UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Robust Mixed-Effects Segmented Regression Models and Independent Component Analysis

Permalink

https://escholarship.org/uc/item/15w481gs

Author

Zhou, Xiaoyang

Publication Date

2017

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Robust Mixed-Effects Segmented Regression Models and Independent Component Analysis

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Xiaoyang Zhou

December 2017

Dissertation Committee:

Dr. Weixin Yao, Chairperson
Dr. Shujie Ma
Dr. Gregory Palardy

Copyright by
Xiaoyang Zhou
2017

The Dissertation of Xiaoyang Zhou is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to thank my advisor, Dr. Weixin Yao, for his support and excellent guidance in developing this dissertation. Without his help, it would not be possible for me to complete my doctoral study.

Additionally, I am indebted to other members of my dissertation committee, Dr. Shujie Ma and Dr. Gregory Palardy. They gave me valuable inspirations of my study at UCR. I also want to express my gratitude towards all the professors who taught me during my graduate study.

I would like to thank my parents for their unconditional support and love. I am especially grateful to my boyfriend, Yizhou Wang, who has been a constant source of support and encourage.

To my family for all the support.

ABSTRACT OF THE DISSERTATION

Robust Mixed-Effects Segmented Regression Models and Independent Component Analysis

by

Xiaoyang Zhou

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, December 2017
Dr. Weixin Yao, Chairperson

In Chapter 2, Renewable energy production has been surging in the United States and around the world in recent years. To mitigate increasing renewable generation uncertainty and intermittency, proactive demand response algorithms and programs are proposed and developed to further improve utilization of load flexibility and increase power system operation efficiency. One of the biggest challenges to efficient control and operation of demand response resources is how to accurately forecast the load impact from demand response resources. In Chapter 2, we propose a mixed-effect segmented regression model and its robust estimate for forecasting the the load impact of demand response resources in Southern California by combining the ideas of random effect regression model, segmented regression model, and trimmed likelihood estimation. Since the log-likelihood of the new model is not differentiable at breakpoints, we propose a new backfitting algorithm to estimate the unknown parameters of the new model. The estimation performance and predictive power of the new model have been demonstrated with both simulation studies and real data application.

In Chapter 3, a new data analysis tool called Fisher Discriminant Information Matrix (FDIM) is developed to find best directions that separate two densities via a simple eigen-analysis. Based on FDIM, we propose a new estimation procedure for Independent Component Analysis (ICA). The new ICA algorithm can recover the independent components via a simple eigenanalysis of the new defined information matrix. Different from existing ICA algorithms, the new method can also detect whether there is any "uninteresting" Gaussian component in the original signal. In addition, the new method can rank the recovered signals in terms of their density information. To estimate the FDIM, we propose both a kernel density estimation and Gaussian mixture model estimation methods to approximate the unknown density, and utilize the density square transformation to avoid the numerical integrations and reduce the computation cost. We demonstrate the performance of the proposed ICA algorithms using the simulation studies and four real data applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The dissertation contains two main chapters. In Chapter 2, I focus on a Robust Mixed-effect Segmented model. This model is originated from a practical problem in electric industry. The renewable energy sector has experienced exponential growth in the past five to ten years. The global annual growth rates of solar photovoltaic and wind energy are 42% and 17% from 2010 through 2015 Adib *et al.* (2016). The renewable penetration level in certain parts of the world are much higher than the global average penetration level. For example, the renewable energy penetration level in California has reached 30% in 2017. The recently passed California Senate Bill 350 will further boost renewable penetration level to 50% by 2030. To mitigate increasing renewable generation uncertainty and intermittency, demand response resources are in critical need. In the past ten years, traditional and passive price-based and incentive based demand response programs have been implemented throughout United States. In recent years, proactive demand response algorithms and programs are proposed and developed to further improve utilization of load flexibility and

dispatchability. Accurate load impact forecasts are needed to effectively leverage the load flexibility from the demand response resources. The load impact from a demand response resource is defined as the difference between load baselines and metered load when demand response event is triggered. In practice it is very challenging to develop a good estimation of the load baseline which represents the electric load that would have occurred without demand response event.

A good baseline estimation methodology should represent an appropriate tradeoff between simplicity and accuracy. The existing baseline methodology can be categorized into two types. In Type-I baseline methodology, the baseline is estimated by using similar day-based algorithm which depends on historical interval meter data and similarity metrics such as weather and calendar data. Simplicity is the biggest advantage of Type-I baseline method (Wei *et al.* , 2016; Yu *et al.* , 2015). In Type-II baseline methodology, more sophisticated statistical methods are adopted to estimate and forecast the baseline electricity consumption. Typically, Type-II baseline method yields better forecasting accuracy. Most of the existing Type-II baseline method is based on multiple linear regressions.

The proposed mixed effects segmented regression model belong to Type-II baseline methodology and is motivated by forecasting the hourly electric load in Southern California area. The hourly electric consumption data are aggregated to 52 220 kV transformer banks from 12/31/2012 to 11/1/2013 in Southern California Edison's service territory. One commonly used method for electric power demand prediction at each hour is to use a multiple linear regression with *hour* as a categorical variable and weather data as continuous covariates. Note that the electric consumption data are essentially longitudinal/panel data.

The electric load data exhibits very strong spatio-temporal dependencies Yu & Jie (2017). In order to incorporate the correlation among observations from each transformer banks, we propose to use the *random effects regression model* (Laird & Ware, 1982a). An alternative model for hour is to include it as a linear predictor. However, it is expected that the linear effect of hour on electric demand does not hold in the whole range of hour. To this end, we propose to model the hour effect by a *segmented regression model* (Feder, 1975a), which can be considered as a comprise between modeling hour as a global linear predictor and modeling hour as a categorial variable. The nonlinear relationship with breakpoints are said to be piece-wised, segmented, broken-line or multi-phased. The breakpoints are also called change-points, transition-points or switch-points in some applications. Using the segmented regression model for hour, the hour's effect on the electric consumption changes continuously across the time and we can borrow the information from other hours when estimating the hour's effect. The estimated breakpoints can also tell us how the hour's linear effect changes across different areas. Segmented regression have been widely used in many areas. In medication area, this method is a powerful statistical tool for estimating intervention effects of interrupted time series studies (Wagner *et al.* , 2002a). Also, segmented regression is used to identify the changes in the recent trend of cancer mortality and incidence data analysis (Kim *et al.* , 2000a). In ecology area, segmented regression is a widely used statistical tool to model ecological thresholds (Toms & Lesperance, 2003a). For the geometric purpose, segmented regression statistically models the trends in groundwater levels (Shao & Campbell, 2002a).

Note that it is not trivial to compute the maximum likelihood estimate (MLE) for

the new model since the log-likelihood of the new model is not differentiable at breakpoints. Many standard computational algorithms, such as Newton-Raphson algorithm, can not be used directly. In this dissertation, we propose a backfitting algorithm to combine the segmented regression estimation method proposed by Muggeo (2003) and the mixed effect regression estimation method proposed by Bates (2011) to maximize the non-differentiable log-likelihood of the new mixed effects segmented regression model. Note that the MLE is sensitive to outliers, which is the case of our electric consumption data collected in Southern California area. We further propose a robust estimation procedure for the new model by extending the idea of the *least trimmed squares* (LTS) estimate. The simulation study demonstrate the effectiveness of the proposed estimation procedures. The LTS also provides much better prediction performance than the standard MLE for the testing data when forecasting the hourly electric power demand in Southern California area.

The rest of the Chapter 2 is organized as follows. Section 2.1 introduces the new mixed-effects segmented model and describes the proposed estimation algorithms. Section 2.2 illustrates the finite sample performance of the proposed estimation method using a simulation study. In Section 2.3 we apply the new model to forecast the hourly electric power demand in Southern California area.

The other part of dissertation discusses the independent component analysis via density information matrix. Independent Component Analysis (ICA) is a widely used unsupervised machine learning method. Usually, people exploit ICA algorithm to solve Blind Source Separation (BSS) problem. The BSS problem is an inductive inference problem which relies on limited available information to infer the most probable solutions (Naik &

Kumar, 2011). This problem is prevalent in many areas such as neuroscience, face recognition and audio signal processing. Generally, the BSS problem contains three components; original source, mixing matrix and mixed signals. The existing algorithms solving BSS problem all confront the same problem that the parameters of mixing or filtering process are unknown.

ICA is a powerful tool for solving BSS problem since it only assumes the independence and nonGaussianity of original source. The concept of ICA is firstly introduced as a BSS method by Jutten & Herault (1991) using neuro-mimetic architecture. The ICA algorithm is later applied to recover the independent components from the linear mixture of statistically independent sources through different optimizing criteria (Comon, 1994). Since then, ICA algorithm is defined as a method to reveal the hidden factors of underlying random variables, measurements and signals. The existing ICA algorithms are usually generated from two ICA assumptions, mutual independence and nonGaussianity. Based on the first assumption, Comon (1994) proposed an ICA algorithm using Edgeworth expansion of Kullback-Leibler divergence. Another popular ICA algorithm, Infomax (Bell & Sejnowski, 1995), was also derived from independence assumption utilizing mutual information as the objective function. Please also see, for example, Amari *et al.* (1996) and Lee *et al.* (1999) for some extensions of Infomax. Many other ICA algorithms rely on the non-Gaussianity assumption. For example, Hyvärinen *et al.* (2004) proposed the FastICA algorithm with a fixed-point iteration to find the maximum nonGaussianity of the objective function. The Joint Approximate Diagonalization of Eigenmatrices (JADE) algorithm (Cardoso & Souloumiac, 1993) is constructed via fourth-order cumulants array with kurto-

sis function. The flexible ICA algorithm is generated by the Gaussian exponent based on estimated kurtosis of unmixing matrix (Choi *et al.* , 2000).

