (1999).

Finally, we compare the performance of the minimum β-divergence method for local PCA with the local PCA based on the Gaussian mixture distribution by simulation study. We found that local PCA based on the Gaussian mixture distribution is slightly better local PCA by the minimum β-divergence method if number of data clusters (c) is known in advance, on the other hand, if number of data clusters (c) is unknown, then local PCA based on the Gaussian mixture distribution is not so good. However, in this case, local PCA by the minimum β-divergence method is good. Also local PCA based on the Gaussian mixture distribution shows misleading results in presence of outliers, while in this situation, minimum β-divergence method shows promising results (e.g.5.2).

Chapter 6

Exploring Local ICA Structures by the Minimum β-Divergence Method

6.1 The Problem of ICA Mixture Models for Exploring Local Structures

Blind source separation (BSS) by independent component analysis (ICA) has been applied in solving various signal processing problems, including speech enhancement, telecommunications, medical signal processing and so forth. Independent component analysis attempts to recover the original sources that have independent and non-Gaussian structure from observable linearly mixed data. In the classical ICA model, all source signal vectors belong to only one source class $\mathscr{S}$, and all mixed signal vectors belong to the same class in the entire data space $\mathscr{D}$. However, in practice, these source vectors may originate from several source classes, and the corresponding mixed signal vectors belong to several classes in the entire data space. In this case, the performance of classical ICA may not be so good. Therefore, Lee *et al.* (2000) proposed an ICA mixture models by modeling the observed data as a mixture of several mutually exclusive classes, each of which is described by linear combinations of independent, non-Gaussian densities. However, one problem encountered when applying this method is that the number of classes c should be known in advance, which is difficult in practice.

We assume that source vectors come from c source classes $\{\mathscr{S}_1, \mathscr{S}_2, ..., \mathscr{S}_c\}$ and that the corresponding mixed signal vectors belong to c different data classes $\{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_c\}$ in the entire data space $\mathscr{D}$, where the number c is unknown. In addition, we assume that the data class $\mathscr{D}_k$ occurs in the entire data space $\mathscr{D}$ due to the source vectors that originate

from the source class $\mathscr{S}_k$, $(k = 1, 2, \ldots, c)$. In other words, source class $\mathscr{S}_k$ is hidden as the data class $\mathscr{D}_k$ in the entire data space $\mathscr{D}$. In practice, the occurrence order of a mixed signal vector in the entire data space $\mathscr{D}$ from a source class is unknown. However, we can assume that an unobservable mixed signal vector $\boldsymbol{z}_{jk} \in \mathscr{D}_k = \{\boldsymbol{z}_{jk};\ j = 1, 2, \ldots, n_k\}$, $(k = 1, 2, \ldots, c;\ \sum_{k=1}^{c} n_k = n)$ follows an ICA model as

$$\boldsymbol{z}_{jk} = A_k \boldsymbol{s}_{jk} + \boldsymbol{b}_k, \tag{6.1}$$

where A_k is an $m \times m$ non-singular mixing matrix, $\boldsymbol{b}_k$ is the bias vector and $\boldsymbol{s}_{jk} \in \mathscr{S}_k = \{\boldsymbol{s}_{jk};\ j = 1, 2, \ldots, n_k\}$, $(k = 1, 2, \ldots, c)$ is the j-th random vector in the source class k with zero mean vector, the components of which are assumed to be independent and non-Gaussian. However, in a practical situation, an observable mixed signal vector $\boldsymbol{x}_t \in \mathscr{D} = \{\boldsymbol{x}_t;\ t = 1, 2, \ldots, n\}$ is obtained as one vector of $\bigcup_{k=1}^{c} \mathscr{D}_k = \{\boldsymbol{z}_{jk};\ j = 1, 2, \ldots, n_k,\ k = 1, 2, \ldots, c;\ \sum_{k=1}^{c} n_k = n\}$ such that $\mathscr{D} = \bigcup_{k=1}^{c} \mathscr{D}_k$. If the permutation of $\{\boldsymbol{z}_{11}, \boldsymbol{z}_{12}, \ldots, \boldsymbol{z}_{jk}, \ldots, \boldsymbol{z}_{n_c c}\}$ into $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ is purely random, then (6.1) reduces to the ICA mixture models. In the ICA mixture models, the observed data in each class are considered to be a linear combination of independent and non-Gaussian sources. (See Lee and Lewicki (2000); Lee, Lewicki and Sejnowski (2000); Lee(2001) for a detailed discussion.) When the data in each class are modeled as multivariate Gaussian, The model is known as a Gaussian mixture model.

One problem with existing method is that it cannot recover hidden classes properly when c is unknown or mis-specified. However, in this difficult situation, our proposed method is sequential application of the minimum β-divergence method (cf. Minami and Eguchi, 2002) to extract all hidden classes sequentially based on a rule of step-by-step change of the shifting parameter. Later, we will propose a stopping rule for repeated application of the minimum β-divergence method based on the cumulative weight. In order to recover k-th hidden class, we estimate a recovering matrix W_k for A_k^{-1} and a shifting parameter $\boldsymbol{\mu}_k$ for $A_k^{-1}\boldsymbol{b}_k$, based on the minimum β-divergence method, initializing $\boldsymbol{\mu}_k$ by a vector $\boldsymbol{x}_0 \in \mathscr{D}_k$ that transforms the mixed signal vector $\boldsymbol{x}_t \in \mathscr{D}$ into a new signal vector $\boldsymbol{y}_t$, $(t = 1, 2, \ldots, n)$ by

$$\boldsymbol{y}_t = W_k \boldsymbol{x}_t - \boldsymbol{\mu}_k, \tag{6.2}$$

where,

$$\begin{aligned} \boldsymbol{y}_t \ &\in\ \widehat{\mathscr{S}_k} = \{\hat{\boldsymbol{s}}_{jk};\ j = 1, 2, \ldots, n_k\}, \qquad \text{if} \quad \boldsymbol{x}_t \in \mathscr{D}_k \\ &\in\ \mathscr{D}^*, \qquad \text{otherwise,} \end{aligned}$$

where $\widehat{\boldsymbol{s}}_{jk}$ is the estimate of the source vector $\boldsymbol{s}_{jk} = A_k^{-1}(\boldsymbol{z}_{jk} - \boldsymbol{b}_k)$. Here, $\widehat{\mathscr{S}_k}$ is the k-th recovered class whose component vectors are classified from the data class $\mathscr{D}_k$, and $\mathscr{D}^*$ is the set corresponding to the unclassified data points. If W_k is properly obtained, then $\widehat{\boldsymbol{s}}_{jk}$ is equal to $\boldsymbol{s}_{jk}$, except for an arbitrary scaling of each signal component and the permutation of the indices. An appropriate value of the tuning parameter β is a key to the proposed method. Therefore, an adaptive selection procedure is proposed for the tuning parameter β.

Section 6.2 reviews the minimum β-divergence method, section 6.3 describes the proposed method for exploring the hidden class. In section 6.3.1, we discuss a selection method for the tuning parameter β. Finally, section 6.4 presents numerical examples, and section 6.5 presents the conclusions of this chapter.

6.2 Minimum β-Divergence Method

Several estimators for ICA can be considered as being derived through the framework of the maximum likelihood estimation, with various choices for density functions. In other words, the estimators are the minimizers of the Kullback-Libler (K-L) divergence between the empirical distribution and a certain form of density function. As for example, Jutten and Herault (1991) heuristic approach, entropy maximization (Bell and Sejnowski, 1995), minimization of cross-cumulants (Cardoso and Souloumiac, 1993), approximation of mutual information by Gram-Charlier expansion, the natural gradient approach (Amari, Chichocki and Yang, 1996) and so forth. Amari and Cardoso (1997) showed that the estimation functions of this type of estimator are unbiased provided that the means of the original signals are zeros. However, this type of estimator is not robust to outliers. Minami and Eguchi (2002) proposed a robust blind source separation method by minimizing β-divergence (Eguchi and Kano, 2001). This method is referred to as the minimum β-divergence method, and the corresponding estimator, is referred to as the minimum β-divergence estimator. Next, we will review the basic formulation of the minimum β-divergence method.

Suppose that an observed signal vector $\boldsymbol{x}$ is a linear transformation of vector $\boldsymbol{s}$ whose components are independent of each other. There exists a matrix W and a shifting parameter vector $\boldsymbol{\mu}$ such that the components of $\boldsymbol{y} = W\boldsymbol{x} - \boldsymbol{\mu}$ are independent of each other. Thus, the joint density of $\boldsymbol{y}$ can be expressed as the product of marginal density functions $q_1, q_2, \ldots, q_m$

by

$$q(\boldsymbol{y}) = \prod_{i=1}^{m} q_i(y_i)$$

and the joint density function of $\boldsymbol{x}$ can be expressed as:

$$r(\boldsymbol{x}, W, \mu) = |\det(W)| \prod_{i=1}^{m} q_i(\boldsymbol{w}_i \boldsymbol{x} - \mu_i), \tag{6.3}$$

where $\boldsymbol{w}_i$ is the i-th row vector of W, and μ_i is the i-th component of $\boldsymbol{\mu}$. The β-divergence between the density of a recovered signal vector and the product of marginal densities (if they are known) would attain the minimum value of zero if and only if the recovered signals are independent of each other. The minimum β-divergence method is an estimating procedure that is based on the empirical β-divergence $\widehat{D}_\beta\big(\tilde{r}, r_0(., W, \boldsymbol{\mu})\big)$ between the empirical distribution $\tilde{r}$ of $\boldsymbol{x}$ and $r_0(\boldsymbol{x}, W, \boldsymbol{\mu})$, rather than the unknown density expressed by (6.3), where

$$r_0(\boldsymbol{x}, W, \boldsymbol{\mu}) \;=\; |\det(W)| \prod_{i=1}^{m} p_i(\boldsymbol{w}_i \boldsymbol{x} - \mu_i). \tag{6.4}$$

Here, p_i is a specific density form, rather than an unknown density q_i, for example, $p_i(z) = c_1 \exp(-c_2 z^4)$ for sub-Gaussian signals and $p_i(z) = c_2 / \cosh(z)$ for super-Gaussian signals. Moreover, the switching scheme of the extended infomax ICA (Lee, Girolami and Sejnowski 1999) between sub-Gaussian and super-Gaussian densities can be adopted if the non-Gaussianities of the source signals are unknown. The minimum β-divergence method finds the minimizer of the empirical β-divergence $\widehat{D}_\beta\big(\tilde{r}, r_0(., W, \boldsymbol{\mu})\big)$. This minimization is equivalent to maximizing the following quasi β- likelihood function:

$$L_\beta(W, \boldsymbol{\mu}) = \frac{1}{n} \sum_{t=1}^{n} l_\beta(\boldsymbol{x}_i; W, \boldsymbol{\mu}) \tag{6.5}$$

where,

$$l_\beta(\boldsymbol{x}; W, \boldsymbol{\mu}) \;=\; \begin{cases} \log\left(r_0(\boldsymbol{x}, W, \boldsymbol{\mu})\right), & \text{for} \quad \beta = 0 \\ \dfrac{1}{\beta} r_0^\beta(\boldsymbol{x}, W, \boldsymbol{\mu}) - b_\beta(W) - \dfrac{1-\beta}{\beta}, & \text{for} \quad 0 < \beta < 1 \end{cases} \tag{6.6}$$

and

$$b_\beta(W) \;=\; \frac{1}{\beta+1} \int r_0^{\beta+1}(\boldsymbol{x}, W, \boldsymbol{\mu}) d\boldsymbol{x} \;=\; \frac{|\det(W)|^\beta}{\beta+1} \int \prod_{i=1}^{m} p_i^{\beta+1}(z_i) d\boldsymbol{z}.$$

The estimating functions (derivatives of $l_\beta(\boldsymbol{x}; W, \boldsymbol{\mu})$) are given by

$$F_1(\boldsymbol{x}, W, \boldsymbol{\mu}) = r_0^\beta(\boldsymbol{x}, W, \boldsymbol{\mu}) \left(I_m - \boldsymbol{h}(W\boldsymbol{x} - \boldsymbol{\mu})\ (W\boldsymbol{x})^T\right) W^{-T} - \beta\ b_\beta(W) W^{-T}, \quad (6.7)$$

$$F_2(\boldsymbol{x}, W, \boldsymbol{\mu}) = r_0^\beta(\boldsymbol{x}, W, \boldsymbol{\mu})\ \boldsymbol{h}(W\boldsymbol{x} - \boldsymbol{\mu}) \quad (6.8)$$

and the estimating equations are as follows:

$$\frac{1}{n}\sum_{t=1}^{n} r_0^\beta(\boldsymbol{x}_t, W, \boldsymbol{\mu}) \left(I_m - \boldsymbol{h}(W\boldsymbol{x}_t - \boldsymbol{\mu})\ (W\boldsymbol{x}_t)^T\right) W^{-T} - \beta\ b_\beta(W) W^{-T} = O, \quad (6.9)$$

$$\frac{1}{n}\sum_{t=1}^{n} r_0^\beta(\boldsymbol{x}_t, W, \boldsymbol{\mu})\ \boldsymbol{h}(W\boldsymbol{x}_t - \boldsymbol{\mu}) = \boldsymbol{0} \quad (6.10)$$

where

$$\boldsymbol{h}(\boldsymbol{y}) = (h_1(y_1), ..., h_m(y_m))^T \quad \text{and}$$

$$h_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} = -\frac{p_i'(y_i)}{p_i(y_i)}.$$

In the estimating function, the multiplicative term

$$r_0^\beta(\boldsymbol{x}, W, \boldsymbol{\mu}) \propto \prod_{i=1}^{m} p_i^\beta(\boldsymbol{w}_i \boldsymbol{x} - \mu_i) \quad (6.11)$$

can be considered to be a weight function that provides a weight for each data point. The weight of each possible outlier is reduced to approximately zero by this weight function. This weight function is a key to robust blind source separation by the minimum β-divergence method. Since β-divergence with $\beta = 0$ is equivalent to the K-L divergence, the minimum β-divergence estimator with $\beta = 0$ is equivalent to the estimator derived from the K-L divergence with explicitly included shift parameters. The minimum β-divergence estimator is locally consistent, as a method derived from the K-L divergence (Minami and Eguchi, 2002).

6.3 New Proposal for Exploring Local ICA Structures by the Minimum β-Divergence Method

Lee, Lewicki and Sejnowski (2000) proposed a method for extracting all hidden classes simultaneously from the mixture of ICA models using the maximum-likelihood method. In this section, we propose an iterative algorithm for the same purpose based on sequential

application of the minimum β-divergence method. The proposed method explores the recovering matrix of each class on the basis of the initial condition of the shifting parameter $\boldsymbol{\mu}$. If the initial value of the shifting parameter is close to the mean of the k-th class, then the estimates for the recovering matrix W_k and the shifting parameter $\boldsymbol{\mu}_k$ can be obtained for this class by considering the data in other classes as outliers. Thus, we can estimate $\{(W_k, \boldsymbol{\mu}_k);\ k = 1, 2, ..., c\}$ by the repeated application of the minimum β-divergence method to recover all hidden classes that are sequentially based on a rule for the step-by-step change of the shifting parameter $\boldsymbol{\mu}$. In order to create a rule for the sequential change of the shifting parameter $\boldsymbol{\mu}$, let us consider the weight function ϕ as

$$\phi(\boldsymbol{x}, W, \boldsymbol{\mu}) = \prod_{i=1}^{m} p_i^{\beta}(\boldsymbol{w}_i \boldsymbol{x} - \mu_i) \tag{6.12}$$

Next, we will discuss a sequential estimating procedure for $\{(W_k, \boldsymbol{\mu}_k);\ k = 1, 2, ..., c\}$ based on a rule of the step-by-step change of the shifting parameter $\boldsymbol{\mu}$.

Step 1: Set the initial value $\widehat{W}_0$ for the recovering matrix W to the identity matrix, and set the initial value $\widehat{\boldsymbol{\mu}}_0$ for $\boldsymbol{\mu}$ to any one vector $\boldsymbol{x}_t \in \mathscr{D}$. Find the estimates for W and $\boldsymbol{\mu}$ by the minimum β-divergence method using these initial values. Let the obtained estimates be denoted as $\widehat{W}_{(1)}$ and $\widehat{\boldsymbol{\mu}}_{(1)}$, respectively.

Let us suppose that $(k-1)$ pairs of estimates

$$\left\{\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right), \left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right), ..., \left(\widehat{W}_{(k-1)}, \widehat{\boldsymbol{\mu}}_{(k-1)}\right)\right\}.$$

are obtained sequentially in steps 1 to $(k-1)$.

Step k: Set the initial value $\widehat{W}_0$ to the identity matrix for the recovering matrix W. As the initial value for the shifting parameter $\boldsymbol{\mu}$, we use the minimizer of the cumulative weight:

$$\widehat{\boldsymbol{\mu}}_0 = \underset{\boldsymbol{x}_t}{\operatorname{argmin}} \sum_{j=1}^{k-1} \phi\left(\boldsymbol{x}_t; \widehat{W}_{(j)}, \widehat{\boldsymbol{\mu}}_{(j)}\right). \tag{6.13}$$

Find the estimates for W and $\boldsymbol{\mu}$ by the minimum β-divergence method using these initial values. Let the obtained estimates be denoted as $\widehat{W}_{(k)}$ and $\widehat{\boldsymbol{\mu}}_{(k)}$, respectively.