In this dissertation, we develop a novel and computationally fast ICA algorithm based on a simple eigen-decomposition of the newly introduced Fisher discriminant information matrix (FDIM). Different from existing ICA algorithms, the new method can also detect whether there is any "uninteresting" Gaussian component in the original sources. In addition, the new method can rank the recovered signal in terms of their density information. When estimating the FDIM, we propose both a kernel density estimation and Gaussian mixture model estimation methods to estimate the unknown density, and utilize the density square transformation to avoid the numerical integrations and reduce the computation cost. The simulation study and real data applications demonstrate the superior or comparable performance of the new ICA algorithm compared to some existing methods.

The remainder of Chapter is organized as follows. A general description of ICA model with its underlying assumptions is discussed in Section 3.1. Section 3.2 introduces the density information matrix and its application to ICA. Section 3.3 illustrates the performance of the new ICA algorithm using simulation study. In Section 3.4, we apply the proposed ICA algorithm to four real data examples.

Chapter 2

A Robust Mixed Effects Segmented Regression Model for Forecasting Electric Power Demand

2.1 Model

Given a random sample $\{y_{ij}, \mathbf{x}_{ij}, \mathbf{s}_{ij}, z_{ij}, i = 1, \ldots, n, j = 1, \ldots, n_i\}$, where n is the number of subjects and n_i is the number of observations collected for ith subject, the proposed mixed effects segmented regression model can be written as

$$y_{ij} = \mathbf{x}_{ij}^T \boldsymbol{\alpha} + \mathbf{s}_{ij}^T \boldsymbol{\gamma}_i + \beta_0 z_{ij} + \sum_{k=1}^{l} \beta_k (z_{ij} - \varphi_k)_+ + \varepsilon_{ij}, \tag{2.1}$$

where y_{ij} is the response, $\mathbf{x}_{ij}$ is the p dimension fixed-effect covariates, $\mathbf{s}_{ij}$ is the q dimensional random-effect covariates, z_{ij} is the breakpoint variable with breakpoints $\{\varphi_k, k = 1, \ldots, l\}$, t_+ equals to t if $t \geq 0$ and 0 otherwise, $\boldsymbol{\gamma}_i \sim N_q(0, \Sigma_\gamma)$, and $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{in_i}) \sim$

$N_{n_i}(0, \Sigma_\varepsilon)$. In this paper, we assume that $\Sigma_\varepsilon = \sigma^2 \boldsymbol{I}_{n_i}$. The new model (2.1) consists of three parts: multiple linear regression $\mathbf{x}_{ij}^T \boldsymbol{\alpha}$, random-effects $\mathbf{s}_{ij}^T \boldsymbol{\gamma}_i$, and segmented regression $\beta_0 z_{ij} + \sum_{k=1}^{l} \beta_k (z_{ij} - \varphi_k)_+$, which models the heterogeneous linear effect of z_{ij} on y_{ij} across different areas of z. β_k measures the difference of slopes (linear effect of z_{ij} on y_{ij}) before and after the breakpoint φ_k.

Let $\mathbf{y}_i = (y_{i1}, \ldots, y_{in_i})^T, \mathbf{X}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})^T, \mathbf{S}_i = (\mathbf{s}_{i1}, \ldots, \mathbf{s}_{in_i})^T$ and $\mathbf{Z}_i = (\mathbf{z}_{i1}^*, \ldots, \mathbf{z}_{in_i}^*)^T$, where $\mathbf{z}_{ij}^* = (z_{ij}, (z_{ij} - \varphi_1)_+, \ldots, (z_{ij} - \varphi_l)_+)^T$. Then (2.1) can be rewritten in matrix format as

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\alpha} + \mathbf{S}_i \boldsymbol{\gamma}_i + \mathbf{Z}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i, \tag{2.2}$$

where $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_l)^T$. Based on (2.2), $\mathrm{E}(\mathbf{y}_i \mid \mathbf{X}_i, \mathbf{z}_i, \mathbf{S}_i) = \mathbf{X}_i \boldsymbol{\alpha} + \mathbf{Z}_i \boldsymbol{\beta}$ and $\mathrm{var}(\mathbf{y}_i \mid \mathbf{X}_i, \mathbf{z}_i, \mathbf{S}_i) = \mathbf{S}_i \Sigma_\gamma \mathbf{S}_i^T + \sigma_\varepsilon^2 \boldsymbol{I}_{n_i} \triangleq \Sigma_i$. Therefore, the random effects $\boldsymbol{\gamma}_i$ make the observations within each subject correlated.

The log-likelihood function of $\{y_{ij}, \mathbf{x}_{ij}, \mathbf{s}_{ij}, z_{ij}, i = 1, \ldots, n, j = 1, \ldots, n_i\}$ is

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log[(2\pi|\Sigma_i|)^{-1/2} \exp\{-\frac{1}{2}(\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \boldsymbol{\beta})^T \Sigma_i^{-1} (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\alpha} - \mathbf{Z}_i \boldsymbol{\beta})\}], \tag{2.3}$$

where $\boldsymbol{\theta}$ collects all unknown parameters. Unlike the traditional mixed effects model, maximizing (2.3) is not trivial since it is not differentiable at φ_k. We propose a backfitting algorithm to maximize (2.3) by alternately updating the segmented regression part and the linear mixed effects part when fixing the other. Next we discuss in detail how to perform such two estimation procedures.

2.1.1 Estimating breakpoints

If the breakpoints are fixed, the model is usually a linear model. Then, the estimation is simple without any problems of estimation and inference. In this paper, we are mainly interested in the situation where the number of breakpoints is known but their locations are unknown. Breakpoints and slopes in segmented regression can be estimated through many ways such as regression spline as well as Bayesian MCMC methods (Gössl & Küchenhoff, 2001a; Hastie & Tibshirani, 1990a). We will extend the linearization technique proposed by Muggeo (2003) to our new model (2.1) due to its simplicity of computation. According to the definition of breakpoint, the log-likelihood is not differentiable at φ_k. The breakpoint estimation can be performed via a first-order Taylor expansion of $(z_{ij} - \varphi_k)_+$ around an initial value $\varphi_k^{(0)}$,

$$(z_{ij} - \varphi_k)_+ = (z_{ij} - \varphi_k^{(0)})_+ + (\varphi_k - \varphi_k^{(0)})(-1)I(z_{ij} > \varphi_k^{(0)}),$$

where $(-1)I(z_{ij} > \varphi_k^{(0)})$ is the first derivative of $(z_{ij} - \varphi_k)_+$ assessed in $\varphi_k^{(0)}$.

Let $v_{ij} = ((-1)I(z_{ij} > \varphi_1^{(0)}), \ldots, (-1)I(z_{ij} > \varphi_l^{(0)}))^T$, $\tilde{\mathbf{z}}_{ij} = (z_{ij}, (z_{ij} - \varphi_1^{(0)})_+, \ldots, (z_{ij} - \varphi_l^{(0)})_+)^T$, and $\delta_k = \beta_k(\varphi_k - \varphi_k^{(0)})$. Define $\mathbf{V}_i = (v_{i1}, \ldots, v_{in_i})^T$, $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_l)^T$ and $\tilde{\mathbf{Z}}_i = (\tilde{\mathbf{z}}_{i1}, \ldots, \tilde{\mathbf{z}}_{in_i})^T$. Given the estimate $\{\hat{\boldsymbol{\alpha}}, \hat{\Sigma}_i\}$, plugging them into the model (2.2), we have

$$\tilde{\mathbf{y}}_i = \tilde{\mathbf{Z}}_i\boldsymbol{\beta} + \mathbf{V}_i\boldsymbol{\delta} + \tilde{\boldsymbol{\varepsilon}}_i, \tag{2.4}$$

where $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{X}_i\hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\varepsilon}}_i \sim N_{n_i}(0, \hat{\Sigma}_i)$. $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ in (2.4) can be easily found by weighted least squares estimate. Note that $\varphi_k = (\delta_k/\beta_k) + \varphi_k^0$. The iterative algorithm will terminate at $\delta_k = 0$. The algorithm to estimate the breakpoints, given the estimate $\{\hat{\boldsymbol{\alpha}}, \hat{\Sigma}_i\}$, is summarized as follows:

Algorithm A	Segmented regression estimation
1	Set initial value of all breakpoints $\varphi_k^{(0)}$, for $k = 1, ..., l$ and calculate the variable $\tilde{\mathbf{Z}}_i$ and the variable $\mathbf{V}_i$
2	Fit the regression model of $\tilde{\mathbf{y}}_i$ on $\tilde{\mathbf{Z}}_i$ and $\mathbf{V}_i$ using the model (2.4).
3	Update the breakpoint with equation $\varphi_k^{(s+1)} = (\delta_k^{(s)}/\beta_k^{(s)}) + \varphi_k^{(s)}$, where $\varphi_k^{(s)}$ is the estimate of φ_k at sth iteration.
4	Repeat step 2-3 until converge.

2.1.2 Estimating covariance matrix of random effects

In this section, we discuss how to maximize (2.3) given the estimate $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\varphi}}$, where $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_l)^T$. Let $\hat{\mathbf{Z}}_i$ be the estimate of $\mathbf{Z}_i$ after replacing φ_k by $\hat{\varphi}_k$. Plugging in the estimate $\{\hat{\mathbf{Z}}_i, \hat{\boldsymbol{\beta}}\}$ into the model (2.1), we have

$$\mathbf{y}_i^* = \mathbf{X}_i\boldsymbol{\alpha} + \mathbf{S}_i\boldsymbol{\gamma}_i + \boldsymbol{\varepsilon}_i, \tag{2.5}$$

where $\mathbf{y}_i^* = \mathbf{y}_i - \hat{\mathbf{Z}}_i\hat{\boldsymbol{\beta}}$. Then, the model (2.5) is reduced to a traditional mixed-effects model. The parameters are optimized the objective function, maximum likelihood function. Because the objective function must be evaluated at many different values of the model parameters during the optimization process, we employ the penalized, weighted least square (PWLS) method to determine the solution (Bates, 2011). If the dimension of solution is tremendous, the solution must be evaluated with repeatedly optimization problem. Then, we can choose PWLS to determine parameter estimates with the Cholesky decomposition.

In model (2.5), the variance-covariance matrix Σ_γ of $\boldsymbol{\gamma}$ must be positive definite. It is convenient to transform the matrix in terms of a relative covariance factor, Λ_λ, which

is a $q \times q$ matrix relying on the parameter $\boldsymbol{\lambda}$, such that

$$\Sigma_\gamma = \sigma^2 \Lambda_\lambda \Lambda_\lambda^T$$

$$\boldsymbol{\gamma}_i = \Lambda_\lambda \mathbf{u}_i, \mathbf{u}_i \sim \mathbf{N}_q(\mathbf{0}, \sigma^2 \mathbf{I}_q)$$

Then (2.5) can be written as

$$\mathbf{y}_i^* = \mathbf{X}_i \boldsymbol{\alpha} + \mathbf{S}_i \Lambda_\lambda \mathbf{u}_i + \boldsymbol{\varepsilon}_i. \tag{2.6}$$

Given $\Gamma = (\boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_n^T)^T$, the conditional distribution of $\mathbf{y}^*$ is

$$\mathbf{y}^* \mid \Gamma \sim \mathbf{N}(\mathbf{X}\boldsymbol{\alpha} + \mathbf{S}\boldsymbol{\Lambda}\mathbf{u}, \sigma^2 \mathbf{I}_N), \tag{2.7}$$

where $\mathbf{y}^* = (\mathbf{y}_1^{*T}, \ldots, \mathbf{y}_n^{*T})^T$, $\mathbf{X} = (\mathbf{X}_1^T, \ldots, \mathbf{X}_n^T)^T$, $\mathbf{S} = \text{diag}(\mathbf{S}_1, \ldots, \mathbf{S}_n)^T$, $\boldsymbol{\Lambda} = \text{diag}(\Lambda_\lambda, \ldots, \Lambda_\lambda)$, $\mathbf{u} = (\mathbf{u}_1^T, \ldots, \mathbf{u}_n^T)^T$ and $N = \sum_{i=1}^n n_i$. Since the value of $\mathbf{y}^*$ is observable, the goal of statistical inference is $f(\boldsymbol{\gamma}|\mathbf{y}^*)$, or the linear transformation $f(\mathbf{u}|\mathbf{y}^*)$. The density $f(\mathbf{u}|\mathbf{y}^*)$ is proportional to the product of $f(\mathbf{u})$ and $f(\mathbf{y}^*|\mathbf{u})$. Thus, the unnormalized conditional density $f(\mathbf{u}|\mathbf{y}^*)$ is defined as

$$h(\mathbf{u}|\mathbf{y}^*, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma) = f(\mathbf{y}^*|\mathbf{u}) f(\mathbf{u}) \tag{2.8}$$

with the deviance as

$$\begin{aligned} -2\log(h(\mathbf{u}|\mathbf{y}^*, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma)) &= (N + nq)\log(2\pi\sigma^2) + \frac{\|\mathbf{y}^* - \mathbf{S}\Lambda_\lambda \mathbf{u} - \mathbf{X}\boldsymbol{\alpha}\|^2 + \|\mathbf{u}\|^2}{\sigma^2} \\ &= (N + nq)\log(2\pi\sigma^2) + \frac{d(\mathbf{u}|\mathbf{y}^*, \boldsymbol{\lambda}, \boldsymbol{\alpha})}{\sigma^2} \end{aligned} \tag{2.9}$$

In (2.9), $d(\mathbf{u}|\mathbf{y}^*, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = \|\mathbf{y}^* - \mathbf{S}\Lambda_\lambda \mathbf{u} - \mathbf{X}\boldsymbol{\alpha}\|^2 + \|\mathbf{u}\|^2$ is called the discrepancy function, where $\|\mathbf{y}^* - \mathbf{S}\Lambda_\lambda \mathbf{u} - \mathbf{X}\boldsymbol{\alpha}\|^2$ is the residual sum of squares and the second term, $\|\mathbf{u}\|^2$, is a penalty on the size of $\mathbf{u}$. It is minimized at the conditional mode, $\tilde{\mathbf{u}}(\boldsymbol{\lambda})$, and the

conditional estimate, $\tilde{\boldsymbol{\alpha}}(\boldsymbol{\lambda})$, which are the solutions to the sparse, positive-definite linear system

$$\begin{bmatrix} \Lambda_\lambda^T \mathbf{S}^T \mathbf{S} \Lambda_\lambda + \mathbf{I}_{nq} & \Lambda_\lambda^T \mathbf{S}^T \mathbf{X} \\ \mathbf{X}^T \mathbf{S} \Lambda_\lambda & \mathbf{X}^T \mathbf{X} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}(\boldsymbol{\lambda}) \\ \tilde{\boldsymbol{\alpha}}(\boldsymbol{\lambda}) \end{bmatrix} = \begin{bmatrix} \Lambda_\lambda^T \mathbf{S}^T \mathbf{y} \\ \mathbf{X}^T \mathbf{y} \end{bmatrix}. \tag{2.10}$$

In the process of solving the positive definite linear system (2.10), we introduce Cholesky factor with the form

$$\begin{bmatrix} \mathbf{L_S} & \mathbf{0} \\ \mathbf{L_{XS}} & \mathbf{L_X} \end{bmatrix} \begin{bmatrix} \mathbf{L_S} & \mathbf{0} \\ \mathbf{L_{XS}} & \mathbf{L_X} \end{bmatrix}^T = \begin{bmatrix} \mathbf{P_S} & \mathbf{0} \\ \mathbf{0} & \mathbf{P_X} \end{bmatrix} \begin{bmatrix} \Lambda_\lambda^T \mathbf{S}^T \mathbf{S} \Lambda_\lambda + \mathbf{I}_{nq} & \Lambda_\lambda^T \mathbf{S}^T \mathbf{X} \\ \mathbf{X}^T \mathbf{S} \Lambda_\lambda & \mathbf{X}^T \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{P_S} & \mathbf{0} \\ \mathbf{0} & \mathbf{P_X} \end{bmatrix}^T, \tag{2.11}$$

where $\mathbf{P_S}$ and $\mathbf{P_X}$ are permutation matrices representing a fill-reducing permutation matrix. Substituting (2.11) and $(\tilde{\mathbf{u}}(\lambda), \tilde{\boldsymbol{\alpha}}(\lambda))$ into (2.9), the new version of deviance is

$$-2\log(h(\mathbf{u}|\mathbf{y}^*, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma)) = (N+nq)\log(2\pi\sigma^2) + \frac{\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda}) + \left\| \begin{bmatrix} \mathbf{L_S} & \mathbf{0} \\ \mathbf{L_{XS}} & \mathbf{L_X} \end{bmatrix} \begin{pmatrix} \mathbf{P_S}(\mathbf{u} - \tilde{\mathbf{u}}) \\ \mathbf{P_X}(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}) \end{pmatrix} \right\|^2}{\sigma^2}, \tag{2.12}$$

where $\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda}) = d(\tilde{\mathbf{u}}(\boldsymbol{\lambda})|\mathbf{y}^*, \boldsymbol{\lambda}, \tilde{\boldsymbol{\alpha}}(\boldsymbol{\lambda}))$ is the minimum discrepancy function assuming $\boldsymbol{\lambda}$ is known. Since the integral of a quadratic form is easily evaluated, we integrate (2.12) with respect to random-effects coefficients $\mathbf{u}$. Then, the profile likelihood is,

$$-2\ell(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma|\mathbf{y}^*) = N\log(2\pi\sigma^2) + \log(|\mathbf{L_S}|^2) + \frac{\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda}) + ||\mathbf{L_X^T P_X}(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})||^2}{\sigma^2}. \tag{2.13}$$

Substituting the conditional estimates $\tilde{\boldsymbol{\beta}}(\boldsymbol{\theta})$ and $\tilde{\sigma^2}(\boldsymbol{\theta}) = \tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda})/N$, the profile likelihood is

$$-2\ell(\boldsymbol{\lambda}|\mathbf{y}^*) = \log(|\mathbf{L_S}(\boldsymbol{\lambda})|^2) + N(1 + \log(\frac{2\pi\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda})}{N})). \tag{2.14}$$

Then, the maximum likelihood estimated of $\boldsymbol{\lambda}$ is

$$\hat{\boldsymbol{\lambda}}_{\mathbf{L}} = \arg\min_{\boldsymbol{\lambda}}(\log(|\mathbf{L_S}|^2) + N(1 + \log(\frac{2\pi\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda})}{N}))). \tag{2.15}$$

Given $\boldsymbol{\lambda}$, the equation (2.13) is a ML estimates are

$$\begin{aligned} \hat{\sigma^2}_{\mathbf{L}} &= \frac{\tilde{d}(\mathbf{y}^*, \hat{\boldsymbol{\lambda}}_{\mathbf{L}})}{N} \\ \hat{\boldsymbol{\alpha}}_{\mathbf{L}} &= \tilde{\boldsymbol{\alpha}}(\hat{\boldsymbol{\lambda}}_{\mathbf{L}}). \end{aligned} \tag{2.16}$$

The mixed-effect regression model estimation, given the estimate of segmented regression, can be summarized as follows.