Accordingly, the desired estimates are

$$\left\{\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right), \left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right), ..., \left(\widehat{W}_{(c)}, \widehat{\boldsymbol{\mu}}_{(c)}\right)\right\}.$$

In order to recover the hidden classes from the observed data, we transform the observed

component vector $\boldsymbol{x}_t$ into a new component vector $\boldsymbol{y}_t$ by

$$\boldsymbol{y}_t = \widehat{W}_{(k)}\,\boldsymbol{x}_t - \widehat{\boldsymbol{\mu}}_{(k)}, \quad t = 1, 2, ..., n; \quad (k = 1, 2, ..., c) \tag{6.14}$$

If $\widehat{W}_{(k)}$ and $\widehat{\boldsymbol{\mu}}_{(k)}$ are the estimates for ${A_{(k)}}^{-1}$ and ${A_{(k)}}^{-1}\boldsymbol{b}_{(k)}$, respectively, then a class of data points

$$\mathscr{D}_{(k)} = \left\{\boldsymbol{x}_t \in \mathscr{D} : \ \phi\left(\boldsymbol{x}_t; \widehat{W}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}\right) \geq \alpha_k\right\} \tag{6.15}$$

are classified into source class (k). Note that weight of each unclassified data points almost close to zero. In ordered to separate the recovered signals, we chose the value of α_k by

$$\alpha_k = (1-\gamma) \min_{\boldsymbol{x}_t \in \mathscr{D}} \phi\left(\boldsymbol{x}_t; \widehat{W}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}\right) + \gamma \max_{\boldsymbol{x}_t \in \mathscr{D}} \phi\left(\boldsymbol{x}_t; \widehat{W}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}\right), \tag{6.16}$$

with heuristically $0.01 \leq \gamma \leq 0.05$. Also one can chose α_k based on percentile of the densities.

The cumulative weighting plot represents the weights of both the classified and unclassified data points. Thus, the classification procedure can be continued until the remaining unclassified data points are transferred to classified data points by monitoring the cumulative weighting plot and the value of the termination index (TI) $= \frac{|J|}{n} \leq 1$ after each step, where $|J|$ is the number of elements in the set

$$J = \left\{t : \sum_{k=1}^{c} \phi\left(\boldsymbol{x}_t; \widehat{W}_{(k)}, \widehat{\boldsymbol{\mu}}_{(k)}\right) \geq \alpha\right\} \tag{6.17}$$

where $\alpha = \sum_{k=1}^{c} \alpha_k$.

The value TI $= a \leq 1$ suggests 100a% observed data points are classified into distinct source classes and the rest 100(1-a)% data points remain unclassified as outliers. The classification procedure is terminated when the value of the termination index TI exceeds a certain value. In our simulation study, we terminated the procedure when TI exceeds 0.90. In the following section, we introduce an adaptive selection procedure for the tuning parameter β.

6.3.1 Selection Procedure for the Tuning Parameter β

The tuning parameter β is a a key to the performance of the proposed method. Minami and Eguchi (2003) proposed an adaptive selection procedure for β, and their procedure is basically followed herein. In order to find an appropriate β, we evaluate the estimates using various values of β. There are four aspects involved in evaluating the estimates:

1. Measure for evaluation
2. Generalization scheme
3. Scaling of estimates for the recovering matrix
4. How to decide β

Measure for Evaluation

We would like to recover a hidden class from the entire data space using the minimum β-divergence method based on the initial condition of the shifting parameter $\boldsymbol{\mu}$ considering other classes as outliers. Therefore, the measure used for evaluation should give a good evaluation when a hidden class is recovered, but should not have an excessive penalty for the existence of outliers. The K-L divergence between the distribution of $\boldsymbol{x}$ and the pseudo model (6.4), or equivalently, the pseudo log-likelihood does not satisfy this condition. We would like to use β-divergence with a fixed value β_0 of β as a measure for evaluating estimators for hidden class separation and so define the measure used for evaluation of the minimum β-divergence estimators as $\widehat{W}_\beta$ and $\widehat{\boldsymbol{\mu}}_\beta$ as:

$$D_{\beta_0}(\beta) = \mathrm{E}\left[D_{\beta_0}\left(\tilde{r}, r_0(\cdot, \hat{\Lambda}_{\beta,\beta_0}\widehat{W}_\beta, \hat{\boldsymbol{\mu}}_{\beta,\beta_0})\right)\right] \tag{6.18}$$

where $\hat{\Lambda}_{\beta,\beta_0}$ and $\hat{\boldsymbol{\mu}}_{\beta,\beta_0}$ are explained later herein, $\tilde{r}$ is the empirical distribution of $\boldsymbol{x}$, r_0 is defined in (6.4), the notation, E, denote the expectation with respect to the underlying distribution of the data.

$$D_{\beta_0}\left(\tilde{r}, r_0(\cdot, \hat{\Lambda}_{\beta,\beta_0}\widehat{W}_\beta, \hat{\boldsymbol{\mu}}_{\beta,\beta_0})\right) = \text{Const.} - \frac{1}{n}\sum_{\boldsymbol{x}\in\mathcal{D}} l_{\beta_0}(\boldsymbol{x},\ \hat{\Lambda}_{\beta,\beta_0}\widehat{W}_\beta,\ \hat{\boldsymbol{\mu}}_{\beta,\beta_0});$$

Here, $l_{\beta_0}(\boldsymbol{x},\ \hat{\Lambda}_{\beta,\beta_0}\widehat{W}_\beta,\ \hat{\boldsymbol{\mu}}_{\beta,\beta_0})$ is defined as (6.6) with $\beta = \beta_0$, $W = \left(\hat{\Lambda}_{\beta,\beta_0}\widehat{W}_\beta\right)$, and $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}_{\beta,\beta_0}$.

Generalization Scheme

The measure $D_{\beta_0}(\beta)$ is a measure of the generalization performance of an estimator, which is related to the prediction capability for independent test data. If we use the same dataset to evaluate $D_{\beta_0}(\beta)$ as that used to estimate a recovering matrix, then $D_{\beta_0}(\beta)$ will be underestimated. In a data-rich situation, the best approach is to divide the dataset into a small number of subsets, and use one of these subsets for estimation and another for evaluation. In other situations, a simple and widely used method of sample reuse is the **K-fold cross-validation (CV)** method (Hastie et al., 2001). The K-fold CV method uses part of the

available data to find the estimate and a different part of the data to test the estimate. For the current problem, we employ the K-fold CV method as a generalization scheme.
We split the data into K approximately equal-sized and similarly distributed sections. For the k th section, we find the estimate using the other $K-1$ parts of the data, and calculate the β_0-divergence for the k th section of the data. We then combine the calculated β_0-divergence values to obtain the CV estimate.

Scaling of Estimates for the Recovering Matrix

For the blind source separation problem, the scaling and shifting of the recovered signals as well as the scaling of a recovering matrix, are arbitrary because scaling and shifting do not affect independence. However, β-divergences differ with the scaling. That is, for any $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$

$$D_{\beta_0}(\tilde{r}, r_0(\cdot, W, \boldsymbol{\mu}_1)) \neq D_{\beta_0}(\tilde{r}, r_0(\cdot, \Lambda W, \boldsymbol{\mu}_2))$$

in general, where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m)$ unless $\Lambda = I_m$.
The scaling and shifting condition for the minimum β-divergence method differs with the value of β. In order to properly evaluate the minimum β-divergence estimates, we need to rescale and shift the estimates under a common condition. For this purpose, we use the scaling and shifting condition for the minimum β-divergence estimator with $\beta = \beta_0$. That is, we rescale the minimum β-divergence estimate $\widehat{W}_\beta$ with β by the diagonal matrix $\hat{\Lambda}_{\beta,\beta_0}$ and use the shift parameter $\hat{\boldsymbol{\mu}}_{\beta,\beta_0}$ for evaluation, where $\hat{\Lambda}_{\beta,\beta_0}$ and $\hat{\boldsymbol{\mu}}_{\beta,\beta_0}$ minimize $\widehat{D}_{\beta_0}\left(\tilde{r}, r_0(\cdot, \Lambda\widehat{W}_\beta, \boldsymbol{\mu})\right)$ among diagonal matrix Λ and vector $\boldsymbol{\mu}$.
Table 1 summarizes the procedure used to find the K-fold CV estimate $\widehat{D}_{\beta_0}(\beta)$.

6.3.2 How to Decide β

As a measure for the variation of $\widehat{P}_{\beta_0}(\beta)$, we compute

$$\mathrm{SD}_{\beta_0}(\beta) = \text{the standard error of } \frac{1}{|\mathscr{P}(k)|}\mathrm{CV}_{(k)},$$

where $|\mathscr{P}(k)|$ denotes the number of elements in the k-th part of data $\mathscr{P}(k)$. Plots of $\widehat{D}_{\beta_0}(\beta)$ for β with the auxiliary boundary curves $\widehat{D}_{\beta_0}(\beta) \pm \mathrm{SD}_{\beta_0}(\beta)$ will help to judge an optimum β. We denote this optimum β by β_{opt}. Often we have to employ the upper auxiliary boundary curve (UABC) with the curve of $\widehat{D}_{\beta_0}(\beta)$ in order to choose the β_{opt}. We choose the smallest β as the β_{opt} whose evaluated value $\widehat{D}_{\beta_0}(\beta_{\mathrm{opt}})$ is not larger than the value of UABC that

Table 6.1: K-fold cross-validation procedure

Split the data set $\mathscr{D}$ into K parts; $\mathscr{P}(1), \cdots, \mathscr{P}(K)$.

Let $\mathscr{P}^{-k} = \{\boldsymbol{x} | \boldsymbol{x} \notin \mathscr{P}(k)\}$.

For $k = 1, \cdots, K$

- Estimate W and $\boldsymbol{\mu}$ by maximizing $L_\beta(W, \boldsymbol{\mu})$ using $\mathscr{P}^{-k}$,

$$(\widehat{W}_\beta,\ \widehat{\boldsymbol{\mu}}) = \underset{W,\ \boldsymbol{\mu}}{\operatorname{argmax}} \sum_{\boldsymbol{x} \in \mathscr{P}^{-k}} l_\beta(\boldsymbol{x}; W, \boldsymbol{\mu}).$$

- Estimate Λ_{β,β_0} and $\boldsymbol{\mu}_{\beta,\beta_0}$ by maximizing $L_{\beta_0}(\Lambda \widehat{W}_\beta, \boldsymbol{\mu})$ using $\mathscr{P}^{-k}$,

$$(\widehat{\Lambda}_{\beta,\beta_0},\ \widehat{\boldsymbol{\mu}}_{\beta,\beta_0}) = \underset{\Lambda,\ \boldsymbol{\mu}}{\operatorname{argmax}} \sum_{\boldsymbol{x} \in \mathscr{P}^{-k}} l_{\beta_0}(\boldsymbol{x}; \Lambda \widehat{W}_\beta, \boldsymbol{\mu}).$$

- Compute $\mathrm{CV}_{(k)}$ using $\mathscr{P}(k)$,

$$\mathrm{CV}_{(k)} = - \sum_{\boldsymbol{x} \in \mathscr{P}(k)} l_{\beta_0}(\boldsymbol{x},\ \widehat{\Lambda}_{\beta,\beta_0} \widehat{W}_\beta,\ \widehat{\boldsymbol{\mu}}_{\beta,\beta_0})$$

End

Then, $\widehat{D}_{\beta_0}(\beta) = \dfrac{1}{n} \sum_{k=1}^{K} \mathrm{CV}_{(k)}$.

corresponds to the smallest value of $\widehat{D}_{\beta_0}(\beta)$. However, there is no theoretical justification for this rule, which is known as the one-standard error rule (Hastie et al., 2001). If the curve of $\widehat{D}_{\beta_0}(\beta)$ is flat for a wide range of β, then there might be only one class with no outlier and β_{opt}=0. When there are more than one data class or outliers exist in the entire data space, typical shapes of curves of $\widehat{D}_{\beta_0}(\beta)$ that enables us to chose an appropriate value β are elbow and dipper shapes. So, if the curve does not have these shapes, we increase the value of β_0. If these shapes do not appear for any β_0, then there might be only one class with no outlier and β_{opt}=0, (Minami and Eguchi, 2003).

6.4 Numerical Examples

We investigated the performance of the proposed procedure for recovering the hidden classes of mixture ICA models using both synthetic and real data sets. For simulation, we generated the following data sets by formula (6.1) using different mixing matrices A_k and bias vectors $\boldsymbol{b}_k$.

Dataset 1 : Two-dimensional, two-class mixtures (figure 6.1(a)) generated with uniform (sub-Gaussian) independent sources. 200 samples were drawn from each class to make 400 samples in total.

Dataset 2 : Two-dimensional, two-class mixtures (figure 6.2(a)) generated with Laplace (super-Gaussian) independent sources. 200 samples were drawn from each class to make 400 samples in total.

Dataset 3 : Five-dimensional, two-class mixture generated with uniform independent sources. Plots of two observed signals are shown in figure 6.3(top) using the combination rule. 200 samples were generated from each class to make 400 samples in total.

Dataset 4 : Two-dimensional, six-class mixtures generated with uniform independent sources. 200 samples were drawn from each class. Also we added two-dimensional 20 random vectors (*) from a Gaussian class and arrange them from 1001 to 1020 sample points to make 1220 sample points in total. Figure 6.5(a) represent the observed values.

Dataset 5 : Two-dimensional, two-class mixtures shown in figures 6.7(d-e). One class is the mixture of sinusoid signal (figure 6.7(a)) and Gaussian noise (figure 6.7(c), the first half), and the other class is the mixture of saw-tooh signal (figure 6.7(b)) and Gaussian noise (figure 6.7(c), the last half).

Dataset 6 : Two-dimensional, two-class mixtures of voices and music noises (figures 6.8(c-d)). The sample size is 100000 in total, the first and third quarter of samples are the mixture of voice of person 1 and background music noise, and the second and last quarter are the mixture of voice of person 2 and background music noise.

In the following simulation study, we used $p_i(z) = \exp(-z^4/4)$ for sub-Gaussian signals and $p_i(z) = 1/\cosh(z)$ for super-Gaussian signals for estimation. For convenience of presentation, samples in dataset 1 to 5 were ordered by class. However, we did not use any information on sample order in estimation so that the estimation results must be the same even when samples were randomly ordered. We used a quasi-Newton method with BFGS update for the minimum β-divergence method and other optimization problems.

6.4.1 Simulation With Randomly Generated Synthetic Data

Datasets 1, 2 and 3 are randomly generated synthetic datasets. There are two hidden classes in each of these datasets. Figures 1, 2, 3 and 4 depict observed signals and estimation results for these datasets. In the plots of observed signals and recovered signals both, one class is represented by the symbol "." and the other one by the symbol "o". For selection of β, we computed $\widehat{D}_{\beta_0}(\beta)$ for β varying from 0 to 1 by 0.1 with $\beta_0 = 0.3, 0.6$ and 0.9 using the ten-fold CV algorithm given in table 1. In the plots of $\widehat{D}_{\beta_0}(\beta)$, asterisks (*) are $\widehat{D}_{\beta_0}(\beta)$ and the smallest value is indicated by a circle outside the asterisk. Dotted lines are $\widehat{D}_{\beta_0}(\beta) \pm SD_{\beta_0}(\beta)$.

For dataset 1(uniform independent source), we used $\beta_0 = 0.3$ for selection of β at step 1 because the plot of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3$ (figure 6.1(d)) shows an elbow shape. By the 'one-standard error rule', we chose $\beta = 0.3$. $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.6$ and 0.9 (figures 6.1(e-f)), had the same property with $\beta_0 = 0.3$ and these also suggested $\beta = 0.3$. At step 2, plot of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3$ (figure 6.1(g)) is flat for small β and have a sudden increase at certain points indicating it cannot be used for selection of β. Therefore, we used $\beta_0 = 0.6$ (figure 6.1(h)), for selection of β with the same reason as step 1 and chose $\beta = 0.6$. $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.9$ (figure 6.1(i)) had the same shape as that with $\beta_0 = 0.6$ and it also suggested $\beta = 0.6$. Figures 6.1(b-c) show recovered signals by the estimate $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ obtained at step 1 and $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$ obtained at step 2, respectively. We observe that one hidden class is properly recovered by $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ and the other one is recovered by $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$. Figures 6.1(j-k) show the weight of each data point corresponding to the estimates $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ and $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$, respectively. One can see that at each step of estimation of W and $\boldsymbol{\mu}$, one class of data were used and the other class of data totally were ignored by the weight function (6.12). *The value of the termination index (TI) was 0.99 when the sequential recovering procedure was terminated.* The arrows in figure 6.1(a) are the estimated mixing matrices $\widehat{A}_k$ and the bias vectors $\widehat{\boldsymbol{b}}_k$ found by the algorithm and these parameters matched the parameters which were used to generate the data for each class. The arrows in figures 6.1(b-c) represent the center of the recovered classes.