Algorithm B	Random-effects Estimation
1	Set initial covariance factor $\boldsymbol{\lambda}^{(0)}$ and obtain $\Lambda_{\lambda}^{(0)}$.
2	With the current $\boldsymbol{\lambda}^{(s)}$, solve the normal equation for $\tilde{\boldsymbol{\alpha}}^{(s)}$ and $\tilde{\mathbf{u}}^{(s)}$ and then calculate discrepancy function $\tilde{d}(\mathbf{y}^*, \boldsymbol{\lambda}^{(s)})$.
3	Using discrepancy function, calculate $\sigma^{2(s)}_{\mathbf{L}}$ and update the covariance factor paramter $\boldsymbol{\lambda}^{(s+1)}$ via optimization method such as Newton-Raphson method.
4	Repeat 2 - 3 until the algorithm reaching the convergence criterion.

2.1.3 Mixed-effects breakpoint estimation

By combining Algorithm A and Algorithm B, we propose the following backfitting algorithm to maximizing the log-likelihood (2.3) for the model (2.2).

Algorithm	MLE
1	Set initial value of breakpoint $\varphi_k^{(0)}$ and $\boldsymbol{\beta}^{(0)}$.
2	Given current breakpoint values $\varphi_k^{(s)}$ and slopes $\boldsymbol{\beta}^{(s)}$, calculate $\mathbf{y}_i^{*(s)} = \mathbf{y}_i - \hat{\mathbf{Z}}_i^{(s)}\hat{\boldsymbol{\beta}}^{(s)}$.
3	Fit mixed-effect model with Algorithm B to obtain covariance matrix $\Sigma_r^{(s)}$ and the fixed effect regression estimate $\boldsymbol{\alpha}^{(s)}$.
4	Calculate $\tilde{y}_i^{(s)} = \mathbf{y}_i - \mathbf{X}_i\boldsymbol{\alpha}^{(s)}$.
5	Fit segmented regression model with $\tilde{y}_i^{(s)}$ using Algorithm A and update segmented regression parameter estimate to $\boldsymbol{\varphi}^{(s+1)}$ and $\boldsymbol{\beta}^{(s+1)}$.
6	Repeat 2 - 5 until convergence.

2.1.4 Robust mixed-effects breakpoint estimation

It is well known that the MLE is sensitive to outliers and might give misleading results when there are outliers in the data, which is the case for our collected electric power demand data in Southern California area. Please see Section 2.3 for more detail. Next we propose to use the idea of least trimmed squares estimate (Rousseeuw, 1984) to provide a robust estimate of the model (2.1). Given an integer trimming parameter $h \leq N$, the least trimmed squares minimizes the sum of the smallest h squared residuals with objective function

$$\sum_{k=1}^{h} (r^2)_{k:N}, \tag{2.17}$$

where $(r^2)_{1:N} \leq ... \leq (r^2)_{N:N}$ are the ordered squared residuals $\{y_{ij} - \hat{y}_{ij}, i = 1, \ldots, n; j = 1, \ldots, n_i\}$ with $\hat{y}_{ij} = \mathbf{x}_{ij}^T \hat{\boldsymbol{\alpha}} + \mathbf{s}_{ij}^T \hat{\boldsymbol{\gamma}}_i + \hat{\beta}_0 z_{ij} + \sum_{k=1}^{l} \hat{\beta}_k (z_{ij} - \hat{\varphi}_k)_+$. Let $\boldsymbol{\theta}$ collect all the unknown parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\varphi}, \sigma, \boldsymbol{\Sigma}_\gamma\}$ in the model (2.1). The robust mixed-effects breakpoint algorithm based on LTS is described in the following table.

Algorithm	LTS
1	A subsample of size h^* is selected randomly from the data sample and then the model (2.1) is fitted to the that subsample using Algorithm MLE. Let $\boldsymbol{\theta}^{(0)}$ be the initial parameter estimate .
2	Based on current model parameter estimate $\boldsymbol{\theta}^{(s)}$, make prediction of N responses: $\hat{y}_{ij}^{(s)}$ = and calculate the residuals $r_{ij} = y_{ij} - \hat{y}_{ij}$. Rank the squared residuals $\{r_{ij}^2, i = 1, \ldots, n; j = 1, \ldots, n_i\}$ from smallest to largest and select the first h observations that correspond to the smallest h squared residuals.
3	Fit the model (2.1) to the subsample selected in Step 2 using Algorithm MLE and get the model parameter estimate $\boldsymbol{\theta}^{(s+1)}$.
4	Repeat 2 - 3 until convergence.

To increase the chance of finding the global minimum, one might run Algorithm LTS from many random subsamples and choose the solution which has the smallest trimmed squares. Let r be the dimension of $\boldsymbol{\theta}$. The initial sample size h^* can be any small number

larger than r as long as the initial parameter estimate $\boldsymbol{\theta}^{(0)}$ can be computed based on the subsample. The maximum breakpoint (i.e., the smallest fraction of contamination that can cause the estimator to take arbitrary large values) of LTS is 0.5 and is attained when $h = [(N + r + 1)/2]$. If we have the prior that the proportion of outliers is no more than α, we can also set $h = [N(1 - \alpha) + 1]$, where α is called the trimming proportion. In practice, one might also try several α values to evaluate LTS and check how the estimate behaves with different trimming proportions.

2.2 Simulation Study

In this section, we use a simulation study to illustrate the performance of the estimation procedure for the proposed mixed-effect segmented regression model. We generate observations $\{y_{ij}, \mathbf{x}_{ij}, \mathbf{s}_{ij}, z_{ij}, i = 1, \ldots, n, j = 1, \ldots, n_i\}$, from the following model

$$y_{ij} = \alpha_0 + \alpha_1 x_{ij} + \gamma_{i0} + s_{ij}\boldsymbol{\gamma}_{i1} + \beta_0 z_{ij} + \beta_1 (z_{ij} - \varphi_1)_+ + \beta_2 (z_{ij} - \varphi_2)_+ + \varepsilon_{ij}, \qquad (2.18)$$

where $x_{ij} \sim \text{Pois}(10)$, $s_{ij} \sim \text{Uniform}(5, 10)$, z_{ij}'s are n_i arithmetic sequence range from $(0, 20)$, $\varepsilon_{ij} \sim N(0, 0.5)$,

$$\begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{r1}^2 & \rho\sigma_{r1}\sigma_{r2} \\ \rho\sigma_{r1}\sigma_{r2} & \sigma_{r2}^2 \end{pmatrix} \right),$$

with $\sigma_{r1} = \sigma_{r2} = 1, \rho = 0.5$. The other parameters in (2.18) are set to be

α_0 = -2.5	$\beta_0 = 1.5$	$\varphi_1 = 6.67$
$\alpha_1 = 1.5$	$\beta_1 = 1.5$	$\varphi_2 = 13.33$
	β_2 = -2.5	

We consider the following two simulation scenarios: 1) $n = 50, n_i$ is randomly chosen from $(90, 110)$; 2) $n = 200, n_i$ is randomly chosen from $(450, 550)$.

	$\alpha_0 = -2.5$			$\alpha_1 = 1.5$		
MLE	Mean	Median	SD	Mean	Median	SD
Scenario 1	-2.505	-2.508	0.125	1.500	1.499	0.002
Scenario 2	-2.498	-2.497	0.064	1.500	1.500	0.001

Table 2.1: Simulation results for Model 2.18 without outliers. It presents the fixed-effect parameter estimates with Algorithm MLE for both simulation scenarios.

	$\varphi_1 = 6.667$			$\varphi_2 = 13.333$		
MLE	Mean	Median	SD	Mean	Median	SD
Scenario 1	6.667	6.666	0.022	13.334	13.332	0.012
Scenario 2	6.667	6.667	0.006	13.333	13.333	0.003

Table 2.2: Simulation results for Model 2.18 without outliers. It presents the breakpoints estimates with Algorithm MLE for both simulation scenarios.

First, we utilize model (2.18) to simulate dataset without outliers. The model is estimated with the Algorithm MLE. In Table 2.1-2.4, we report the Mean, Median, and Standard Deviation for the estimates of fixed-effects regression parameters, breakpoints, segmented regression parameters, and random-effects covariance matrix, respectively based on 500 replications.

	$\beta_0 = 1.5$			$\beta_1 = 1.5$			$\beta_2 = -2.5$		
MLE	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
Scenario 1	1.499	1.500	0.006	1.499	1.499	0.008	-2.499	-2.499	0.008
Scenario 2	1.500	1.500	0.001	1.500	1.500	0.001	-2.500	-2.500	0.002

Table 2.3: Simulation results for Model 2.18 without outliers. It presents the breakpoint slope estimates with Algorithm MLE for both simulation scenarios.

	$\sigma_{r1} = 1$			$\sigma_{r2} = 1$			$\rho = 0.5$		
MLE	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
Scenario 1	0.969	0.959	0.108	0.978	0.976	0.101	0.504	0.503	0.121
Scenario 2	0.990	0.991	0.050	0999	0.999	0.056	0.499	0.499	0.001

Table 2.4: Simulation results for Model 2.18 without outliers. It presents the random-effect estimates with Algorithm MLE for both simulation scenarios.

From Table 2.1-2.4, we can see that the proposed MLE algorithm performs well when the dataset does not contain any outliers. Also, when the sample size increases, standard deviation of each parameter estimate decreases.

Next, we simulate dataset with outliers based on model (2.18). Model parameters are estimated by both Algorithm MLE and Algorithm LTS. In order to check how robust each estimate is against the outliers, we randomly choose 5% of each simulated data and add 30 to the response Y (the range of Y is $(15, 69)$) and 10 to the value of X (the range of X is $(0, 10)$). When applying LTS, we need to choose the trimming proportion α, which has long been a difficult problem. However, LTS can provide a robust model estimate as long as the proportion of outliers is less than α but with low efficiency if the α is too large. Usually a conservative choice of α is recommended in practice. For our examples, we report the results for both $\alpha = 0.1$ and $\alpha = 0.2$. Note that the results of LTS will be better if $\alpha = 0.05$ is used.

In Table 2.5-2.8, we report the simulation results for the estimates of fixed-effects regression parameters, breakpoints, segmented regression parameters, and random-effects covariance matrix, respectively based on 200 replications. From the tables, we can see that the standard MLE fails to provide reasonable estimates of fixed-effects regression parameters

	$\alpha_0 = -2.5$			$\alpha_1 = 1.5$		
Scenario 1	Mean	Median	SD	Mean	Median	SD
MLE	3.332	3.334	0.663	1.019	1.017	0.026
LTS $\alpha = 0.2$	-2.535	-2.539	0.283	1.500	1.500	0.004
LTS $\alpha = 0.1$	-2.521	-2.522	0.210	1.500	1.500	0.003
Scenario 2	Mean	Median	SD	Mean	Median	SD
MLE	3.310	3.314	0.180	1.017	1.017	0.006
LTS $\alpha = 0.2$	-2.502	-2.507	0.130	1.500	1.500	0.001
LTS $\alpha = 0.1$	-2.502	-2.505	0.089	1.500	1.500	0.001

Table 2.5: Simulation results for Model 2.18 with outliers. The table presents the fixed-effect estimates for both simulation scenarios via Algorithm MLE and Algorithm LTS with different α level.

and random-effects covariance matrix when the data contains 5% outliers while LTS can provide reasonable estimates for all parameters with both $\alpha = 0.1$ and $\alpha = 0.2$.

2.3 Real Data Analysis

In this Section, we illustrate the application of the proposed mixed-effects segmented regression model to forecast the electric load in Southern California.

2.3.1 Data

The electric consumption data are aggregated to 52 220 kV transformer banks from 12/31/2012 to 11/1/2013 in Southern California Edison's service territory. The objective is to build a prediction model for the total residential customer electricity consumption at each 220 kV transformer bank on weekdays.

	$\varphi_1 = 6.667$			$\varphi_2 = 13.333$		
Scenario 1	Mean	Median	SD	Mean	Median	SD
MLE	6.647	6.679	0.546	13.322	13.317	0.324
LTS $\alpha = 0.2$	6.670	6.38	0.755	13.351	13.342	0.519
LTS $\alpha = 0.1$	6.670	6.677	0.415	13.323	13.332	0.283
Scenario 2	Mean	Median	SD	Mean	Median	SD
MLE	6.671	6.673	0.172	13.333	13.337	0.098
LTS $\alpha = 0.2$	6.670	6.685	0.293	13.325	13.330	0.166
LTS $\alpha = 0.1$	6.670	6.677	0.166	13.328	13.330	0.094

Table 2.6: Simulation results for Model 2.18 with outliers. The table presents the breakpoint estimates for both simulation scenarios via Algorithm MLE and Algorithm LTS with different α level.

	$\beta_0 = 1.5$			$\beta_1 = 1.5$			$\beta_2 = -2.5$		
Scenario 1	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
MLE	1.493	1.494	0.109	1.521	1.518	0.154	-2.516	-2.522	0.163
LTS $\alpha = 0.2$	1.501	1.502	0.123	1.510	1.500	0.166	-2.519	-2.505	0.168
LTS $\alpha = 0.1$	1.506	1.503	0.070	1.505	1.501	0.0083	-2.509	-2.507	0.071
Scenario 2	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
MLE	1.500	1.502	0.035	1.500	1.499	0.040	-2.500	-2.502	0.042
LTS $\alpha = 0.2$	1.500	1.502	0.054	1.502	1.500	0.051	-2.499	-2.502	0.054
LTS $\alpha = 0.1$	1.495	1.497	0.022	1.506	1.505	0.026	-2.506	-2.499	0.028

Table 2.7: Simulation results for Model 2.18 with outliers. The table presents the breakpoint slope estimates for both simulation scenarios via Algorithm MLE and Algorithm LTS with different α level.

	$\sigma_{r1} = 1$			$\sigma_{r2} = 1$			$\rho = 0.5$		
Scenario 1	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
MLE	0.518	0.379	0.582	1.019	1.026	0.117	0.843	0.999	0.612
LTS $\alpha = 0.2$	1.000	0.993	0.097	0.993	1.003	0.097	0.509	0.510	0.115
LTS $\alpha = 0.1$	0.992	0.998	0.111	0.994	0.999	0.097	0.511	0.519	0.107
Scenario 2	Mean	Med	SD	Mean	Med	SD	Mean	Med	SD
MLE	0.769	0.897	0.435	1.013	1.015	0.056	0.619	0.592	0.272
LTS $\alpha = 0.2$	0.992	0.990	0.048	0.998	0.998	0.047	0.495	0.496	0.050
LTS $\alpha = 0.1$	0.992	0.989	0.048	0.998	0.997	0.047	0.496	0.496	0.049

Table 2.8: Simulation results for Model 2.18 with outliers. The table presents the random-effect estimates for both simulation scenarios via Algorithm MLE and Algorithm LTS with different α level.

The response variable is customers' hourly electricity consumption, Usage_t, recorded through the smart meters. Usage_t is an aggregated variable at the transformer bank level. We use the following transformation to make it comparative among 52 subgroups

$$\text{logUsage}_{per,t} = \log(\text{Usage}_t/\text{Total AC tonnage}). \tag{2.19}$$

In equation (2.19), the transformed response variable is derived through dividing the aggregated usage by total air conditioning tonnage of residential customer in the air conditioning cycling program under the transformer bank and applying the log-transformation. The new response variable indicates electricity consumption level per unit of air conditioning tonnage. We collect several explanatory variables to perform the prediction listed in Table 2.3.1. Two-day lagged electricity consumption variable is selected rather than one-day lagged variable because the demand response resources load impact estimates need to be submitted to the independent system operator one day before the actual operations. The weather average temperature and humidity are included because they are highly correlated

Notation	Explanatory Variable
$\log(\text{Usage}_{per,t-48})$	two-day lagged electricity consumption
Temperature_t	Daily average ambient temperature
Humidity_t	Humidity of the day
Hour_t	Hour/Time of the day
$\text{AC tonnage}_{per,t}$	Duty cycle percentage
Total Ac tonnage	Total AC tonnage under the same transformer bank
A Bank	The indicator variable of transformer bank

Table 2.9: Seven explanatory variables in real data application. Variable *A Bank* is the random-effect variable. Variable *Hour* is the segmented variable.

with electricity consumption. The duty cycle option variable indicates the percentage of participation rate of air conditioning load in the program and has strong influence over the load impact for air conditioning cycling demand response program.

2.3.2 Model and Result

We apply the proposed mixed-effect segmented regression model to forecast the electricity consumption. Figure 2.1 displays the hourly trend for average electric consumption. Obviously, the curve indicates three segments with two breakpoints. We also tried the model with three breakpoints (one more breakpoint in the middle area) but BIC for two breakpoints is smaller. Also, the observations collected over time within the same transformer bank are correlated. Ignoring such correlation by fixed effect model would result in inefficient estimates and lose prediction power. In order to incorporate such correction, the transformer bank is treated as random-effects. Using a random-effects model can also drastically reduce the number of unknown parameters in the model and thus has more efficient parameter estimates. In addition, two-way and three-way interactions are considered

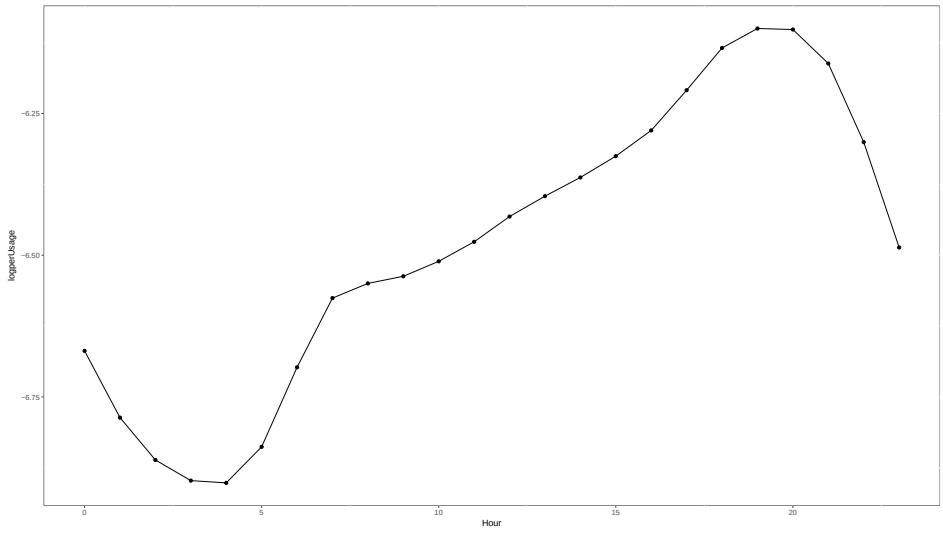

Figure 2.1: The plot shows the trend between average hourly electric consumption *Usage* with variable *Hour* for all *A Bank*. This plot shows two breakpoints. The first breakpoint locates between 2am and 3am. The second breakpoints locates between 6pm and 8pm.

as potential explanatory variables. In order to further simplify the model, stepwise selection method is applied to simplify the model. The final selected mixed-effects segmented regression model is shown in (2.20).