For dataset 2(Laplace independent source), we used $\beta_0 = 0.6$ for selection of β at step 1 because the plot of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.6$ (figure 6.2(e)) shows an elbow shape, while that with

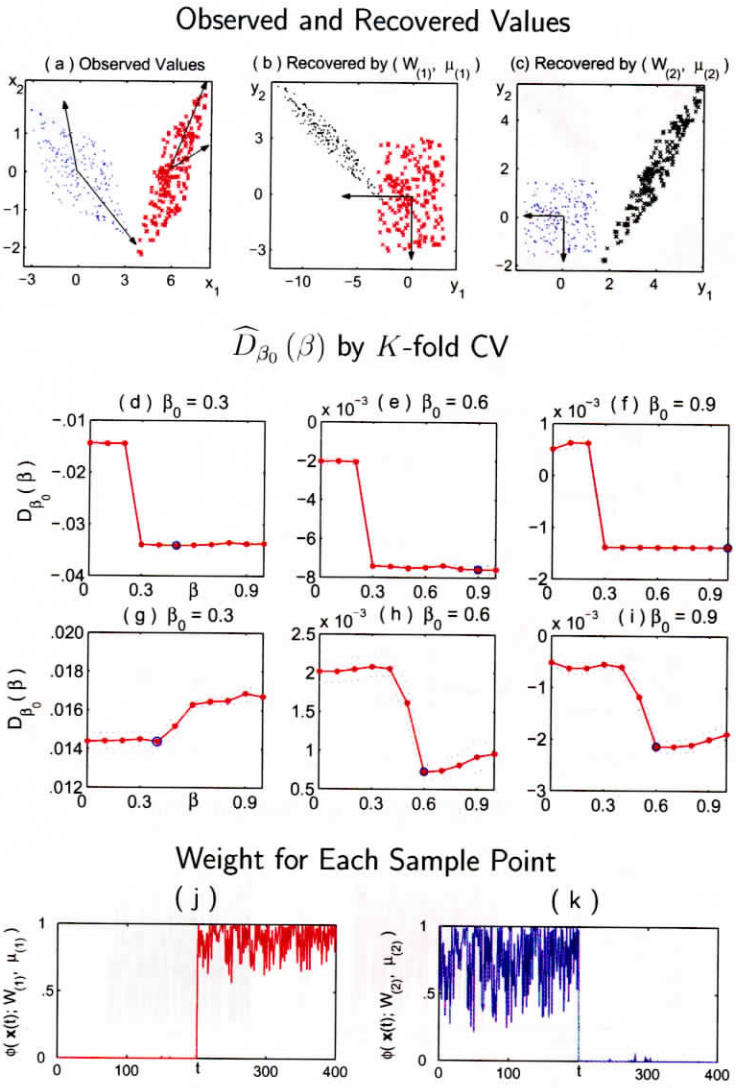

Figure 6.1: For dataset 1, (a) Observed values. (b-c) Recovered values by $(\widehat{W}_{(1)}\ ,\ \widehat{\boldsymbol{\mu}}_{(1)})$ and $(\widehat{W}_{(1)}\ ,\ \widehat{\boldsymbol{\mu}}_{(1)})$, respectively. (d-f, g-i) $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3, 0.6$ and 0.9 at step 1 and 2, respectively. (j-k) Weight for each sample point at step 1 and 2, respectively.

$\beta_0 = 0.3$ (figure 6.2(d)) is flat for small β and have a sudden increase around 0.4 indicating it cannot be used for selection of β. By the 'one-standard error rule', we chose $\beta = 0.5$ from $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.6$. $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.9$ had the same shape as that with $\beta_0 = 0.5$ and it also suggested $\beta = 0.5$. At step 2, we again used $\beta_0 = 0.6$ (or $\beta = 0.9$) for selection of β with the same reason (figures 6.2(g-i)), and chose $\beta = 0.6$. Figures 6.2(b-c) show recovered signals by the estimate $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ obtained at step 1 and $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$ obtained at step 2, respectively. We see that one hidden class is recovered properly by the estimate $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$

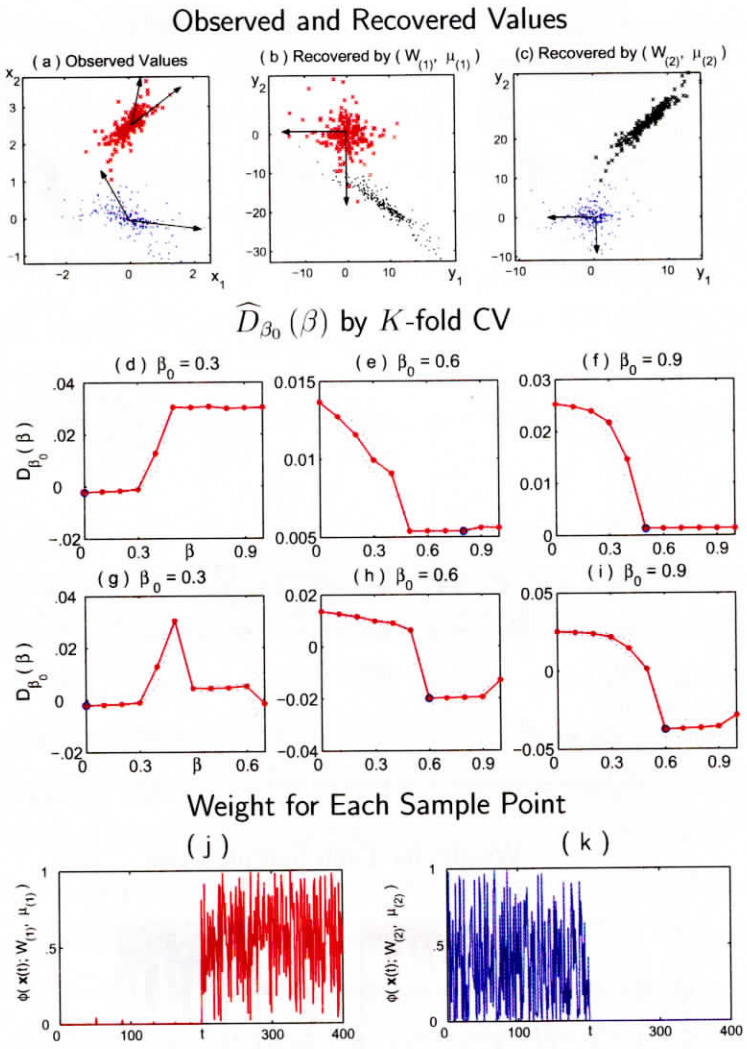

Figure 6.2: For dataset 2, (a) Observed values. (b-c) Recovered values by $(\widehat{W}_{(1)},\ \widehat{\boldsymbol{\mu}}_{(1)})$ and $(\widehat{W}_{(1)},\ \widehat{\boldsymbol{\mu}}_{(1)})$, respectively. (d-f, g-i) $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3, 0.6$ and 0.9 at step 1 and 2, respectively. (j-k) Weight for each sample point at step 1 and 2, respectively.

and the other is recovered by the estimate $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$. Figures 6.2(j-k) displays the weight of each data point corresponding to the estimates $\left(W_{(1)}, \boldsymbol{\mu}_{(1)}\right)$ and $\left(W_{(2)}, \boldsymbol{\mu}_{(2)}\right)$, respectively. Again, at each step for estimation of W and $\boldsymbol{\mu}$, one class of data were used and the other class of data were totally ignored by the weight function (6.12). *The value of TI was 0.92 when the sequential recovering procedure was terminated.*

To investigate the performance of the proposed procedure for high-dimensional data, we

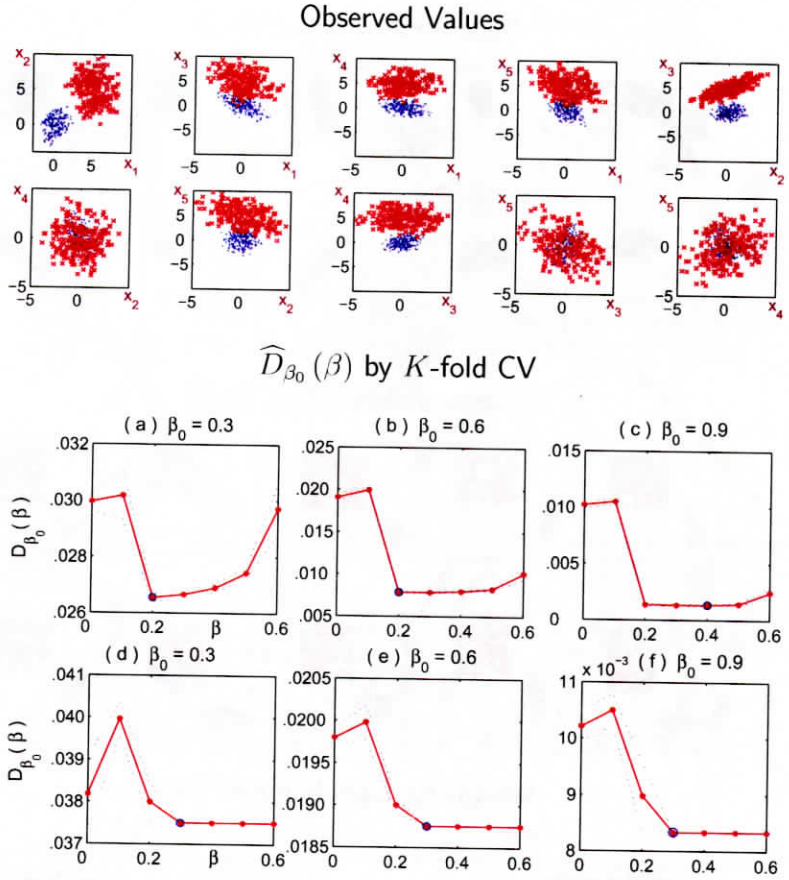

Figure 6.3: For dataset 3, (Top) Observed values. (a-c, d-f) $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3, 0.6$ and 0.9 at step 1 and 2, respectively.

analyzed the five-dimensional sub-Gaussian(uniform) data. Dataset 3 consist of two classes. With projection of observed data onto two-dimensional coordinates, two classes are overlapped as shown in figure 3(top). For estimation of recovering matrix at step 1, we chose $\beta = 0.2$, because all of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 0.3, 0.6$ and 0.9 (figure 6.3(a-c)) have an elbow shape and $\beta = 0.2$ is suggested by all of them. At step 2, we chose $\beta = 0.2$ as in the previous step using figures 6.3(d-f). Figure 6.4(Top) and 6.4(Middle) show recovered values by the estimate $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ obtained at step 1 and $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$ obtained at step 2, respectively. It is observed that one hidden class is recovered properly by the estimates $\left(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)}\right)$ and the other is recovered by the estimates $\left(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)}\right)$. Figures 6.4(a-b) show the weight for each data point at step 1 and step 2, respectively. Same as the previous examples, at each step one class of data were used for estimation and the other class of data were totally ignored by the weight function. *The value of TI was 0.99 when the sequential recovering procedure*

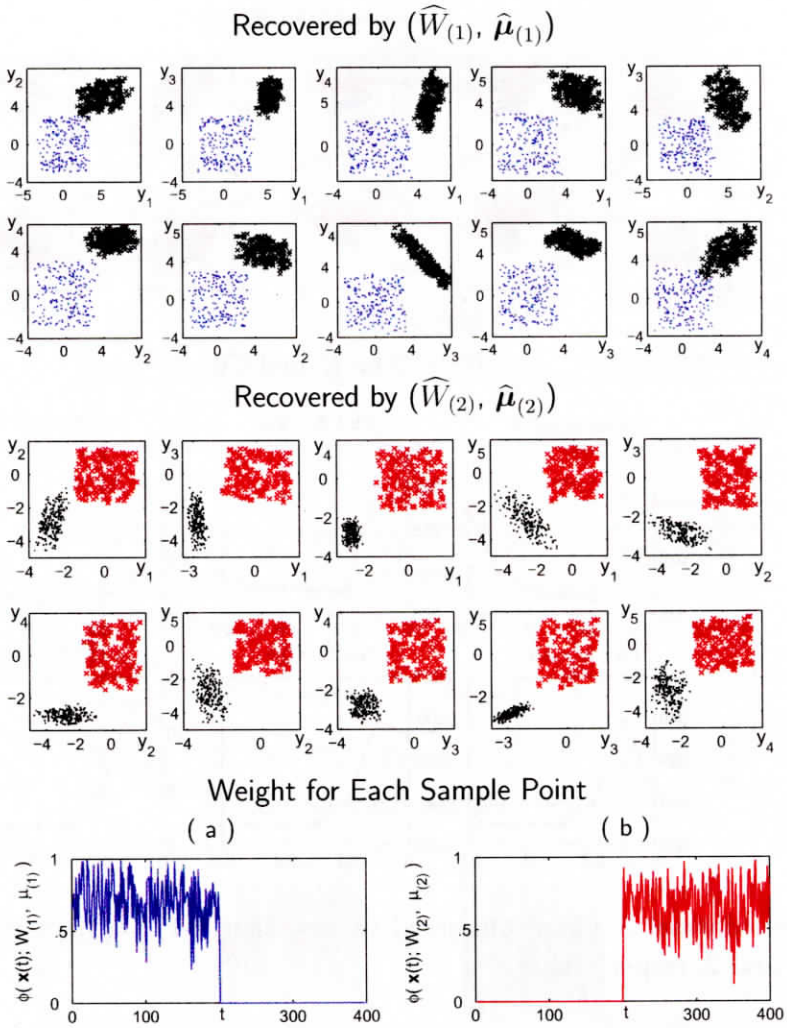

Figure 6.4: For dataset 3, (Top) Recovered values by $(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)})$. (Middle) Recovered values by $(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)})$. (Bottom) (a-b) Weight for each sample point at step 1 and 2, respectively.

was terminated.

To demonstrate the validity of the proposed methods for mixtures of several classes, we considered two-dimensional, seven class mixture of synthetic data shown in figure 6.5(a). Original independent sources are uniform random numbers in six classes and the rest one class consist of two-dimensional 20 Gaussian random numbers (*). For this data, we used $\beta_0 = 1.2$ for selection of the values of the tuning parameter β. Figures 6.5(j-p) depicts the values of $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 1.2$ for steps 1 to 7, respectively. We chose $\beta = 0.8$ for steps 1 to 6

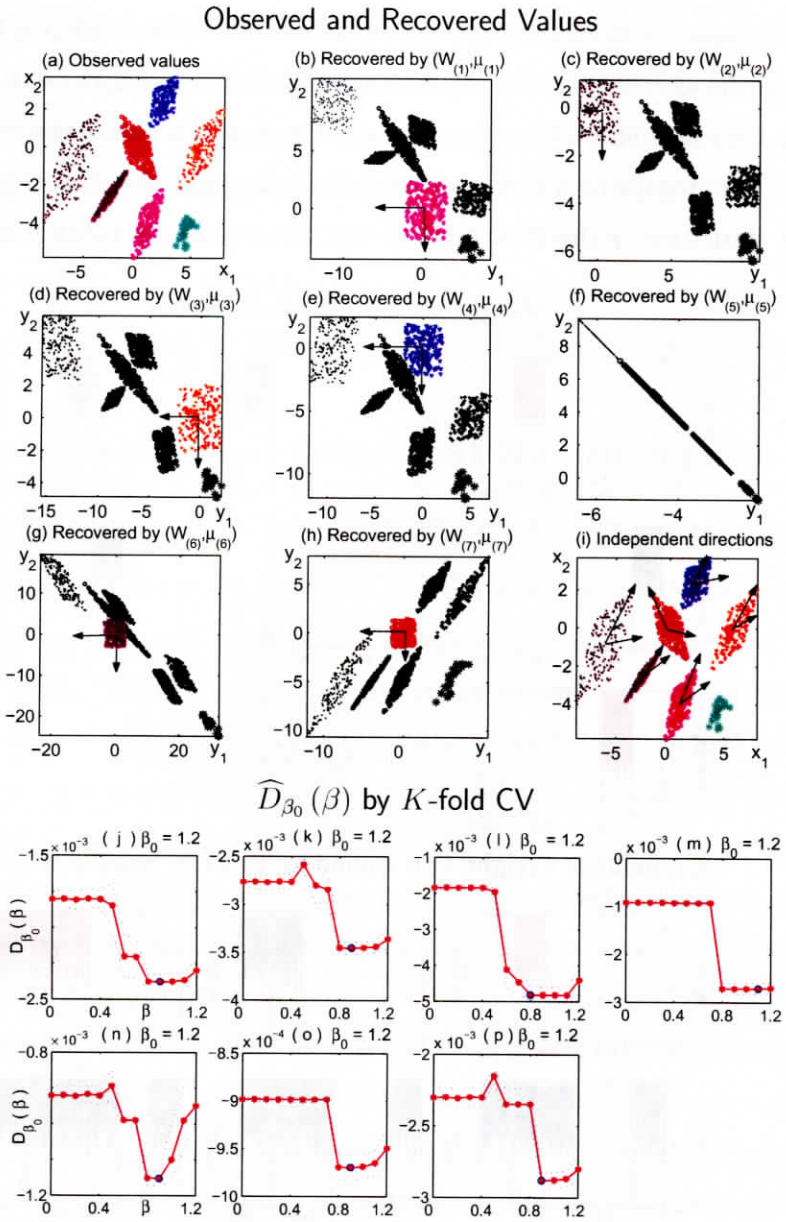

Figure 6.5: For dataset 4, (a) Observed values. (b-h) Recovered values by $(\widehat{W}_{(i)}, \widehat{\boldsymbol{\mu}}_{(i)})$ for $i = 1, \cdots, 7$, respectively. (i) Independent direction. (j-p) $\widehat{D}_{\beta_0}(\beta)$ with $\beta_0 = 1.2$ at step 1 to 7, respectively.

and $\beta = 0.9$ for step 7 based on the 'one-standard error rule'. Figures 6.5(b-h) show the plots of recovered classes by the estimated recovering matrices and shifting vectors at step 1 to 7, respectively. Figures 6.5(b-e, g-h) show that each estimated recovering matrix at step 1, 2, 3, 4, 6 and 7 recovers independent sources for one of sub-Gaussian classes, while figure 6.5(f)

shows that estimated recovering matrix at step 4 does not recover independent sources for any one class, since the shifting vector was initialized to the Gaussian class (figure 6.6(e)) at this step and ICA cannot separate Gaussian signals. Figure 6.5(i) shows the estimated structures of dataset 4. The origins of arrows are located at $\widehat{\boldsymbol{\mu}}_{(i)}$ and the directions are the columns of $\widehat{W}_{(i)}^{-1}$. The structure of the ICA mixture data was properly estimated. Figures 6.6(a-g)

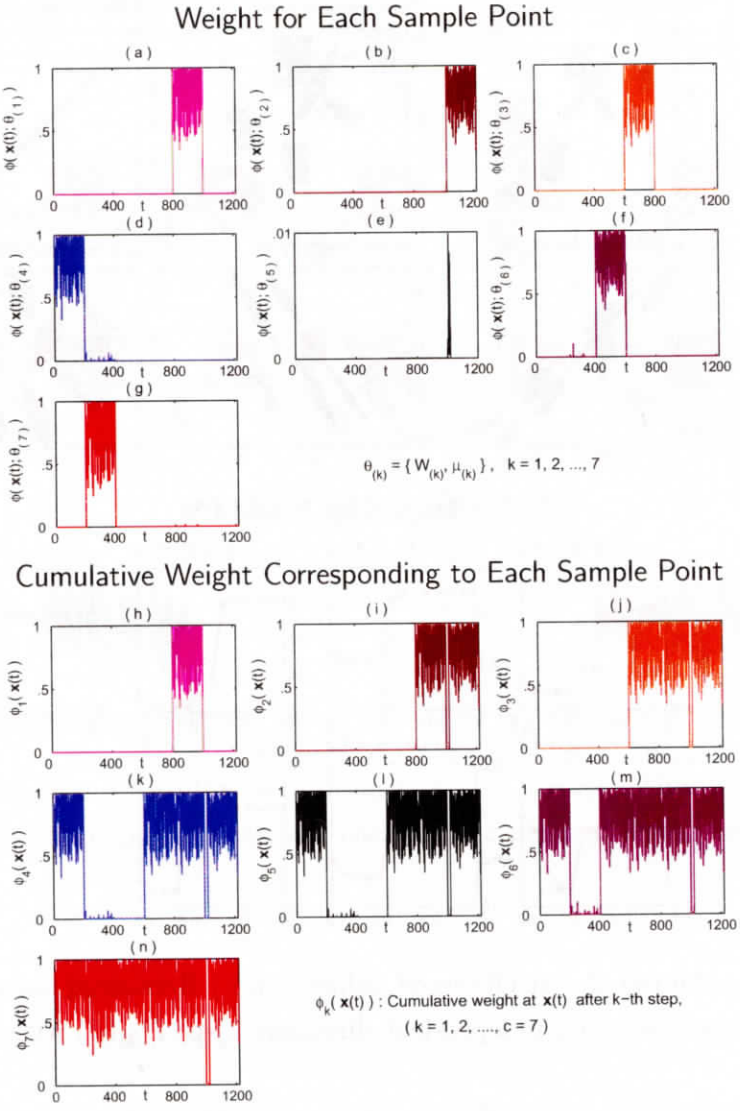

Figure 6.6: For dataset 4, (a-g) Weight for each sample point at step 1 to 7, respectively. (h-n) Cumulative weight at step 1 to 7, respectively.

show the weight for each data point corresponding to the estimates $\left(W_{(i)}, \boldsymbol{\mu}_{(i)}\right), i = 1, ..., 7$,

and figures 6.6(h-n) show the cumulative weight after each step from 1 to 7, respectively. Figure 6.6(n) shows that cumulative weights corresponding to Gaussian data points are negligible in a comparison of the cumulative weights corresponding to non-Gaussian data points. Therefore, data belongs to the Gaussian class were considered as outliers in each step by the proposed algorithm. Same as the previous examples, at each step one class of data were used for estimation and the other classes of data were totally ignored by the weight function. *The value of TI was 0.97 when the sequential recovering procedure was terminated.*

6.4.2 Simulation With Artificial and Natural Signals

Dataset 5 and 6 were generated with artificial and natural signals, respectively. Both datasets were used to investigate the performance of the proposed procedure for automatic context switching in blind source separation problem, which was first introduced by Lee, Lewicki and Sejnowski (2000). There are two hidden classes in dataet 5. One class is a mixture of sinusoid signals (figure 6.7(a)) and Gaussian noises (figure 6.7(c), the first half) and the other class is a mixture of saw-tooth signals (figure 6.7(b)) and Gaussian noises (figure 6.7(c)), the last half). Although there are three different source signals, at any given moment only 2 source signals were linearly mixed and two mixed signals were observed (figures 6.7(g-h)). We chose β=0.45 and 0.5 with the same procedure as in the previous example by K-fold CV for steps 1 and 2, respectively. Figures 6.7(f-g) and 6.7(h-i) show recovered signals by the estimated recovering matrices at step 1 and step 2, respectively. Sinusoid signals were recovered by the estimated recovering matrix at step 1 and saw-tooth signals were recovered by the estimated recovering matrix at step 2. *The value of TI was 0.92 when the sequential recovering procedure was terminated.* For dataset 6, let us imagine a situation that two students were talking to each other while they are listening to music in the background. Two microphones were placed somewhere in the room to record the conversation. The conversation alternates so that person number 1 talks while person number 2 listens, then person number 1 listens to person number 2 and so on. In this case, the voice of person number 1 overlaps with the background music signal by a mixing matrix A_1 and bias vector $\boldsymbol{b}_1$, whereas voice of person number 2 overlaps with the background music signal by a mixing matrix A_2 and bias vector $\boldsymbol{b}_2$. Although there are three different source signals, at any given moment there are only two source signals mixed in the observed data. The original signals and observed mixed signals are shown in figures 6.8(a-b) and 6.8(c-d), respectively. The

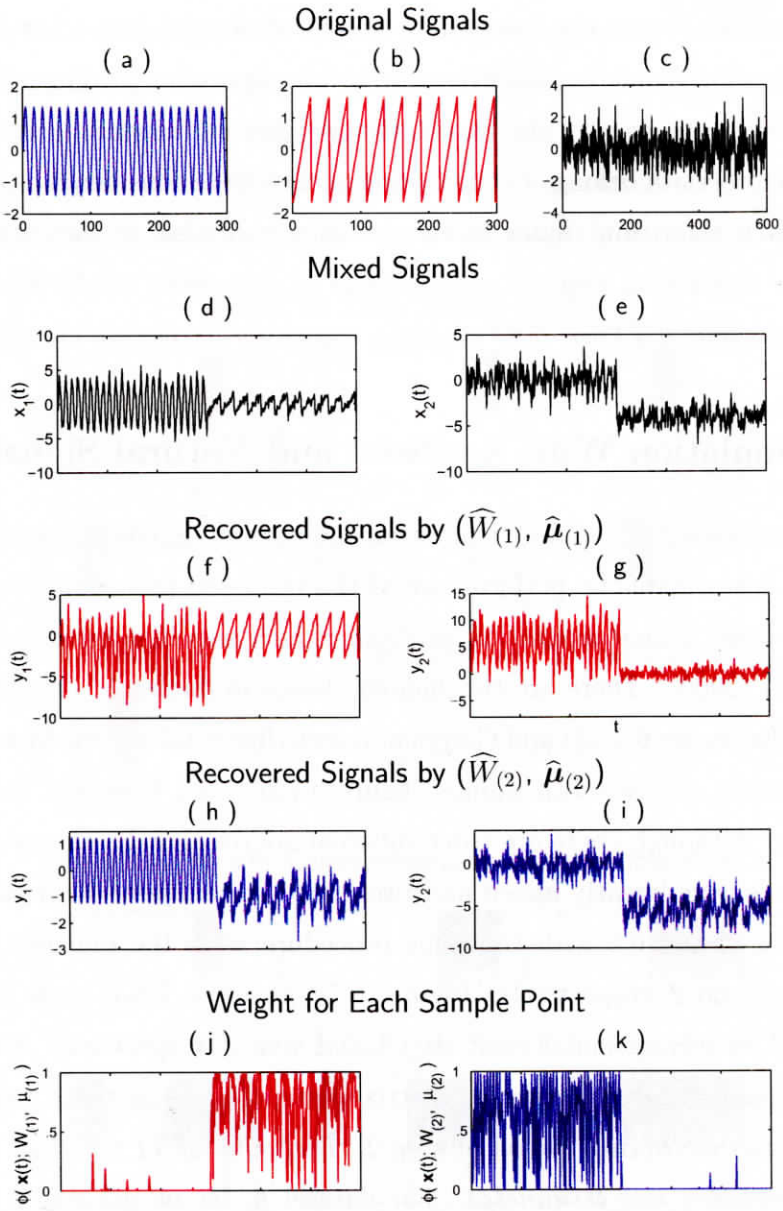

Figure 6.7: For dataset 5, (a-c) Original signals, where (a) sinusoid signals, (b) saw-tooth signals and (c) Gaussian noises. (d-e) Mixed signals. (f-g) Recovered signals by $(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)})$. (h-i) Recovered signals by $(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)})$. (j-k) Weight for each sample point at step 1 and 2, respectively.

scatter plot of mixed signals are shown in figure 6.8(e). In the scatter plot, we see that some data points are overlapped between two clusters. Figures 6.8(f-g) represent the recovered voice conversion and background music noise, respectively, by the proposed method, in which we change the scale of the recovered signals to compare with the original voice conversation

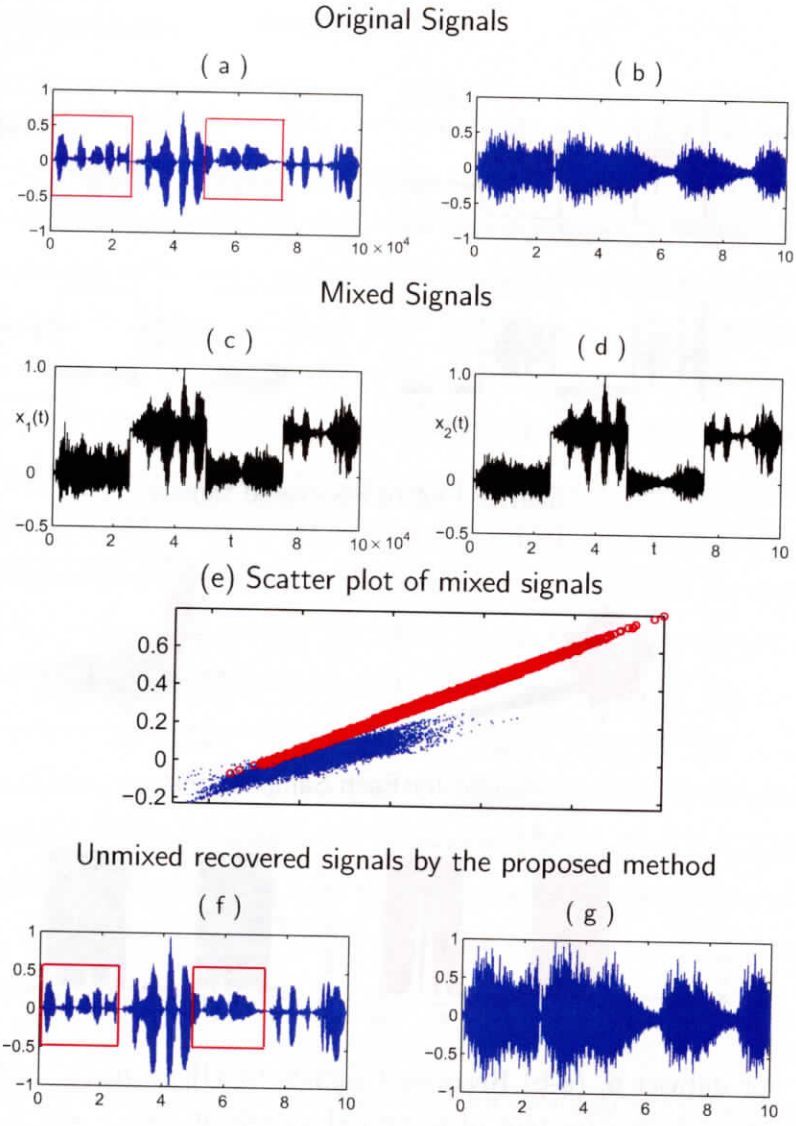

Figure 6.8: For dataset 6, (a) Original signals from a voice conversation, where the rectangular boxes represent the voice of person 1 and the rest are the voice of person 2. (b) Original music signal. (c-d) Mixed signals obtained by the mixture of original voice signals and music signals. (e) Scatter plot of mixed signals. (f-g) Unmixed recovered signals by the proposed method.

and music noise explicitly. Comparing recovered conversion with the original one, we can say that performance of the proposed method is good in our current context. We obtained the above results with two steps. At step 1, we used $\beta = 0.45$, and at step 2, $\beta = 0.45$ with the same procedure as in the previous example by K-fold CV. Figure 6.9(a-b) show recovered signals by the estimate obtained at step 1 and figure 6.9(c-d) show those by the estimate

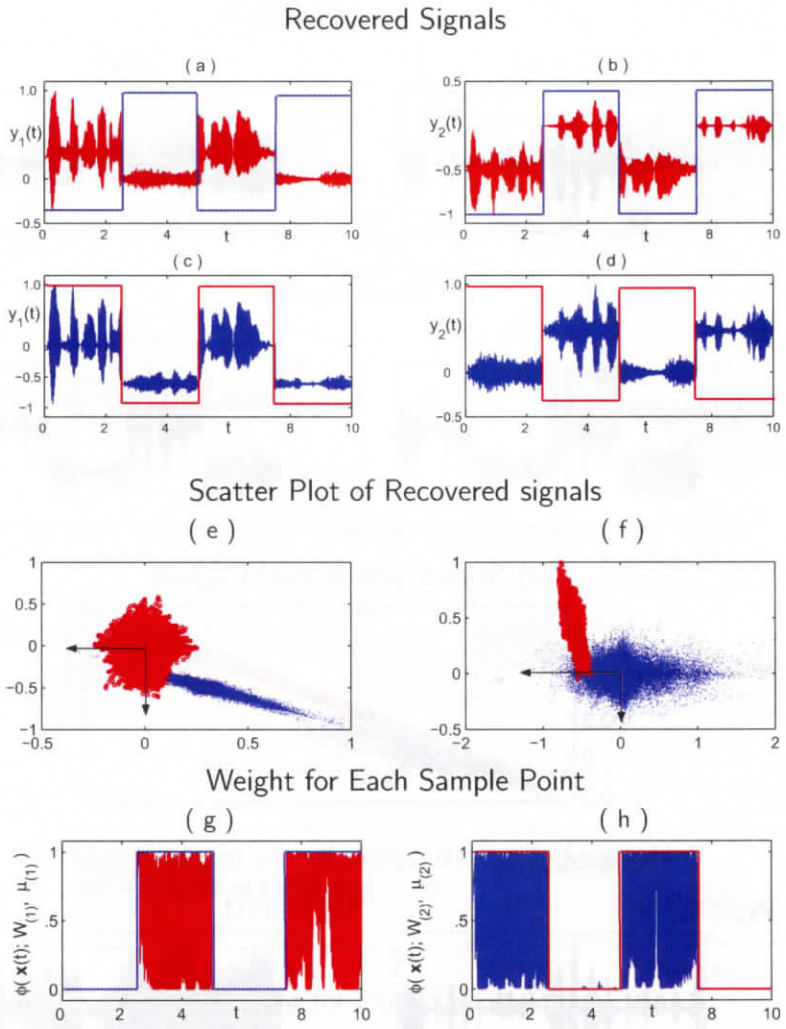

Figure 6.9: For dataset 6, (a-b) Recovered signals by $(\widehat{W}_{(1)}, \widehat{\boldsymbol{\mu}}_{(1)})$. (c-d) Recovered signal by $(\widehat{W}_{(2)}, \widehat{\boldsymbol{\mu}}_{(2)})$. (e-f) Scatter plot of recovered signals at step 1 and 2, respectively. (g-h) Weight for each sample at step 1 and 2, respectively.

obtained at step 2. Figures 6.9(e-f) represent the scatter plot of recovered signals at step 1 and 2, respectively. Figures 6.9(g-h) show the weights of sample points for estimation at step 1 and 2, respectively. The voice of person 1 was recovered by the estimate obtained at step 1. The weights for mixed signals of the voice of person 2 were almost zero for the estimation at step 1. By the estimate obtained at step 2, the voice of person 2 was recovered and the weights for mixed signals of the voice of person 1 were almost zero for the estimation. *The value of TI was 0.91 when the sequential recovering procedure was terminated.*

6.5 Conclusions

The present chapter proposed a one-by-one hidden class separation algorithm based on the minimum β-divergence method for ICA mixture models. The proposed procedure searches the recovering matrix of each class on the basis of the initial conditions of the shifting parameter. If the initial value of the shifting parameter vector $\boldsymbol{\mu}$ belongs to a data class, then the minimum β-divergence estimator finds the estimates of the recovering matrix and shifting parameter for this class. In order to obtain estimates of the recovering matrix and the shifting parameter for other data classes, the initial value of the shifting parameter is changed according to the observed vector having the minimum cumulative weight. Using the proposed method, all hidden classes can be explored sequentially from the entire data space. We suggested a termination index for the proposed method based on the cumulative weight. On the basis of our simulation results, the value of the termination index (TI) should be greater than 0.90 to terminate the classification procedure.