$$
\begin{aligned}
\log(\text{Usage}_{per,t}) =& \text{A Bank} + \text{Hour}_t + (\text{Hour}_t - \varphi^1)_+ + (\text{Hour}_t - \varphi^2)_+ \\
&+ \text{Temperature}_t + \text{Humidity}_t + \text{AC tonnage}_{per,t} \\
&+ \log(\text{Usage}_{per,t-48}) \\
&+ \left[(\text{Hour} + (\text{Hour}_t - \varphi^1)_+ + (\text{Hour}_t - \varphi^2)_+\right] \times \text{Temperature}_t \\
&+ \left[(\text{Hour} + (\text{Hour}_t - \varphi^2)_+\right] \times \text{Humidity}_t \\
&+ \left[(\text{Hour} + (\text{Hour}_t - \varphi^1)_+ + (\text{Hour}_t - \varphi^2)_+\right] \times \text{AC tonnage}_{per,t} \\
&+ [\text{Temperature}_t + \text{Humidity}_t] \times \text{AC tonnage}_{per,t} \\
&+ \text{Temperature}_t \times \text{Humidity}_t,
\end{aligned}
\tag{2.20}
$$

where A Bank $\sim N(0, \sigma^2_{ABank}\boldsymbol{I})$. We apply both MLE and LTS algorithm to estimate the model and compare their forecasting performance. Since the true proportion of outliers is unknown, we choose three proportions $\alpha = 0.15, 0.10, 0.05$ for LTS to fit the model (2.20). In electric industry, the popular performance evaluation indexes are mean absolute percentage error (MAPE) and mean absolute percentage error (RMSE) with the formula

$$
\text{MAPE} = \frac{1}{N}\sum \frac{|y_{ij} - \hat{y}_{ij}|}{y_{ij}},
$$

$$
\text{RMSE} = \sqrt{\frac{\sum (y_{ij} - \hat{y}_{ij})^2}{N}}.
$$

For better comparison, we also present the model forecasting results with quantiles. The electricity consumption data in the last 10 observed weekdays (18720 observations) are chosen as testing sample.

From Table 2.10 and 2.11, each evaluation criterion reaches the lowest value when

Performance	MAPE	25% APE	50% APE	75% APE
MLE	13.94%	4.55%	8.48%	13.66%
LTS $\alpha = 0.05$	11.08%	2.78%	5.45%	9.37%
LTS $\alpha = 0.1$	10.75%	2.46%	4.95%	8.77%
LTS $\alpha = 0.15$	10.88%	2.55%	5.10%	9.01%

Table 2.10: Prediction results are evaluated by Absolute Percentage Error for the last 10-days in October 2013. Algorithm MLE is compared with Algorithm LTS at different α level.

Performance	RMSE	25% RSE	50% RSE	75% RSE
MLE	672.88	5.78	42.19	164.08
LTS $\alpha = 0.05$	449.68	3.98	27.01	97.45
LTS $\alpha = 0.1$	414.42	3.65	24.73	86.33
LTS $\alpha = 0.15$	420.00	4.75	25.26	88.84

Table 2.11: Prediction results are evaluated by Root Square Error for the last 10-days in October 2013. Algorithm MLE is compared with Algorithm LTS at different α level.

breakpoint	φ_1	φ_2
MLE	2.64	20.47
LTS $\alpha = 0.05$	2.27	20.47
LTS $\alpha = 0.1$	2.27	20.77
LTS $\alpha = 0.15$	2.27	20.47

Table 2.12: Breakpoints estimation for electric power demand dataset via Algorithm MLE and Algorithm LTS at different α level.

$\alpha = 0.1$ and is much smaller than those of MLE. The breakpoint estimates shown in Table 2.12 confirm the locations of breakpoints plotted in Figure 2.1. Table 2.13 displays the fixed-effects and breakpoints slope estimates and Table 2.14 shows the variances of random-effects and the error term for LTS with $\alpha = 0.1$.

According to Table 2.13, all the parameters are significant at level $\alpha = 0.05$. The variable *Hour* and its breakpoints have both positive and negative slopes and the signs match the plot in Figure 2.1. Also, there is a positive relationship between AC *tonnage* and electric power demand *Usage*.

Parameter	Estimate	p-value
Intercept	1.063	< 0.0001
Hour_t	2.100e-02	0.0003
$(\text{Hour}_t - \varphi_1)_+$	-4.713e-02	< 0.0001
$(\text{Hour}_t - \varphi_2)_+$	6.665e-02	< 0.0001
Temperature_t	-2.198e-02	< 0.0001
Humidity_t	2.556e-02	0.0010
$\text{AC tonnage}_{per,t}$	8.838e-01	< 0.0001
$\log(\text{Usage}_{per,t-48})$	-2.223e-00	< 0.0001
$\text{Hour}_t \times \text{Humidity}_t$	-2.493e-05	< 0.0001
$(\text{Hour}_t - \varphi_2)_+ \times \text{Humidity}_t$	2.017e-04	< 0.0001
$\text{Hour}_t \times \text{Temperature}_t$	-8.165e-04	< 0.0001
$(\text{Hour}_t - \varphi_1)_+ \times \text{Temperature}_t$	1.018e-03	< 0.0001
$(\text{Hour}_t - \varphi_2)_+ \times \text{Temperature}_t$	-1.606e-03	< 0.0001
$\text{Hour}_t \times \text{AC tonnage}_t$	2.012e-02	0.0002
$\text{Temperature}_t \times \text{Humidity}_t$	-6.684e-05	< 0.0001
$\text{Temperature}_t \times \text{AC tonnage}_{per,t}$	2.763e-02	< 0.0001
$\text{Humidity}_t \times \text{AC tonnage}_{per,t}$	1.926e-03	0.0124

Table 2.13: Parameter estimation for electric power demand dataset are evaluated with Algorithm LTS method at $\alpha = 0.1$. All the parameter estimates are significant at significance level 0.05.

Groups	Variance	Std.Dev
A Bank	0.0015	0.0392
Error	0.0052	0.0718

Table 2.14: Random-effects estimation for electric power demand dataset are evaluated with Algorithm LTS method at $\alpha = 0.1$. The variance and standard deviation estimates stay within a reasonable range.

Chapter 3

Fisher Discriminant Information Matrix and Its Application to Independent Component Analysis

3.1 Background about ICA model

A general ICA model contains three components, which are original source $\mathbf{s}(t)$, mixing matrix $\mathbf{A}$ and mixed signals $\mathbf{x}(t)$. Suppose we have p statistically independent signals, $s_i(t), i = 1, 2, ..., p$, which are not observable. We assume that each signal is a realization of a certain distribution at a time point t. Also, suppose p sensors are installed for receiving signals, denoted by $x_i(t), i = 1, 2, ..., p$, from sources. Without loss of generality, we assume that both the source and the receiving signal are centered at zero. Let $\mathbf{x}(t) = (x_1(t), \ldots, x_p(t))^T$ and $\mathbf{s}(t) = (s_1(t), \ldots, s_p(t))^T$. Thus, a simple matrix multiplication could

explain the relationship between $\mathbf{x}(t)$ and $\mathbf{s}(t)$ as follows

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{3.1}$$

where $\mathbf{A}$ is an unknown square mixing matrix, $\mathbf{x}(t)$ and $\mathbf{s}(t)$ are $p \times 1$ vectors storing the mixed signals and original sources, respectively. The ICA algorithm aims to estimate the mixing matrix $\mathbf{A}$ based on the information only from $\mathbf{x}(t)$. ICA model usually assumes that the mixing matrix $\mathbf{A}$ is an invertible square matrix. Then, the original source $\mathbf{s}(t)$ can be easily recovered with $\mathbf{s}(t) = \mathbf{A}^{-1}\mathbf{x}(t)$. Most ICA algorithms define an "unmixing" matrix $\mathbf{W}$ to recover the original sources using

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t). \tag{3.2}$$

Here, $\mathbf{W}$ can be considered as an estimate of $\mathbf{A}^{-1}$.

The ICA problem has two main assumptions of the original sources, independence and non-Gaussianity. The independence assumption requires that all the original sources are mutually independent. The non-Gaussianity assumption demands that at most one original source follows Gaussian distribution.

From model (3.2), the original sources are recovered by $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$. Let y denote the estimate of one coordinate in the sources, and it is trivial to get $y = \mathbf{w}^T\mathbf{x}$ where $\mathbf{w}$ is one of the rows in unmixing matrix $\mathbf{W}$. If $\mathbf{W}$ is the true inverse of mixing matrix $\mathbf{A}$, this linear combination $\mathbf{w}^T\mathbf{x}$ will present a real coordinate of sources. Since the absence of the prior information in mixing matrix, it will be more clear if we define a vector $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then a trivial conclusion is only one element in $\mathbf{z}$ is nonzero if $\mathbf{W}$ perfectly recover the mixing matrix. By simple linear algebra, we could rewrite $y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s}$,

which indicates the estimated source is a linear combination of independent non-Gaussian distributed sources (Hyvärinen *et al.* , 2004). According to Central Limit Theory, the sum of independent sources is more Gaussian compared with the single element in the original sources. However, based on the non-Gaussian assumption, we hope the estimated source component y is non-Gaussian distributed. Therefore, the source components have to be non-Gaussian with possible exception of at most one source component.

Without any prior information of the original sources and the mixing matrix, it is impossible to perfectly recover the original sources. Thus, ICA algorithms usually make a compromise by focusing on the independence quality of the original sources. As a consequence of lacking prior information, the scale of sources and mixing matrix is not identifiable. Moreover, the order of sources is also not identifiable. Thus, the estimated sources are ambiguous up to the magnitude and permutation.

3.2 New ICA method

As we mentioned before, most ICA algorithms are designed to estimate the unmixing matrix $\mathbf{W}$ based on certain objective functions that are usually derived from statistical independence or non-Gaussianity. In this section, we propose a new ICA algorithm based on a simple eigen-decomposition of newly proposed Fisher's Discriminant Information matrix.