The value of the tuning parameter β is a key to the performance of the proposed method. We used an adaptive selection procedure for β proposed by Minami and Eguchi(2003). The β-divergence $D_{\beta_0}(\cdot)$ with fixed β_0 was used as a measure for the evaluation of the tuning parameter value β. $D_{\beta_0}(\beta)$ for different values of β were estimated by K-fold cross-validation. This procedure is summarized in Table 1.

In our simulation, we used fixed density functions for estimation by the minimum $\beta-$ divergence method. However, it can be modified by the same switching scheme employed by extended infomax algorithm (Lee *et al.*, 1999) between sub- and super-Gaussian distributions.

The main purpose of the proposed method is similar to the conventional ICA mixture models proposed by Lee *et al.* (2000). The procedure proposed by Lee et al. finds the estimates for all mixing matrices and shifting parameters simultaneously, whereas the method proposed herein finds the estimate for each recovering matrix and shifting parameter sequentially. The proposed algorithm always converges after 20 to 60 iterations for the estimation of a recovering matrix, whereas the mixture ICA algorithm of Lee et al. converges within 80 iterations for the simultaneous estimation of all recovering matrices. However, for the recovery

of all hidden classes, the computational cost may be similar for both methods because the proposed method requires cross-validation.

The procedure proposed by Lee *et al.* may be simpler than the method proposed herein when the number of hidden classes, c, is known. However, if the number of hidden classes is unknown or mis-specified, their method fails to find good estimates. When their procedure was applied to data set 4, described in Section 4, with $c = 3, 7$(exact) and 12, their procedure successfully estimated the mixing matrices when $c = 7$, but failed to find good estimates when $c = 3$ or 12.

Unsupervised classification might be one of the most important applications of mixture ICA model. In section 3, we proposed a sequential classification procedure carried out at the same time as the sequential extraction of hidden classes. However, once all estimates of hidden class structures are obtained, one may use the Bayes rule for simultaneous classification of observations. scaling of sources to compute class probability can be obtained by the method described in section 3.3 as $\widehat{\Lambda}_{\beta_0}$ and $\widehat{\mu}_{\beta_0}$, that is, scaling when $\beta_0 = 0$.

When classes are not overlapped so much, the sequential classification methods and the Bayes rule will give similar results. If some classes are overlapped lightly, then the proposed method is able to find the independent directions (e.g. Fig. 6.8(e)). However, the case in which classes are heavily overlapped is still difficult for the proposed method as well as the model-based classification by Lee *et al.* (2000).

Chapter 7

Pending/Future Research Plan

7.0.1 A Short Review on FastICA

Let us shortly review the conventional FastICA (Hyvärinen et al., 2001) focusing on the contrast function. A prewhitened data set is necessary for FistICA algorithm. Let $\boldsymbol{z}$ be a prewhitened vector of $\boldsymbol{x}$. The maximization of non-Gaussianity after whitening the data set is the major routine for FastICA. We focus on the robustness issues in the class of FastICA (Hyvärinen, 1999) procedures. The aim of FastICA algorithm is to find an orthogonal direction $\boldsymbol{w}$ for independent components by maximizing the objective function $\mathrm{E}\{\pm G(y)\}$, that is

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \ \mathrm{E}\{\pm G(y)\} \tag{7.1}$$

where $y = \boldsymbol{w}^T\boldsymbol{z}$ and $\boldsymbol{w}$ is a unit vector in R^n and $G(y)$ is the contrast function. A well known contrast function for robust FastICA is

$$G(y) = -\exp(-y^2/2) \tag{7.2}$$

Fixed-point iteration based on Gradient method for (7.1) is

$$\boldsymbol{w} \longleftarrow \mathrm{E}\{\boldsymbol{z}g(y)\} \tag{7.3}$$

under the constraint $\|\boldsymbol{w}\|^2 = 1$. The basic fixed-point iteration in FastICA based on approximative Newton iteration for (7.1) is

$$\boldsymbol{w} \longleftarrow \mathrm{E}\{\boldsymbol{z}g(y)\} - \mathrm{E}\{g'(y)\}\boldsymbol{w} \tag{7.4}$$

under the constraint $\|\boldsymbol{w}\|^2 = 1$, where $g(y) = G'(y) = y\exp(-y^2/2)$ and $g'(y) = G''(y) = (1-y^2)\exp(-y^2/2)$. The complete FastICA algorithm is given in appendix (A.1.1).

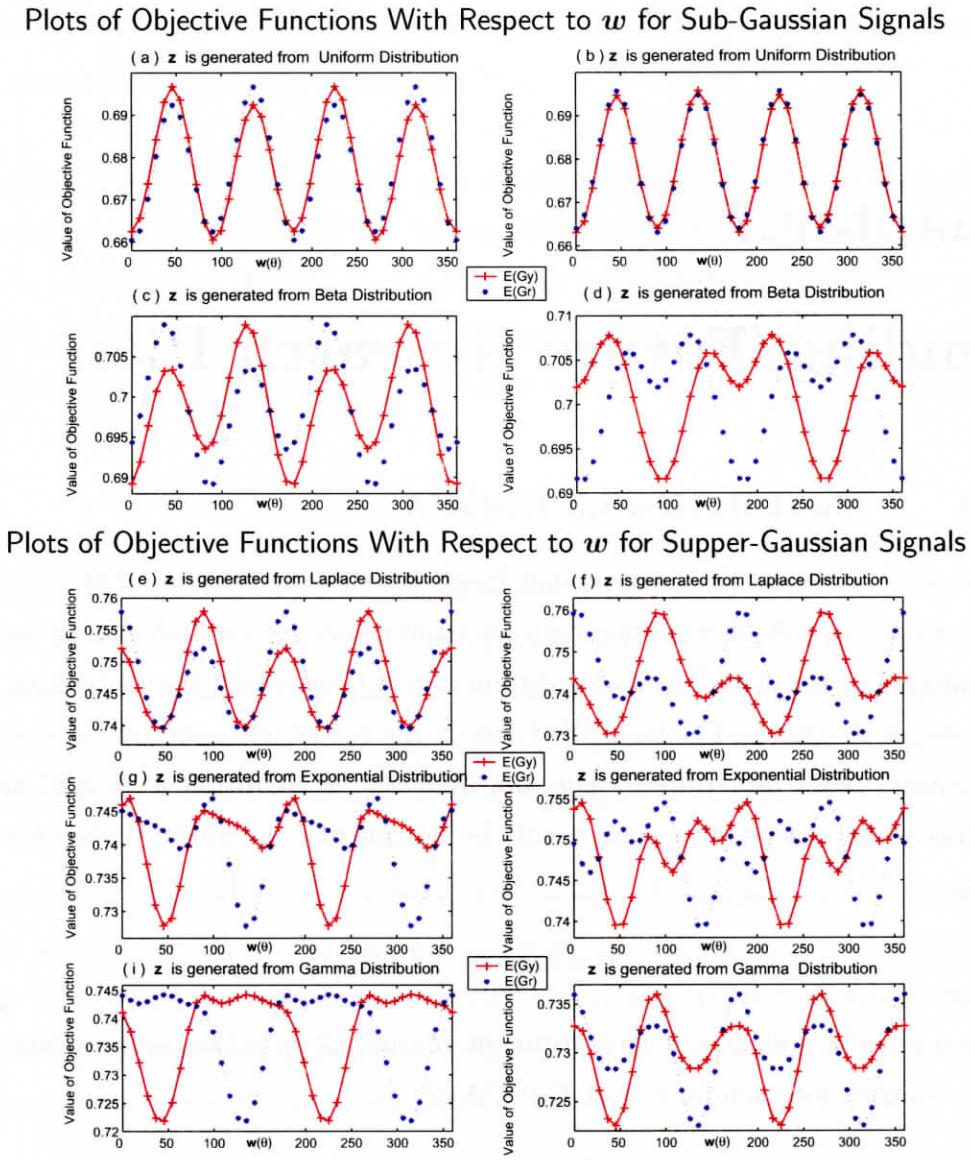

Figure 7.1: (a-d) Plots of objective functions with respect to recovering vector $\boldsymbol{w}$ for sub-Gaussian signals. (e-j) Plots of objective functions with respect to recovering vector $\boldsymbol{w}$ for supper-Gaussian signals.

7.0.2 Dual FastICA

Let us consider an alternative idea to FastICA by orthogonal decomposition of the prewhitened random vector $\boldsymbol{z}$ as

$$\boldsymbol{z} = y\boldsymbol{w} + (\boldsymbol{z} - y\boldsymbol{w}), \tag{7.5}$$

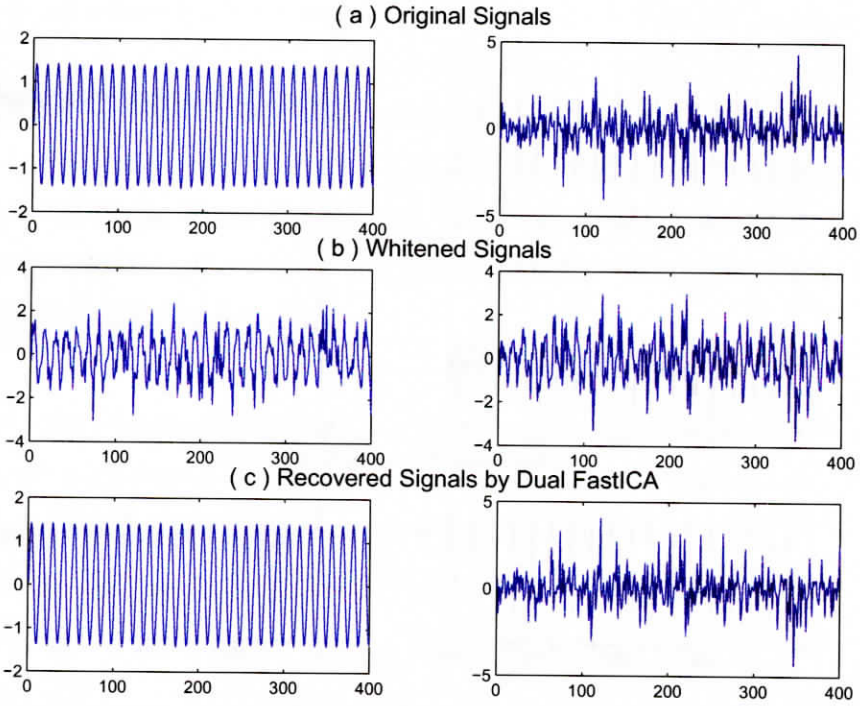

Figure 7.2: (a) Original sub-Gaussian Signals (left) and impulsive noise (right) . (b) Globally mixed signals or Whitened Signals. (c) Recovered signals by Dual FastICA.

which implies $\|\boldsymbol{z}\|^2 = y^2 + r^2$, where $r = \left(\|\boldsymbol{z}\|^2 - y^2\right)^{1/2}$. The aim of Dual FastICA algorithm is to find an orthogonal direction $\boldsymbol{w}$ for independent components by maximizing the objective function $\mathrm{E}\{\pm G(r)\}$, that is

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \ \mathrm{E}\{\pm G(r)\} \tag{7.6}$$

where $\boldsymbol{w}$ is a unit vector in R^n and $G(r)$ is the contrast function. Our proposed contrast function for robust ICA is

$$G(r) = -\exp(-r^2/2) \tag{7.7}$$

Figure 7.0.2 represent the value of the objective functions $\mathrm{E}\{-G(y)\}$ and $\mathrm{E}\{-G(r)\}$ with respect to $\boldsymbol{w} = \boldsymbol{w}(\theta) = (\cos\theta \quad \sin\theta)^T$ with $0 \leq \theta \leq 2\pi$ based on the contrast functions proposed in (7.2) and (7.7), respectively. The solid line with marker style (+) and the dotted line with marker style (*) represent the value of the objective functions $\mathrm{E}\{-G(y)\}$ and $\mathrm{E}\{-G(r)\}$, respectively. Figures 7.0.2a-7.0.2b and 7.0.2c-7.0.2d represent the value of the objective functions $\mathrm{E}\{-G(y)\}$ and $\mathrm{E}\{-G(r)\}$ for prewhitened sub-Gaussian signals generated from uniform distribution and beta distribution respectively. Figures 7.0.2e-7.0.2f,

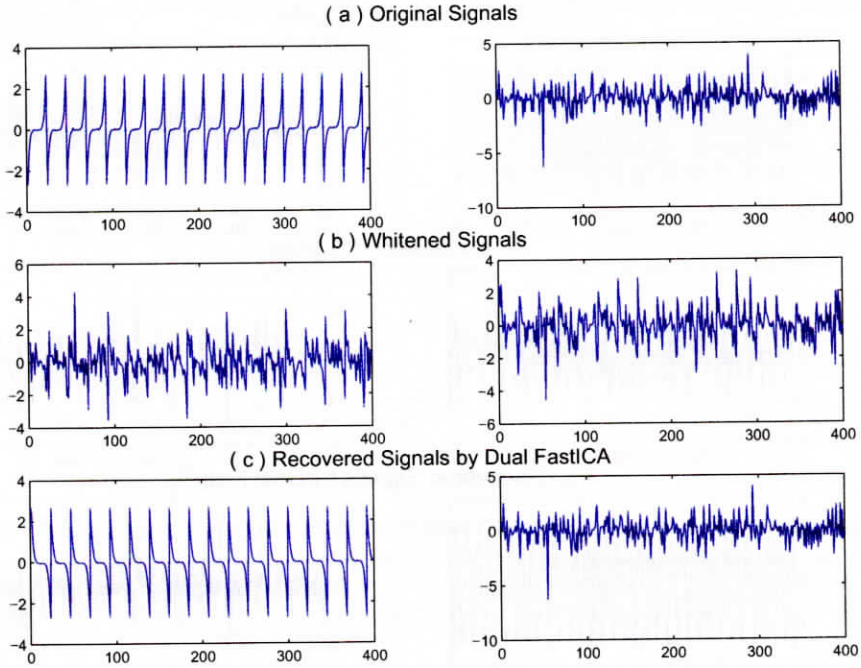

Figure 7.3: (a) Original supper-Gaussian Signals (left) and impulsive noise (right) . (b) Globally mixed signals or Whitened Signals. (c) Recovered signals by Dual FastICA.

7.0.2g-7.0.2h and 7.0.2i-7.0.2j represent the value of the objective functions $\mathrm{E}\{-G(y)\}$ and $\mathrm{E}\{-G(r)\}$ for prewhitened supper-Gaussian signals generated from Laplace, Exponential and Gamma distribution, respectively. From all Figures discussed above, we see that behavior of both objective function are almost similar in some cases and symmetric in other cases. Therefore, performance of both objective function for robust ICA should be similar or symmetric. From the situation discussed above, we consider Dual FastICA. The fixed-point iteration based on Gradient method for (7.6) is

$$\boldsymbol{w} \longleftarrow \mathrm{E}\left\{\frac{y}{r}g(r)\boldsymbol{z}\right\} \tag{7.8}$$

under the constraint $\|\boldsymbol{w}\|^2 = 1$. The basic fixed-point iteration in Dual FastICA based on approximative Newton iteration for (7.1) is

$$\boldsymbol{w} \longleftarrow \mathrm{E}\left\{\frac{y}{r}g(r)\boldsymbol{z}\right\} - \mathrm{E}\left\{\frac{1}{r}\left(1+\frac{y^2}{r^2}\right)g(r) - \frac{y^2}{r^2}g'(r)\right\}\boldsymbol{w} \tag{7.9}$$

under the constraint $\|\boldsymbol{w}\|^2 = 1$, where $g(r) = G'(r) = r\exp(-r^2/2)$ and $g'(r) = G''(r) = (1-r^2)\exp(-r^2/2)$.