3.2.1 Introduction of Fisher's Discriminant Information Matrix

Without loss of generality, we assume $\mathrm{E}(\mathbf{x}) = 0$ and $\mathrm{var}(\mathbf{x}) = I$. Suppose we want to compare two possible densities $f(\mathbf{x})$ and $g(\mathbf{x})$, respectively, based on a data set. In the

applications we will consider, the density f is defined as the true unknown density, to be estimated nonparametrically, and g will be parametric or semiparametric density modeling from the data. Alternatively, f and g could represent two distinct populations we wish to compare.

Define the *sample space score vector* $\mathbf{u}_f(\mathbf{x})$ for a density f to be $\mathbf{u}_f(\mathbf{x}) = \nabla_{\mathbf{x}} \log f(\mathbf{x})$. We define the basic *discrimination score* for comparing f and g to be $\mathbf{u}_f(\mathbf{x}) - \mathbf{u}_g(\mathbf{x})$. Note, if f and g are normal densities with means μ_f and μ_g, and with the same covariance Σ, then $\mathbf{u}_f - \mathbf{u}_g = \Sigma^{-1}(\mu_f - \mu_g)$. This is exactly the *Fisher's linear discriminant direction.*

We define *Fisher discriminant information matrix (FDIM)* to be the matrix quadratic form in the discrimination scores, given by

$$\mathbf{D}_w(f, g) = \int (\mathbf{u}_f(\mathbf{x}) - \mathbf{u}_g(\mathbf{x}))(\mathbf{u}_f(\mathbf{x}) - \mathbf{u}_g(\mathbf{x}))^T w(\mathbf{x}) d\mathbf{x}, \tag{3.3}$$

where $w(\mathbf{x})$ is a context-specific weighting density. For the normal example, this matrix is rank 1, with the nonnull eigenvector being the linear discriminant $\Sigma^{-1}(\mu_f - \mu_g)$. The defined matrix $\mathbf{D}_w(f, g)$ in (3.4) summarizes the local discrimination directions for separating f and g, and will be zero if and only if f and g are the same.

Let's use a simple example to see how $\mathbf{D}_w(f, g)$ works. Assuming that $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T$, the conventional definition of sufficiency indicates that $\mathbf{x}_1$ is sufficient for comparing f and g if the conditional densities for f and g are the same:

$$f(\mathbf{x}_2|\mathbf{x}_1) = g(\mathbf{x}_2|\mathbf{x}_1).$$

In this case, $\log f(\mathbf{x}) - \log g(\mathbf{x}) = \log f_1(\mathbf{x}_1) - \log g_1(\mathbf{x}_1)$, so the optimal discriminant function

only depends on $\mathbf{x}_1$. In this case it is obvious that

$$\mathbf{D}_w(f,g) = \begin{bmatrix} \mathbf{D}_w(f_1,g_1) & 0 \\ 0 & 0 \end{bmatrix}.$$

That is, the discrimination information matrix identifies, through their positive information, variables that are sufficient for discriminating between f and g as well as the set of variables, in the null space, that are ignorable. The eigenanalysis of $\mathbf{D}_w(f,g)$ will tell us which linear directions can best discriminate between f and g.

3.2.2 Application of Fisher discrimination information matrix to ICA

We use non-Gaussianity to perform ICA in (3.4) by letting $g = \phi(\mathbf{x})$ be the standard multivariate normal density and $f(\mathbf{x})$ be the true density of $\mathbf{x}$, to be estimated nonparametrically. If $w(\mathbf{x}) = f(\mathbf{x})$, using the fact that $\mathrm{E}_f\{\mathbf{u}_f\mathbf{u}_g^T\} = \mathrm{E}_f\{\mathbf{u}_g\mathbf{u}_f^T\} = \mathbf{0}$, we get

$$\mathbf{D}_f(f,g) = \int (\mathbf{u}_f - \mathbf{u}_g)(\mathbf{u}_f - \mathbf{u}_g)^T f(\mathbf{x}) d\mathbf{x} = \mathbf{J}_f - I_p, \tag{3.4}$$

where

$$\mathbf{J}_f = \mathrm{E}_f\{\mathbf{u}_f(\mathbf{x})\mathbf{u}_f(\mathbf{x})^T\} = \int \frac{\bigtriangledown_{\mathbf{x}} f(\mathbf{x}) \times \bigtriangledown_{\mathbf{x}} f^T(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x}, \tag{3.5}$$

where $\mathbf{J}_f$ is the so called *Density Information Matrix (DIM)* for f proposed by Hui & Lindsay (2010). In the DIM we use the derivative with respect to $\mathbf{x}$ instead of the parameters as done in traditional Fisher information matrix. So DIM can also be viewed as a measure of the information in the density and can characterize the multivariate properties of f. Statisticians are very familiar with the use of the covariance matrix $\mathbf{\Sigma}_f$ to help describe f. It turns out that $\mathbf{J}_f$ provides a complementary description of f. Principal component analysis uses the eigenanalysis of the covariance matrix to find directions that carry most of

the variability of the data. We will demonstrate that an eigenanalysis of $\mathbf{J}_f$ can be used to perform *independent components analysis* (ICA, Common 1994). Since $\mathbf{D}$ is non-negative definite, the result (3.4) implies $\mathbf{J}_f \geq I_p$, with the equality holding if and only if $f(\mathbf{x})$ is the multivariate normal density. Therefore, we used the newly defined discrimination matrix $\mathbf{D}_f(f, g)$ to provide an alternative proof of the *Fisher information inequality* (Kagan *et al.* , 1973).

Note that the eigen-space of $\mathbf{D}_f(f, g)$ associated with eigenvalue λ is the same as that of $\mathbf{J}_f$ associate with eigenvalue $\lambda + 1$. Specifically, the null space of $\mathbf{D}_f(f, g)$ is the same as the eigen-space of $\mathbf{J}_f$ associated with eigenvalue 1. Therefore, the eigen-analysis of $\mathbf{D}_f(f, g)$ can be also performed by the eigen-analsyis of $\mathbf{J}_f$ and it can find directions of the original multivariate variable $\mathbf{x}$ that have the largest departure from Gaussianity. Suppose $\Gamma^T \mathbf{J}_f \Gamma = \Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_p\}$. WLOG, we assume $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Then $\mathbf{z} = \Gamma^T \mathbf{x} = (z_1, \ldots, z_p)^T$ has density information matrix Λ, and the diagonal entries of Λ measure the information in each $\mathbf{z}$ coordinate in terms of extent of discrimination between $\mathbf{z}$ and the normal density.

We will use the notation $\mathbf{J}_{\mathbf{x}}$, expressed with a random variable, instead of $\mathbf{J}_f$, using $\mathbf{x}'s$ density f, when it is useful for the clarity regarding different variables involved. The eigenanalysis of DIM is usually performed in two stages.

Stage 1 Standardize the original variable $\mathbf{x}$ to the vector $\mathbf{y} = \mathbf{\Sigma}_{\mathbf{x}}^{-1/2}\mathbf{x}$. Based on the variable $\mathbf{y}$ we then create the density information matrix for $\mathbf{y}$, denoted $\mathbf{J}_{\mathbf{y}} = \mathbf{\Sigma}_{\mathbf{x}}^{1/2}\mathbf{J}_{\mathbf{x}}\mathbf{\Sigma}_{\mathbf{x}}^{1/2} \triangleq \mathbf{J}_{\mathbf{x}}^*$. We call $\mathbf{J}_{\mathbf{x}}^*$ *standardized density information matrix* since it is the DIM for the standardized variable $\mathbf{y}$.

Note that $\text{var}(\mathbf{y}) = I$, so the variables are uncorrelated and the $\mathbf{y}$ variables have no principal components information. Therefore, after standardization, $\mathbf{J_y}$ will provide the complimentary information about $\mathbf{y}$ that the principal component analysis can not.

Stage 2 We create an orthogonal matrix $\mathbf{\Gamma}$ using the eigenanalysis of $\mathbf{J_y}$ such that $\mathbf{J_y} = \mathbf{\Gamma\Lambda\Gamma}^T$, where $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_p\}$ is diagonal matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Then our interested projections are the new vector $\mathbf{z} = \mathbf{\Gamma}^T\mathbf{y} = \mathbf{\Gamma}^T\mathbf{\Sigma}_\mathbf{x}^{-1/2}\mathbf{x}$. Note that $\text{var}(\mathbf{z}) = \mathbf{I}$ and $\mathbf{J}_z = \mathbf{\Gamma}^T\mathbf{J_y}\mathbf{\Gamma} = \text{diag}\{\lambda_1, \ldots, \lambda_p\}$. Therefore, $\mathbf{z}$ has a diagonal covariance matrix and diagonal density information matrix.

Note that when the covariance $\mathbf{\Sigma_x}$ are diagonal, the density information matrix $\mathbf{J_x}$ is not necessary diagonal. However, if the elements of $\mathbf{x}$ are independent, then both the covariance $\mathbf{\Sigma_x}$ and information matrix $\mathbf{J_x}$ are diagonal. The new variables $\mathbf{z}$ thereby mimic this dual property of independent variables. Lindsay & Yao (2012) proved such found $\mathbf{z}$ after two stages of DIM analysis are the independent component variables.

Proposition 1 *If the data* $\mathbf{x}$ *is generated by an independent components analysis model with covariance matrix* $\mathbf{\Sigma}$*, and the eigenvalues of the standardized DIM* $\mathbf{J}_\mathbf{x}^*$ *are distinct, then the transformed variables* $\mathbf{z} = \mathbf{\Gamma}^T\mathbf{\Sigma}^{-1/2}\mathbf{x}$ *are the independent components variables, up to the permutation, where* $\mathbf{\Gamma}$ *is the matrix of eigenvectors of* $\mathbf{J}_\mathbf{x}^*$.