Dual FastICA algorithm finds the orthogonal matrix $W = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m)$ to obtain the independent components (ICs) vector $\boldsymbol{y}$ from the whitened vector $\boldsymbol{z}$ by the orthogonal transformation $\boldsymbol{y} = W^T\boldsymbol{z}$. Also orthogonal column vectors $\boldsymbol{w}_i$, i=1, 2, ..., m can be estimated sequentially or simultaneously by Dual FastICA under deflationary and symmetric orthogonalization, respectively. We give a detailed version of the Dual FastICA algorithm under β-prewhitening that uses both deflationary and symmetric orthogonalization in tables (7.1) and 7.2), respectively.

---

1. Whiten the data by β-prewhitening to give $\boldsymbol{z}$.
2. Choose m, the number of ICs to estimate. Set counter $p \longleftarrow 1$.
3. Choose an initial value of unit norm for $\boldsymbol{w}_p$, (e.g., randomly).
4. Let $\boldsymbol{w}_p \longleftarrow \mathrm{E}\left\{\frac{y}{r}g(r)\boldsymbol{z}\right\} - \mathrm{E}\left\{\frac{1}{r}\left(1+\frac{y^2}{r^2}\right)g(r) - \frac{y^2}{r^2}g'(r)\right\}\boldsymbol{w}_p$
5. Do the orthogonalization: $\boldsymbol{w}_p \longleftarrow \boldsymbol{w}_p - \sum_{j=1}^{p-1}(\boldsymbol{w}_p^T\boldsymbol{w}_j)\boldsymbol{w}_j$.
6. Let $\boldsymbol{w}_p \longleftarrow \frac{\boldsymbol{w}_p}{\|\boldsymbol{w}_p\|}$
7. If $\boldsymbol{w}_p$ has not converged, go back to step 4.
8. Set $p \longleftarrow p+1$. If $p \leq m$, go back to step 3.

Table 7.1: Dual FastICA Algorithm under β-prewhitening for estimating several ICs with deflationary orthogonalization. The expectations are estimated in practice as sample averages.

We note that fixed-point iteration based on equation (7.9) recovers only sub-Gaussian signals. In order to recover super-Gaussian signals, we modified fixed point iteration (equation 7.9) heuristically as

$$\boldsymbol{w} \longleftarrow \mathrm{E}\left\{\frac{y}{r}g(r)\boldsymbol{z}\right\} + \mathrm{E}\left\{\frac{1}{r}\left(1+\frac{y^2}{r^2}\right)g(r) - \frac{y^2}{r^2}g'(r)\right\}\boldsymbol{w}. \tag{7.10}$$

This modification works well to recover supper-Gaussian signals. We are trying to find the theoretical justification for this modification.

To demonstrate the validity of Dual FastICA algorithm, we consider a sub-Gaussian signals and impulsive noise as source signals that is shown in Figure 7.2a. Then we linearly mixed source signals by an orthogonal matrix. Figure 7.2b shows the globally mixed signals or prewhitened signals. In order to recover original signals from the prewhitened data set, we

1. Whiten the data by β-prewhitening to give $\boldsymbol{z}$.
2. Choose m, the number of independents components (ICs) to estimate.
3. Choose initial values for the $\boldsymbol{w}_i$, (i=1,2,..., m) each of unit norm. Orthogonalize the matrix W as in step 5 below.
4. For every i = 1, 2, ..., m, let $\boldsymbol{w}_i \longleftarrow \mathrm{E}\left\{\frac{y}{r}g(r)\boldsymbol{z}\right\} - \mathrm{E}\left\{\frac{1}{r}\left(1+\frac{y^2}{r^2}\right)g(r) - \frac{y^2}{r^2}g'(r)\right\}\boldsymbol{w}_i$
5. Do a symmetric orthogonalization of the matrix $W = (\boldsymbol{w}_1, ..., \boldsymbol{w}_m)$ by
$$W \longleftarrow (WW^T)^{-\frac{1}{2}}W.$$
6. If not converged, go back to step 4.

Table 7.2: Dual FastICA Algorithm under β-prewhitening for estimating several ICs with symmetric orthogonalization. The expectations are estimated in practice as sample averages.

applied Dual FastICA algorithm given in table (7.1). Figure 7.2c shows the recovered signals by Dual FastICA. We see that recovered signals are almost similar to the original signals. Again we consider a supper-Gaussian signals and impulsive noise as source signals that is shown in Figure 7.3a. Then we linearly mixed source signals by an orthogonal matrix. Figure 7.3b shows the globally mixed signals or prewhitened signals. In order to recover original signals from the prewhitened data set, we applied Dual FastICA algorithm given in table (7.1) changing the fixed-point iterative formula (equation 7.9) by (equation 7.10). Figure 7.3c shows the recovered signals by Dual FastICA. We see that recovered signals are almost similar to the original signals. Also we investigated the performance for high-dimensional data set and found the same results.

7.0.3 Robustness and Consistency

Let us discuss robustness of FastICA and Dual FastICA simultaneously to find out our main objectives. For convenience of presentation, let us define $G(y)$ and $G(r)$ by $\varrho(y^2)$ and $\varrho(r^2)$, respectively. Then estimating equations for FastICA and Dual FastICA with respect to recovering vector $\boldsymbol{w}$ are

$$2\mathrm{E}\{\varphi(y^2)y(\boldsymbol{z} - y\boldsymbol{w})\} = O \tag{7.11}$$

$$-2\mathrm{E}\{\varphi(r^2)y(\boldsymbol{z} - y\boldsymbol{w})\} = O \tag{7.12}$$

If $\boldsymbol{w} = \boldsymbol{e}$, (i-th basic vector), then

$$\mathrm{E}\{\varphi(z_i^2) z_i(\boldsymbol{z} - z_i \boldsymbol{e}_i)\} = O \tag{7.13}$$

$$\mathrm{E}\left\{\varphi\left(||\boldsymbol{z}_{(-i)}||^2\right) z_i(\boldsymbol{z} - z_i \boldsymbol{e}_i)\right\} = O \tag{7.14}$$

Thus estimators obtained by (7.11) and (7.12) are consistent. Then

$$||\varrho'(y^2)||^2 = 4\{\varphi(y^2)y\}^2\{||\boldsymbol{z}||^2 - y^2\} = 4h^2(y)\left(\frac{1}{\gamma^2} - 1\right) < \infty, \quad \text{if } \gamma \neq 0 \text{ and } h(y) < \infty$$

$$||\varrho'(r^2)||^2 = 4\{\varphi(r^2)y\}^2\{||\boldsymbol{z}||^2 - y^2\} = 4h^2(r)\left(\frac{1}{\gamma^2} - 1\right)^{-1} < \infty, \quad \text{if } \gamma \neq \pm 1 \text{ and } h(y) < \infty$$

where, $h(y) = \varphi(y^2)y^2$ and $h(r) = \varphi(r^2)r^2$ are assumed to be bounded, and $\gamma = \frac{\boldsymbol{w}^T\boldsymbol{z}}{||\boldsymbol{z}||} = \cos\theta$. Note that $\gamma = 0 \Longrightarrow \theta = 90^o, 270^o, ..., \Longrightarrow$ directions of $\boldsymbol{w}$ and $\boldsymbol{z}$ are perpendicular of each other, that is, outlier $\boldsymbol{z}$ is orthogonal to $\boldsymbol{w}$. On the other hand, $\gamma = \pm 1 \Longrightarrow \theta = 0^o, 180^o, ..., \Longrightarrow$ directions of $\boldsymbol{w}$ and $\boldsymbol{z}$ are same or opposite. From the above discussion, clearly, we see that influence function for the recovering vector $\boldsymbol{w}$ obtained by FastICA becomes unbounded when outlying vectors are almost orthogonal to $\boldsymbol{w}$. On the other hand, influence function for the recovering vector $\boldsymbol{w}$ obtained by Dual fastICA becomes unbounded when directions of outlying vectors are almost same or opposite.

7.1 Robust FastICA in Presence of All-Rounding Outliers

From the discussion in subsection 7.0.3, we see that both FastICA and Dual FastICA are not robust for outliers of all directions. An attempt would be made to propose a new robust FastICA algorithm in presence of outliers in all directions by combining FastICA and Dual FastICA. A possible way is to find a recovering orthogonal vector $\boldsymbol{w}_k$ such that

$$\boldsymbol{w}_k = \begin{cases} \boldsymbol{w}_i, & \text{by FasICA if } \gamma = \frac{\boldsymbol{w}_k^T\boldsymbol{z}_{out}}{||\boldsymbol{z}_{out}||} \neq 0 \\ \boldsymbol{w}_j, & \text{by Dual FasICA otherwise,} \end{cases} \tag{7.15}$$

where, $i \neq j = 1, 2, ..., m$; $k = 1, 2, ..., m$.

Another possible way is to find a recovering orthogonal vector $\boldsymbol{w}$ by maximizing $\mathrm{E}\{G^*(\boldsymbol{w})\}$, where $G^*(\boldsymbol{w})$ is the convolution of two contrast functions $G(y)$ and $G(r)$ and is defined by

$$G^*(\boldsymbol{w}) = \exp\left\{-\frac{1}{2}\{(1-\alpha)y^2 + \alpha r^2\}\right\} \tag{7.16}$$

7.2 Robust FastICA by Maximizing β-Negentropy

FastICA fixed-point algorithm extracts all independent components sequentially or simultaneously by maximizing the approximation of negentropy (Hyvärinen et al., 2001). However, in FastICA algorithm, some contrast functions are suggested by a little bit heuristic way for robust ICA. So one cannot easily define contrast function for robust ICA. In this section we would like to discuss a new robust FasICA algorithm by maximizing the approximation of β-negentropy. To define β-negentropy, let us start with the classical entropy. The differential entropy H of a random vector $\boldsymbol{z}$ with density $p_{\boldsymbol{z}}$ is defined as

$$H = -\int p_{\boldsymbol{z}} \log p_{\boldsymbol{z}} d\boldsymbol{z} = -\mathrm{E}(\log p_{\boldsymbol{z}}). \tag{7.17}$$

An important property of entropy is that the the Gaussian variable has the largest entropy among all random variables of unit variance. This means that entropy could be used as a measure of non-Gaussianity. Let us extend classical entropy (H) as β-entropy (H_β) and defined by

$$H_\beta = \frac{1}{\beta(\beta+1)}\left(1 - \int p_{\boldsymbol{z}}^{\beta+1} d\boldsymbol{z}\right) \tag{7.18}$$

Note that $\lim_{\beta \downarrow 0} H_\beta = H$.

Property: Under a moment maching condition, Gaussian variable has the largest entropy.

Proof: Let $\varphi_{\boldsymbol{z}} \sim N(\boldsymbol{\mu}, V)$ and $p_{\boldsymbol{z}}$ is the data distribution. Then

$$\begin{aligned} H_\beta(\varphi_{\boldsymbol{z}}) - H_\beta(p_{\boldsymbol{z}}) &= \int \frac{1}{\beta(\beta+1)}\left(p_{\boldsymbol{z}}^{\beta+1} - \varphi_{\boldsymbol{z}}^{\beta+1}\right) d\boldsymbol{z} \\ &= D_\beta(p_{\boldsymbol{z}}, \varphi_{\boldsymbol{z}}) + \int \frac{1}{\beta}\varphi_{\boldsymbol{z}}^{\beta}\left(p_{\boldsymbol{z}} - \varphi_{\boldsymbol{z}}\right) d\boldsymbol{z} \\ &= D_\beta(p_{\boldsymbol{z}}, \varphi_{\boldsymbol{z}}) > 0 \end{aligned}$$

Hence

$$H_\beta(\varphi_{\boldsymbol{z}}) > H_\beta(p_{\boldsymbol{z}}) \tag{7.19}$$

Note that

$$\int \frac{1}{\beta}\varphi_{\boldsymbol{z}}^{\beta}\left(p_{\boldsymbol{z}} - \varphi_{\boldsymbol{z}}\right) d\boldsymbol{z} = 0 \tag{7.20}$$

Therefore, β-entropy can be used as a measure of non-Gaussianity. We can define a measure that is zero for the Gaussian variable and positive for other variables can be simply obtained

from β-entropy, and is considered to be β-negentropy. Hence, β-negentropy can be defined by

$$J_\beta(\boldsymbol{z}) = H_\beta(\varphi_{\boldsymbol{z}}) - H_\beta(p_{\boldsymbol{z}}) \tag{7.21}$$

We note that $\lim_{\beta \downarrow 0} J_\beta(\boldsymbol{z}) = J(\boldsymbol{z}) = H(\varphi_{\boldsymbol{z}}) - H(p_{\boldsymbol{z}})$, classical negentropy.
Therefore, we can extract independent components by maximizing β-negentropy. An important property of β-negentropy is that it is invariant under orthogonal transformation.

7.3 Cluster Analysis Based on Minimum β-Divergence Estimators

Cluster analysis is a technique for grouping data and finding structures in data. The most common application of clustering methods is to partition a data set into clusters or classes, where similar data are assigned to the same cluster while dissimilar data should belong to different clusters. The resulting data partition improves data understanding and reveals its internal structure. There are some popular clustering methods, for example, K-mean clustering and Fuzzy clustering. However, one problem in those method is that the number of clusters should be known in advance, which is difficult in practice. Therefore, we would like to propose new algorithm based on Minimum β-Divergence Estimators for data clustering. If observed data vector follows multivariate Gaussian distribution, then using 5.50 and 5.51 we can partition the data set into appropriate number of clusters. If observed data vector follows multivariate non-Gaussian distribution, then using 6.15 and 6.17, one can partition the data set into appropriate number of clusters as before.

Chapter 8

Conclusion Remarks

In this thesis, we proposed some new algorithms for multivariate analysis, especially robust prewhitening for ICA, exploring local structures for both PCA and ICA by minimum β-divergence method. In the context of minimum β-divergence method, the β-divergence between the empirical distribution of a sample (data distribution) and the specific distribution corresponding to the problem under study is minimized with respect to the parameters to be estimated (Minami and Eguchi, 2002; Mollah, Minami and Eguchi, 2006). The minimum β-divergence method with $\beta = 0$ reduces to the minimum Kullback-Leibler (K-L) divergence method.

A prewhitening data set is necessary in most of the ICA algorithms. It reduces the complexity of the ICA problems. However, existing prewhitening procedures are not suitable always that we discussed in detail in chapter 4. A new adaptive robust prewhitening named β-prewhitening procedure is proposed by minimizing the empirical β-divergence over the space of all the Gaussian distributions. Also we proposed a new measure of performance index to assess the performance of prewhitening procedures. The performance of the proposed prewhitening is compared with the classical prewhitening by newly proposed performance index and FastICA (Hyvärinen, 1999) using both synthetic and real data sets. Simulation result shows that β-prewhitening efficiently improves the performance over the classical prewhitening when outliers exist; it reduce to classical prewhitening otherwise.

A comparatively new problem in PCA is to explore local PCA structures for dimensionality reduction. However, existing methods for exploring local PCA structures gives misleading results if data set is corrupted by outliers or number of data cluster is unknown. To overcome

this problem, we propose a new learning algorithm to explore local PCA structures. The proposed method is based on a sequential application of the minimum β-divergence method to search local PCA structures sequentially. The proposed method searches the local PCA structure on the basis of a rule of sequential change of the shifting parameter and a local kernel vector. If the initial choice of the shifting parameter vector and the local kernel vector belongs to a data cluster, then all data belonging to that cluster are transformed into a local PCA structure considering the data in other clusters as outliers.

Same as previous, a comparatively new problem in ICA is to explore local ICA structures. However, existing methods for exploring local ICA structures gives misleading results if data set is corrupted by outliers or number of data cluster is unknown. To overcome this problem, we propose a new learning algorithm to explore local ICA structures. The proposed method is based on a sequential application of the minimum β-divergence method to search local ICA structures sequentially. The proposed method searches the local ICA structure on the basis of a rule of sequential change of the shifting parameter. If the initial choice of the shifting vector belongs to a data cluster, then all data belonging to that cluster are transformed into a local ICA structure considering the data in other clusters as outliers.

The value of the tuning parameter is a key in the performance of the proposed methods mentioned above. A cross-validation technique is proposed as an adaptive selection procedure for the tuning parameter β.

Also we presented some incomplete research work in this thesis. We would like to finish this incomplete research work in near future.

Appendix A

Existing Methods for Multivariate Analysis Related to our Research

Robust FastICA algorithm (Hyvärinen, 1999), Infomax ICA algorithm (Bell et al., 1995), ICA mixture models (Lee et al., 2000b) and Mixture of probabilistic PCA algorithms (Tipping et al., 1999) are directly related to our main research. Therefore, we discuss the summary of algorithms early mentioned in this section.

A.1 ICA Algorithms Related to Our Research

The following ICA algorithms are related with our main research described in this thesis.

A.1.1 FastICA Algorithm

FastICA algorithm finds the orthogonal matrix $W = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m)$ to obtain the independent components (ICs) vector $\boldsymbol{y}$ from the whitened vector $\boldsymbol{z}$ by the orthogonal transformation $\boldsymbol{y} = W^T\boldsymbol{z}$. Also orthogonal column vectors $\boldsymbol{w}_i$, i=1, 2, ..., m can be estimated sequentially or simultaniously by FastICA under deflationary and symmetric orthogonalization, respectively. We give a detailed version of the FastICA algorithm under β-prewhitening that uses both deflationary and symmetric orthogonalization in tables (2.2) and 2.3), respectively.

1. Whiten the data by β-prewhitening to give $\boldsymbol{z}$.
2. Choose m, the number of ICs to estimate. Set counter $p \longleftarrow 1$.
3. Choose an initial value of unit norm for $\boldsymbol{w}_p$, (e.g., randomly).
4. Let $w_p \longleftarrow \mathrm{E}\{\boldsymbol{z}g(\boldsymbol{w}_p^T\boldsymbol{z})\} - \mathrm{E}\{g\,'(\boldsymbol{w}_p^T\boldsymbol{z})\}$,
5. Do the orthogonalization: $\boldsymbol{w}_p \longleftarrow \boldsymbol{w}_p - \sum_{j=1}^{p-1}(\boldsymbol{w}_p^T\boldsymbol{w}_j)\boldsymbol{w}_j$.
6. Let $\boldsymbol{w}_p \longleftarrow \frac{\boldsymbol{w}_p}{\|\boldsymbol{w}_p\|}$
7. If $\boldsymbol{w}_p$ has not converged, go back to step 4.
8. Set $p \longleftarrow p+1$. If $p \leq m$, go back to step 3.

Table A.1: FastICA Algorithm under β-prewhitening for estimating several ICs with deflationary orthogonalization. The expections are estimated in practice as sample averages.

1. Whiten the data by β-prewhitening to give $\boldsymbol{z}$.
2. Choose m, the number of independents components (ICs) to estimate.
3. Choose initial values for the $\boldsymbol{w}_i$, (i=1,2,..., m) each of unit norm. Orthogonalize the matrix W as in step 5 below.
4. For every i = 1, 2, ..., m, let $w_i \longleftarrow \mathrm{E}\{\boldsymbol{z}g(\boldsymbol{w}_i^T\boldsymbol{z})\} - \mathrm{E}\{g\,'(\boldsymbol{w}_i^T\boldsymbol{z})\}$,
5. Do a symmetric orthogonalization of the matrix $W = (\boldsymbol{w}_1, ..., \boldsymbol{w}_m)$ by
$$W \longleftarrow (WW^T)^{-\frac{1}{2}}W.$$
6. If not converged, go back to step 4.

Table A.2: FastICA Algorithm under β-prewhitening for estimating several ICs with symmetric orthogonalization. The expections are estimated in practice as sample averages.

A.1.2 Infomax ICA Algorithm

Bell and Sejnowski (1995) proposed Infomax (Information Maximization) ICA algorithm by maximizing joint entropy. The derivation is based on a simple neural network architechture that can realize the mapping from $\boldsymbol{x}$ to $\boldsymbol{y} = g(\boldsymbol{u})$. They show that maximizing the joint entropy $H(\boldsymbol{y})$ of the output of a neural processor can approximately minimize the mutual information among the output components $y_i = g_i(u_i)$, where $g_i(u_i)$ is an invertible monotonic nonlinearity and $\boldsymbol{u} = W\boldsymbol{x}$. The joint entropy at the outputs of a neural network is

$$H(y_1, y_2, ..., y_m) = H(y_1) + ... + H(y_m) - I(y_1, y_2, ..., y_m), \tag{A.1}$$

where, $H(y_i)$ are the marginal entropies of the outputs and $I(y_1, y_2, ..., y_m)$ is their mutual information. Maximizing $H(y_1, y_2, ..., y_m)$ consists of maximizing the marginal entropies and minimizing the mutual information. Equation A.1 can be expressed in vector notation as

$$H(\boldsymbol{y}) = H(y_1) + ... + H(y_m) - I(\boldsymbol{y}), \tag{A.2}$$

Each marginal entropy can be written as

$$H(y_i) = -E\{\log p(y_i)\}, \tag{A.3}$$

The nonlinear mapping between the output density $p(y_i)$ and source estimate density $p(u_i)$ can be described by the absolute value of the derivative with respect to u_i, that is

$$p(y_i) = \frac{p(u_i)}{|\frac{\partial y_i}{\partial u_i}|}, \tag{A.4}$$

which can be substituted in equation A.6 giving

$$H(y_i) = -E\{\log \frac{p(u_i)}{|\frac{\partial y_i}{\partial u_i}|}\}, \tag{A.5}$$

Rewriting equation A.2 gives

$$H(\boldsymbol{y}) = -\sum_{i=1}^{m} E\{\log \frac{p(u_i)}{|\frac{\partial y_i}{\partial u_i}|}\} - I(\boldsymbol{y}), \tag{A.6}$$

Then learning infomax rule (Bell and Sejnowski, 1995) is

$$\frac{\partial H(\boldsymbol{y})}{\partial W} = (W^T)^{-1} + \left(\frac{\frac{\partial p(\boldsymbol{u})}{\partial \boldsymbol{u}}}{p(\boldsymbol{u})}\right) \boldsymbol{x}^T, \tag{A.7}$$

A much more efficient way to maximize the joint entropy is to follow the natural gradient. The natural gradient rescales the entropy gradient by post-multiplying the entropy gradient by $W^T W$ giving

$$\triangle W \propto \frac{\partial H(\boldsymbol{y})}{\partial W} W^T W = \left[I + \left(\frac{\frac{\partial p(\boldsymbol{u})}{\partial \boldsymbol{u}}}{p(\boldsymbol{u})}\right) \boldsymbol{u}^T \right] W, \tag{A.8}$$

as proposed by Amari et. al. (1996), or equivalently the relative gradient by Cardoso et al (1996).

An alternative way to derive the general infomax learning is given by the maximum likelihood

estimator (MLE). The probability density function of the observations $\boldsymbol{x}$ can be expressed as (Amari, Chen and Cichocki, 1997)

$$p(\boldsymbol{x}) = |\det(W)|p(\boldsymbol{u}), \tag{A.9}$$

where, $p(\boldsymbol{u}) = \prod_{i=1}^{m} p_i(u_i)$ is the hypothesized distribution of $p(\boldsymbol{u})$. The log-likelihood of equation A.9 is

$$L(\boldsymbol{u}, W) = \log|\det(W)| + \sum_{i=1}^{m} \log p_i(u_i), \tag{A.10}$$

Maximizing the log-likelihood with respect to W gives a learning algorithm for W, (Bell and Sejnowski, 1995)

$$\triangle W \propto \left[(W^T)^{-1} + \varphi(\boldsymbol{u})\boldsymbol{x}^T\right], \tag{A.11}$$

where,

$$\varphi(\boldsymbol{u}) = -\left(\frac{\frac{\partial p(\boldsymbol{u})}{\partial \boldsymbol{u}}}{p(\boldsymbol{u})}\right) = \left[-\frac{\frac{\partial p(u_1)}{\partial u_1}}{p(u_1)}, \ldots, -\frac{\frac{\partial p(u_m)}{\partial u_m}}{p(u_m)}\right]^T \tag{A.12}$$

An efficient way to maximize the log-likelihood is to follow the natural gradient. The natural gradient rescales the gradient by post-multiplying the gradient of the log-likelihood by W^TW giving (Amari, 1998)

$$\triangle W \propto \frac{\partial L(\boldsymbol{u}, W)}{\partial W} W^T W = [I - \varphi(\boldsymbol{u})\boldsymbol{u}^T]W, \tag{A.13}$$

as proposed by Amari et. al. (1996), or equivalently the relative gradient by Cardoso et al (1996). It should be noted here that Infomax ICA Algorithm is a limiting case of minimum β-diveregence algorithm for ICA.

A.1.3 Extended Infomax ICA Algorithm

The conventional infomax ICA algorithms can separate sub- or super-Gaussian signals based on the contrast function. However, in some situation, it is difficult to know in advance about the type of source signals those are hidden in the mixed signals. To overcome this problem, Lee, Girolami and Sejnowski (1999) extended the infomax ICA algorithm. This method can blindly separate the sub- or super-Gaussian signals. The switching between the sub- and super-Gaussian learning rule is

$$\triangle W \propto [I - K\tanh(\boldsymbol{u})\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{u}^T]W, \quad \begin{cases} k_i = 1 & : \text{super-Gaussian} \\ k_i = -1 & : \text{sub-Gaussian} \end{cases} \tag{A.14}$$

where k_i are the elements of the m-dimensional diagonal matrix K. It is also called switching matrix. The switching component k_i is estimated by

$$k_i = \text{sign}\Big(E\{\text{sech}^2(u_i)\}E\{u_i^2\} - E\{[\tanh(u_i)]u_i\}\Big) \tag{A.15}$$

A.1.4 Conventional ICA Mixture models

Lee, Lewicki and Sejnowski (2000b) proposed ICA mixture model for extracting local ICA structures by modeling the observed data as a mixture of several ICA models. Assume that the data $\{x_t\}$ are drawn independently from a mixture density

$$p(\boldsymbol{x}_t \mid \Theta) = \sum_{k=1}^{c} p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k), \tag{A.16}$$

where C_k denotes the k-th non-Gaussian class and $\Theta = \{\theta_1, ..., \theta_c\}$ are the unknown parameters. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ be a random sample drawn from (5.17), then the log-likelihood of the data for the unknown parameter $\Theta = \{\theta_1, \theta_2, ...\theta_c\}$ is given by

$$L = \sum_{t=1}^{n} \log p(\boldsymbol{x}_t \mid \Theta). \tag{A.17}$$

Assume that the component densities are non-Gaussian and the data with in each class are described by

$$\boldsymbol{x}_t = A_k \boldsymbol{s}_k + \boldsymbol{b}_k \tag{A.18}$$

where A_k is the mixing matrix and $\boldsymbol{b}_k$ is the bias vector for class k. The vector $\boldsymbol{s}_k$ is called the source vector, whose components are assume to be independent of each other. The task is to classify the unlabeled data points and to determine the parameters for each class, $(A_k, \boldsymbol{b}_k)$ and the probability of each class $p(C_k|\boldsymbol{x}_t, \Theta)$ for each data point. The gradients of the parameters for class k is given by

$$\frac{\partial L}{\partial \theta_k} = \sum_{t=1}^{n} \frac{1}{p(\boldsymbol{x}_t \mid \Theta)} \frac{\partial}{\partial \theta_k} p(\boldsymbol{x}_t \mid \Theta) \tag{A.19}$$

$$= \sum_{t=1}^{n} \frac{\frac{\partial}{\partial \theta_k} p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k)}{p(\boldsymbol{x}_t \mid \Theta)} \tag{A.20}$$

Using the Bayes relation, the class probability for a given data vector $\boldsymbol{x}_t$ is

$$p(C_k \mid \boldsymbol{x}_t, \Theta) = \frac{p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k)}{\sum_{k=1}^{c} p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k)} \tag{A.21}$$

Substituting $(A.21)$ in $(A.20)$ leads to

$$\frac{\partial L}{\partial \theta_k} = \sum_{t=1}^{n} \frac{p(C_k \mid \boldsymbol{x}_t, \Theta)}{p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k)} \frac{\partial}{\partial \theta_k} p(C_k)p(\boldsymbol{x}_t \mid \theta_k, C_k) \tag{A.22}$$

$$= \sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta) \frac{\partial}{\partial \theta_k} \log p(\boldsymbol{x}_t \mid \theta_k, C_k) \tag{A.23}$$

Now, $\frac{\partial L}{\partial \boldsymbol{b}_k} = 0$ implies

$$\boldsymbol{b}_k = \frac{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)\boldsymbol{x}_t}{\sum_{t=1}^{n} p(C_k \mid \boldsymbol{x}_t, \Theta)} \tag{A.24}$$

and

$$\frac{\partial L}{\partial A_k} \propto p(C_k \mid \boldsymbol{x}_t, \Theta) \frac{\partial}{\partial A_k} \log p(\boldsymbol{x}_t \mid \theta_k, C_k) \tag{A.25}$$

$$\propto p(C_k \mid \boldsymbol{x}_t, \Theta) A_k [I - K \tanh(\boldsymbol{s}_k)\boldsymbol{s}_k^T - \boldsymbol{s}_k \boldsymbol{s}_k^T] \tag{A.26}$$

where $\boldsymbol{s}_k = A_k^{-1}(x_t - \boldsymbol{b}_k)$ and the switching matrix K is defined in Appendix 1.3. Note that $W_k = A_k^{-1}$ is called the filter matrix. The switching component k_i is obtained as

$$k_i = \text{sign}\Big(E\{\text{sech}^2(s_{k,i})\}E\{s_{k,i}^2\} - E\{[\tanh(s_{k,i})]k_{k,i}\}\Big) \tag{A.27}$$

A.2 PCA Algorithm Related to Our Research

The following PCA algorithms are related with our main research described in this thesis.

A.2.1 Mixture of Probabilistic PCA

Tipping et al. (1999) proposed mixture of probabilistic PCA (PPCA) model for extracting local PCA structures by modeling the observed data as a mixture of several PPCA models.

PPCA :

Let us consider the latent variable model as

$$\boldsymbol{x} = W\boldsymbol{s} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{A.28}$$

where, $\boldsymbol{x}$ is d-dimensional observed data vector, $\boldsymbol{s}$ is q-dimensional latent vector whose components are assumed to be independent and Gaussian with unit variance, that is $\boldsymbol{s} \sim N(\boldsymbol{0}, I)$. W is the $d \times q$ parameter matrix which contains factor loadings, $\boldsymbol{\mu}$ is the shifting parameter

and ϵ is the noise term. For the case of isotropic noise $\epsilon \sim N(0, \sigma^2 I)$, equation A.28 implies a probability distribution over $\boldsymbol{x}$-space for a given $\boldsymbol{s}$ of the form

$$p(\boldsymbol{x}|\boldsymbol{s}) = (2\pi\sigma^2)^{-d/2} \exp\left\{-\frac{1}{2\sigma^2}||\boldsymbol{x} - W\boldsymbol{s} - \boldsymbol{\mu}||^2\right\} \tag{A.29}$$

where

$$p(\boldsymbol{s}) = (2\pi)^{-q/2} \exp\left\{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right\}. \tag{A.30}$$

We obtain the marginal distribution of $\boldsymbol{x}$ in the form

$$\begin{aligned} p(\boldsymbol{x}) &= \int p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})d\boldsymbol{s} && \text{(A.31)} \\ &= (2\pi)^{-d/2}|C|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T C^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right\}, && \text{(A.32)} \end{aligned}$$

where the model covariance is

$$C = \sigma^2 I + WW^T. \tag{A.33}$$

The log-likelihood of observing the data under this model is

$$\begin{aligned} L &= \sum_{t=1}^{n} \ln\{p(\boldsymbol{x}_t)\} && \text{(A.34)} \\ &= -\frac{n}{2}\{d\ln(2\pi) + \ln|C| + \mathrm{tr}(C^{-1}S_{\boldsymbol{x}})\} && \text{(A.35)} \end{aligned}$$

where

$$S_{\boldsymbol{x}} = \frac{1}{n}\sum_{t=1}^{n}(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^T \tag{A.36}$$

is the sample covariance matrix of the observed $\{\boldsymbol{x}_t\}$. Then maximum likelihood (ML) estimators are

$$\boldsymbol{\mu}_{ML} = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{x}_t \tag{A.37}$$

$$W_{ML} = U_q(\Lambda_q - \sigma^2 I)^{1/2}R, \tag{A.38}$$

and

$$\sigma^2_{ML} = \frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j, \tag{A.39}$$

where the q column vectors in the $d \times q$ matrix U_q are eigenvectors of $S_{\boldsymbol{x}}$, with corresponding eigenvalues in the $q \times q$ diagonal matrix Λ_q, and R is an arbitrary $q \times q$ orthogonal rotation matrix. Also note that $\lambda_{q+1}, ..., \lambda_d$ are the eigenvalues of $S_{\boldsymbol{x}}$ and so σ^2_{ML} has a clear interpretation as the average variance 'lost' per discarded dimension.

Mixtures of PPCA :

The log-likelihood of observing the data for such a mixture model is

$$L = \sum_{t=1}^{n} \ln\{p(\boldsymbol{x}_t)\} \tag{A.40}$$

$$= \sum_{t=1}^{n} \ln\left\{\sum_{i=1}^{c} \pi_i p(\boldsymbol{x}_t|i)\right\} \tag{A.41}$$

where, $p(\boldsymbol{x}|i)$ is the i-th PPCA model whose parameters are $(\boldsymbol{\mu}_i, W_i, \sigma_i^2)$ and π_i is the corresponding mixing proportion, with $\pi \geq 0$ and $\sum_{i=1}^{c} \pi_i = 1$. An iterative EM algorithm is developed for optimization of all of the model parameters $\pi_i, \boldsymbol{\mu}_i, W_i$ and $\sigma_i^2, (i = 1, 2, ..., c)$. If $R_{ti} = p(i|\boldsymbol{x}_t)$ is the posterior responsibility of mixture i for generating data point $\boldsymbol{x}_n$, given by

$$R_{ti} = \frac{p(\boldsymbol{x}_t|i)\pi_i}{p(\boldsymbol{x}_t)}, \tag{A.42}$$

where,

$$p(\boldsymbol{x}_t|i) = (2\pi)^{-d/2}|C_i|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_i)^T C_i^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_i)\right\}, \tag{A.43}$$

with the model covariance is

$$C_i = \sigma_i^2 I + W_i W_i^T. \tag{A.44}$$

Then iterative EM algorithm gives the following update rule for MLE

$$\tilde{\pi}_i = \frac{1}{n}\sum_{t=1}^{n} R_{ti} \tag{A.45}$$

$$\tilde{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{n} R_{ti}\boldsymbol{x}_t}{\sum_{t=1}^{n} R_{ti}}, \tag{A.46}$$

$$\tilde{W}_i = S_{\boldsymbol{x}i} W_i (\sigma_i^2 I + M_i^{-1} W_i^T S_{\boldsymbol{x}i} W_i), \tag{A.47}$$

$$\tilde{\sigma_i}^2 = \frac{1}{d}\mathrm{tr}\left(S_{\boldsymbol{x}i} - S_{\boldsymbol{x}i} W_i M_i^{-1} \tilde{W}_i^T\right), \tag{A.48}$$

where

$$S_{\boldsymbol{x}i} = \frac{1}{\pi_i n}\sum_{t=1}^{n} R_{ti}(\boldsymbol{x}_t - \tilde{\boldsymbol{\mu}}_i)(\boldsymbol{x}_t - \tilde{\boldsymbol{\mu}}_i)^T. \tag{A.49}$$

Note that given the established results for the single PPCA model, there is no need to use the iterative updates (A.47) and (A.48), since W_i and σ_i^2 can be determined by (A.38) and (A.39) using eigen-decomposition of $S_{\boldsymbol{x}i}$. Also note that all information should be entered in the EM algorithm through the sample covariance matrix.

Acknowledgments

First and foremost thanks are to almighty **ALLAH** for giving me strength, patience and ability to carry out this study within 3 years.

I wish to express my deepest sense of gratitude to **Prof. Shinto Eguchi** for giving me opportunity to accomplish this research as a foreign student under his kind supervision through the Japanese Government Scholarship scheme. I am so grateful to him for his valuable guidance, constant encouragement, financial assistance, and helpful discussions throughout the progress of this work. Without his insights, it was impossible for me to finish this dissertation.

I am so grateful to **Prof. Mihoko Minami** also for her continuous encouragement and helpful advices regarding my research during 3 years. I appreciate **Prof. Noboru Murata** of Waseda University for his brilliant comments on defense that help me very much to improve this dissertation.

I am grateful to **Prof. Shiro Ikeda** and **Prof. Hironori Fujisawa** of The Institute of Statistical Mathematics for their valuable suggestions during my presentation.

Acknowledgment is also given to all staff members of The Institute of Statistical Mathematics (ISM), Tokyo. Especially, **Noriko WATANABE - san** for their cordial cooperation and assistance during 3 years.

I appreciate my colleagues, **M. Henmi, T. Takenouchi** and **M. Kawakita**. I enjoyed a lot of seminar and meetings with them.

Last but not least, I am indebted to my parents for their enthusiastic encouragement, and

thanks to my wife, **Nayeema Sultana** for her encouragement, sacrifice and her infinite patience, and my both daughters, **Tasfia Noor** and **Alvia Noor**, for making life interesting in Japan.

Bibliography

Amari, S., Chichocki, A. and Yang, H. H. (1996): A new learning algorithm for blind source separation. In *Advances in Neural Information Processing 8*, Cambridge, MA: MIT Press, 757-763.

Amari, S., Chen, T. and Cichocki, A. (1997): Stability analysis of learning algorithm for blind source separation. *Neural Networks*, 10(8), 1345-1351.

Amari, S.(1998): Natural gradient works efficiently in learning. *Neural Computation*, 10, pp. 251-276.

Atick, J.J. (1992): Entropy minimization: A design principle for sensory perception? *International Journal of Neural Systems*, 3:81-90.

Barlow, H.B. (1989a): Unsupervised learning. *Neural Computation*, 1:295-311.

Barlow, H.B., Kaushal, T.P. and Mitchison, G.J. (1989a): Finding minimum entropy codes. *Neural Computation*, 1:412-423.

Bell, A. J. and Sejnowski, T. J. (1995): An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 1129-1159.

Bell, A. J. and Sejnowski, T. J. (1997): The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327-3338.

Belouchrani, A. and Cichocki, A. (2000): Robust whitening procedure in blind source separation context. *Electronics Letters*, vol. 36, no. 24, pp. 2050-2053.

Bishop, C.M. (1995): Neural Networks for Pattern Recognition, Oxford University Press, Walton Street, Oxford.

Campbell, N.A. (1980). Robust procedures in multivariate analysis 1: Robust co-variance estimation. *Appl. Statist.*, 29, 231-237.

Cardoso, J.-F. (1989): Source separation using high order moments. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'89)*, pages 2109-2112, Glasgow, UK.

Cardoso, J. F. and Souloumiac, A. (1993): Blind beamforming for non-Gaussian signals.

Proc. IEEE, 140, 362-370.

Cardoso, J.-F. and Laheld, B.H. (1996): Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 44(12): 3017-3030.

Cardoso, J.-F. (1997): Infomax and maximum likelihood for source separation. IEEE Letters on Signal Processing, 4:112-114.

Cardoso, J.-F. (1999): High-order contrasts for Independent Component Analysis. *Neural Computation*, vol. 11, pp. 157–192.

Caussinus, H. and Ruiz, A. (1990): Interesting projections of multidimensional data by means of generalized principal component analysis. *COMPSTAT*, 90, 121-126.

Choi, S., Cichocki, A. and Belouchrani, A. (2002): Second order nonstationary source separation. *Journal of VLSI Signal Processing*, vol. 32, no. 1-2, pp. 93-104.

Cichocki, A., Unbehauen, R. and Rummert, E., (1994): Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17): 1386-1387.

Cichoki, A. and Amari, S. (2002): *Adaptive Blind Signal and Image Processing* Wiley, New York.

Comon, P., Jutten, C. and Herault, J. (1991): Blind separation of sources, Part II: Problem statement. *Signal Processing*, vol. 24, pp. 11–20.

Comon. P. (1994): Independent component analysis - a new concept? Signal Processing, 36:287-314.

Cover, T. M. and Thomas, J. A.(1991): Elements of Information Theory. John Wiley & Sons.

Croux, C. and Haesbroeck, G. (2000): Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika*, 87, 603-618.

Deco, G. and Obradovic, D.,(1995): Linear redundancy reduction learning. *Neural Networks*, 8(5):751-755.

Delfosse, N. and Loubaton, P. (1995): Adaptive blind separation of independent sources: a deflation approach. Signal Processing, 45:59-83.

Donoho, D. (1981): On minimum entropy deconvolution. *Applied Time Series Analysis II*, pages 565-608. Academic Press.

Everson, R. and Roberts, S.J. (1999): Independent Component Analysis: A flexible nonlinearity and decorrelating manifold approach. *Neural Computation*, vol. 1, no. 8.

Eguchi, S. and Kano, Y. (2001): Robustifying maximum likelihood estimation," Research memorandum 802, Tokyo, Institute of Statistical Mathematics.

Field, D.J. (1994): What is the goal of sensory coding? *Neural Computation*, 6:559-601.

Friedman, J. H. and Tukey, J. W. (1974): A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, c-23(9):881-890.

Friedman, J. H. (1987): Exploratory projection pursuit. *J. of the American Statistical Association,* 82(397):249-266.

Gaeta, M. and Lacoume, J-L. (1990): Source separation without prior knowledge: The maximum likelihood approach. *Proceedings of Eusipo*, pp. 621–624.

Ghahramani, Z. and Beal, M. (2000): "Variational inference for Bayesian mixtures of factor analysers," in Advances in Neural Information Processing Systems, vol. 12, pp. 449–455.

Girolami, M. and Fyfe. C.,: Negentropy and kurtosis as projection pursuit indices provide generalised ICA algorithms. *in Advances in Neural Information Processing Systems*

Haykin, S. (1994): *Blind Deconvolution*, Prentice-Hall.

Haykin, S. (1996): *Adaptive Filter Theory*, Prentice-Hall International, 3rd edition.

Haykin, S. (1999): *Neural Networks.* Toronto: Prentice Hall.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J. and Stahel, W.A. (1986): *Robust Statistics: The Approach Based on Influence Functions.* Wiley, New York.

Harman, H. H. (1967): *Modern Factor Analysis.* University of Chicago Press, 2nd edition.

Hastie, T., Tibshirani, R. and Friedman, J. (2001): *The Elements of Statistical Learning.* New York: Springer.

Herault, J. and Jutten, C. (1986):Space or time adaptive signal processing by neural models. *Proceedings of AIP Conference: Neural Networks for Computing*, J.S. Denker, Ed. American Institute for Physics, vol. 151, pp. 206-211.

Higuchi, I. and Eguchi, S. (2004): Robust Principal Component Analysis With Adaptive Selection for Tuning Parameters. *J. Machine Learning Research* 5, 453-471. 82(397):249-266.

Hotelling, H. (1933): Analysis of a complex of statistical variables into principal components. *Jounnal of Educational Psychology*, 24, 417-441.

Huber, P.J.(1985): Projection pursuit. *The Annals of Statistics*, 13(2):435-475.

Hurri, J., Hyvrinen, A., and Oja, E., (1997): Wavelets and natural image statistics. *Proc. Scandinavian Conf. on Image Analysis '97*, Lappenranta, Finland.

Hyvärinen, A. and Oja, E. (1996): Simple neuron models for independent component analysis. *Int. Journal of Neural Systems*, 7(6):671-687.

Hyvärinen, A. (1997): One-unit contrast functions for independent component analysis: A statistical analysis. In *Neural Networks for Signal Processing VII(Proc. IEEEE Workshop on Neural Networks for Signal Processing)*. pages 388-397, Amelia Island, Florida.

Hyvärinen, A. and Oja, E. (1997): A fast fixed-point algorithm for independent component analysis, *Neural Computation*, 9(7):1483-1492.

Hyvärinen, A. (1998): Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. Neurocomputing, 22:49-67.

Hyvrinen, A. (1998b): New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems,* volume 10, pages 273-279. MIT Press.

Hyvärinen, A. and Oja, E. (1998c): Independent component analysis by general nonlinear Hebbian-like learning rules. *Signal Processing,* 64(3):301-313.

Hyvärinen, A. (1999): Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Network*, **10**(3), 626-34.

Hyvärinen, A. (1999a): Gaussian moments for noisy independent component analysis. IEEE Signal Processing Letters, 6(6):145-147.

Hyvärinen, A. (1999c): Survey on independent component analysis. *Neural Computing Surveys,* 2:94-128.

Hyvärinen, A. and Oja, E. (2000): Independent Component Analysis: A Tutorial. *Neural Networks,* 13(4-5):411-430.

Hyvärinen, A, Karhunen, J. and Oja, E. (2001): *Independent Component Analysis,* Wiley, New York.

Hyvärinen, A.: http://www.cis.hut.fi/projects/ica/fastica/

Jolliffe, I.T. (2002): *Principal Component Analysis.* Springer-Verlag.

Jones, M.C. and Sibson, R. (1987): What is projection pursuit ? *J. of the Royal Statistical Society, ser. A,* 150:1-36.

Jutten, C. and Herault, J. (1991): Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24, 1-20.

Kendall, M. (1975): *Multivariate Analysis.* Charles Griffin& Co., 1975.

Kambhatla, N. and Leen, T. K. (1997): Dimensionality reduction by local principal compo-

nent analysis. *Neural Computation*, Vol. 9, pp.1493-1516.

Karhunen, J. and Joutensalo, J. (1994): Representation and separation of signals using nonlinear PCA type learning. *Neural Networks*, vol. 7, no. 1, pp. 113–127.

Karhunen, J., Oja, E., Wang, L., Vigario, R., and Joutsensalo, J. (1997): A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486-504.

Lee, T.-W., Girolami, M., Lewicki, M. S. and Sejnowski, T. J. (1999): Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian Sources. *Neural Computation*. Vol.11(2): 609-633.

Lee,T.-W. and Lewicki, M. S. (2000a): The Generalized Gaussian Mixture Model Using ICA, International Workshop on Independent Component Analysis (ICA'00), Helsinki, 239-244.

Lee, T.-W., Lewicki, M. S. and Sejnowski, T. J. (2000b): ICA Mixture Models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation. *IEEE Trans. on Pattern Analysis an Machine Int.*, 22, pp. 1078-1089.

Lee, T.-W. (2001):*Independent Component Analysis: Theory and applications*, Kluwer Academic Publishers.

Linsker, R. (1989): An application of the principle of maximum information transfer to linear systems. *Advances in Neural Information Systems*, D.S. Touretzky, Ed. 1989, vol. 1, Morhan Kaufmann.

Linsker, R. (1992): Local synaptic learning rules suffice to maximise mutual information in a linear network. *Neural Computation*, vol. 4, pp. 691–702.

Mackay, D.J.C. (1996): Maximum likelihood and covariant algorithms for Independent Component Analysis. *Tech. Rep., University of Cambridge.*

McLachlan, G. J., and Peel, D. (2000): *Finite Mixture Models*, New York, Wiley.

Minami, M. and Eguchi, S. (2002): Robust Blind Source Separation by beta-Divergence. *Neural Computation* **14**, 1859-1886.

Minami, M. and Eguchi, S. (2003): Adaptive selection for minimum β-divergence method. *Proceedings of ICA-2003 Conference*, Nara, Japan.

Mollah, M.N.H., Minami, M. and Eguchi, S. (2005a): Robust Prewhitening for ICA by Minimizing β-Divergence and Its Application to FastICA, Submitted to *Neural Processing Letters.*

Mollah, M.N.H., Minami, M. and Eguchi, S. (2006): Exploring Latent Structure of Mixture

ICA Models by the Minimum β-Divergence Method, *Neural Computation*, **18(1)**, pp. 166-190.

Mollah, M.N.H., Sultana, N., Minami, M. and Eguchi, S. (2005b): Exploring Local PCA Structure for Dimensionality Reduction by Minimizing β-Divergence, Submitted to the *Journal of Machine Learning Research.*

Nadal, J.-P. and Parga, N. (1994): Non-linear neurons in the low noise limit: a factorial code maximizes information transfer. *Network,* 5:565-581.

Oja, E. (1982): A simplified neuron model as a principal component analyser. *J. Math. Biol.,*, vol. 15, pp. 267–273.

Nocedal, J. (1992): Theory of algorithms for unconstrained optimization. *Acta Numerica* 199–242, London: Cambridge University Press.

Oja, E. (1989): Neural networks, principal components and subspace,. *J. Neural Systems*, 1, pp. 61-68.

Oja, E. (1997): The nonlinear PCA learning rule in Independent Component Analysis. *Neurocomputing*, vol. 17, pp. 25–45.

Penny, W.D. and Roberts, S.J. (2001): Mixtures of Independent Component Analysers. *Artificial Neural Networks - ICANN2001.* International Conference on Artificial Neural Networks, pp. 527–534.

Pearlmutter, B. and Parra, L. (1996): A context-sensitive generalization of ICA. in *ICONIP '96*, pp. 151–157.

Papoulis, A. (1992): Probability, Random Variables, and Stochastic Processes. McGraw-Hill, 3rd edition.

Pham, D.-T., Garrat, P., and Jutten, c. (1992): Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771-774.

Pham, D.T. and Garat, P. (1997): Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. on Signal Processing*, 45, 7, 1712-1725.

Schmidhuber, J., Eldracher, M. and Foltin, B., (1996): Semilinear predictability minimization produces well-known feature detectors. *Neural Computation*, 8:773-786.

Shalvi, O. and Weinstein, E., (1993): Super-exponential methods for blind deconvolution. *IEEE Trans. on Information Theory*, 39(2):504:519.

Tipping, M.E. and Bishop, C.M. (1997): Probabilistic principal component analysis. *J. Royal*

Statistical Society B, 61, Part 3, pp. 611-622

Tipping, M.E. and Bishop, C.M. (1999): Mixtures of Probabilistic principal component analysers. *Neural Computation* **11(2)**, pp. 443-482.

Xu, L. and Yuille, A. (1995): Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. on Neural Networks*, **6**, 131-143.