Therefore, stage 1 will first transform $\mathbf{x}$ to the uncorrelated variable $\mathbf{y} = \mathbf{\Sigma}_\mathbf{x}^{-\frac{1}{2}}\mathbf{x}$ with diagonal covariance matrix, which is only independent if $\mathbf{x}$ is multivariate Gaussian. The stage 2 makes the transformed variable $\mathbf{z}$ even closer to independent variables by forcing its DIM $\mathbf{J_z}$ to be diagonal. The above results tell us if $\mathbf{x}$ is generated by ICA model, such

created $\mathbf{z}$ can recover the independent component variables. Based on the above arguments, we can also see why the DIM can provide complement information of the covariance matrix.

Compared to existing ICA methods, the new method can further detect whether there is any Gaussian component by checking whether any diagonal value of $\mathbf{J_z}$ is equal to 1. In addition, we can also rank the recovered independent components of $\mathbf{z}$ in term of the defined density information using the corresponding diagonal values of $\mathbf{J_z}$ with Gaussian component has the least information. The next proposition validates the above claim.

Proposition 2 *Under the assumption of Proposition 1, the jth diagonal value λ_j of $\mathbf{J_z}$ is exactly the density information of the jth marginal distribution of $\mathbf{z}$, i.e, the distribution of z_j.*

Proof. Since, $z_1, \ldots, z_p$ are independent, $f(\mathbf{z}) = f_1(z_1) \cdots f_p(z_p)$ and $\log f(\mathbf{z}) = \sum_{j=1}^{p} \log f(z_j)$. Then

$$\mathbf{U}(\mathbf{z}) = \nabla_{\mathbf{z}} \log f(\mathbf{z}) = (\nabla_{z_1} \log f(z_1), \ldots, \nabla_{z_p} \log f(z_p))^T.$$

Therefore, the jth diagonal term of $\mathbf{J_z}$ can be expressed as $\mathrm{E}[\mathbf{U}_j(\mathbf{z})^2] = \mathrm{E}[\nabla_{z_j} \log f(z_j)^2] = J_{z_j}$. ■

3.2.3 Density square transformation

So far our discussion has been at the population level. In practice, in order to recover $\mathbf{z} = \mathbf{\Gamma}^T \mathbf{\Sigma}^{-1/2} \mathbf{x}$, we need to replace $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ by some estimate. $\mathbf{\Sigma}$ can be usually estimated by sample covariance. The main difficult lies on the estimation of $\mathbf{\Gamma}$ and thus $\mathbf{J_x}$. Since the variable $\mathbf{x}$'s density $f(\mathbf{x})$ is unknown, an appropriate non-parametric or semi-parametric estimation method is needed. However, based on (3.5), there is a second

computational problem that the integration will not have an explicit form due to the density function in denominator and thus the numerical integration would generally be required. Although one could proceed with a simulation based method, we will instead use the idea of the *density square transformation* proposed by Hui & Lindsay (2010) to make the calculation fast and explicit. They did so by slightly altering the information problem as follows. Let the variable $\mathbf{s}$ have the density

$$f_2(\mathbf{s}) \equiv \frac{f^2(\mathbf{s})}{\int f^2(\mathbf{x})d\mathbf{x}},$$

where $f(\mathbf{x})$ is the density of $\mathbf{x}$. They proposed to estimate the information in the density $f_2(\mathbf{s})$, which we can denote as $\mathbf{J}_\mathbf{s}$ or $\mathbf{J}_{f_2}$

$$\mathbf{J}_{f_2} = \frac{\int \bigtriangledown_\mathbf{x} f \times \bigtriangledown_\mathbf{x} f^T d\mathbf{x}}{\int f^2(\mathbf{x})d\mathbf{x}}. \tag{3.6}$$

Then $\mathbf{\Gamma}$ can be estimated by the eigenanalysis of $\mathbf{\Sigma}_{f_2}^{1/2}\mathbf{J}_{f_2}\mathbf{\Sigma}_{f_2}^{1/2}$. If $f(\mathbf{x})$ is estimated by a kernel density estimate with normal kernel or Gaussian mixture models, then $\mathbf{J}_{f_2}$ and $\mathbf{\Sigma}_{f_2}$ have an explicit formula. Estimating the most informative directions for $\mathbf{s}$ turned out to be a very useful surrogate for estimating the most informative directions for $\mathbf{x}$. There is good intuition for this: as argued by Hui & Lindsay (2010), the square density $f_2(\mathbf{s})$ has the same contour lines as the original $f(\mathbf{x})$ and in particular the same peaks and valleys. In addition, $\mathbf{x}$ is normal if and only if $\mathbf{s}$ is normal, so the white noise subspaces are preserved again by the density square transformation, *regardless* of bandwidth. This property makes the method work well even when the dimension of $\mathbf{x}$ is moderately large. Finally, as a weighting factor, $f_2(\mathbf{s})$ puts more weight on the peaks and less weight in the tails than $f(\mathbf{x})$; this seems to improve the robustness of the method based on the empirical studies. We will call this the *f2 method of computation*. Please see Hui & Lindsay (2010) for examples of the success of

this methodology in higher dimensions.

With Kernel density estimation, suppose a multivariate sample $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ are drawn from a density f. We propose to estimate f by the multivariate kernel estimate

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{n|\mathbf{H}|} \phi_p(\mathbf{x} - \mathbf{x}_i \,;\, \mathbf{0}, \mathbf{H}^2),$$

where $\phi_p(\cdot \,;\, \mathbf{0}, \mathbf{H}^2)$ is the p-variate Gaussian density with mean $\mathbf{0}$ and covariance $\mathbf{H}^2$.

Here, we choose normal density as Kernel function $\underline{\mathrm{K}}$ and use the bandwidth recommended by Bowman & Foster (1993),

$$\mathbf{H}_{opt} = (\frac{4}{p+2})^{\frac{1}{p+4}} \mathbf{\Sigma}^{\frac{1}{2}} n^{-\frac{1}{p+4}}. \tag{3.7}$$

where the unknown $\mathbf{\Sigma}$ is usually replaced by its sample estimate. Then, the estimated density information matrix has the following form

$$\hat{\mathbf{J}}_{f_2} = \frac{\int \bigtriangledown_{\mathbf{x}} \hat{f}_{\mathbf{H}} \times \bigtriangledown_{\mathbf{x}} \hat{f}_{\mathbf{H}}^T d\mathbf{x}}{\int \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}}. \tag{3.8}$$

where

$$\begin{aligned}
\int \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x} &= \frac{1}{n^2} \sum_{i=i}^{n} \sum_{j=1}^{n} \phi_p(\mathbf{x}_i - \mathbf{x}_j; 0, 2\mathbf{H}^2), \\
\int \bigtriangledown_{\mathbf{x}} \hat{f}_{\mathbf{H}} \times \bigtriangledown_{\mathbf{x}} \hat{f}_{\mathbf{H}}^T d\mathbf{x} &= \frac{1}{n^2} \sum_{i=i}^{n} \sum_{j=1}^{n} \phi_p(\mathbf{x}_i - \mathbf{x}_j; 0, 2\mathbf{H}^2) \\
&\quad \times \left[\frac{\mathbf{H}^{-2}}{2} + \frac{\mathbf{H}^{-2}}{2} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \frac{\mathbf{H}^{-2}}{2} \right].
\end{aligned}$$

In addition, the variance of $\mathbf{s}$ can be estimated by

$$\hat{\mathbf{\Sigma}}_{\mathbf{s}} = \frac{\int \mathbf{x}\mathbf{x}^T \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}}{\int \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}} - (\frac{\int \mathbf{x} \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}}{\int \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}})(\frac{\int \mathbf{x} \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}}{\int \hat{f}_{\mathbf{H}}^2(\mathbf{x}) d\mathbf{x}})^T,$$

where

$$\int \mathbf{x}\mathbf{x}^T \hat{f}^2_{\mathbf{H}}(\mathbf{x})d\mathbf{x} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi_p(\mathbf{x}_i - \mathbf{x}_j; 0, 2\mathbf{H}^2)\left[\frac{\mathbf{H}^2}{2} + \frac{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{4}\right],$$

$$\int \mathbf{x}\hat{f}^2_{\mathbf{x}}(\mathbf{x})d\mathbf{x} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi_p(\mathbf{x}_i - \mathbf{x}_j; 0, 2\mathbf{H}^2)\frac{(\mathbf{x}_i + \mathbf{x}_j)}{2}.$$

We will call the above ICA method *DIM-KDE* using the kernel density estimation and *f2* transformation to estimate DIM. Based on the above formula, we can see that one of the major advantages of *f2* computation method is that it provides explicit formula for all integrations when normal kernel is used.

Alternatively, Gaussian mixture model (GMM) can be also applied to estimate unknown density of $f(\mathbf{x})$. It is well known that the mixture models can be used as a nonparametric density estimate if the number of components is large enough. In fact, the kernel density estimate is a special case of mixture models with n components. More specifically, we estimate $f(\mathbf{x})$ by

$$f_{\mathbf{GMM}}(\mathbf{x}) = \sum_{i=1}^{k}\pi_i\phi_p(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{3.9}$$

where $\sum_{i=1}^{k}\pi_i = 1$ and k is the number of mixture components and assumed to be unknown. By data adaptively estimate $\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i$, we can use much smaller k, compared to the n used by kernel density estimate, to approximate the density of $\mathbf{x}$.

The model (3.9) can be easily estimated by the EM algorithm (Dempster *et al.* , 1977) and k can be selected by Bayesian information criteria (BIC). Using the estimate

(3.9), we have

$$\int \hat{f}^2_{\mathbf{GMM}}(\mathbf{x})d\mathbf{x} = \sum_{l=1}^{k}\sum_{m=1}^{k} \pi_l \pi_m \phi_p(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m; 0, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m),$$

$$\int \bigtriangledown_{\mathbf{x}} \hat{f}_{\mathbf{GMM}} \times \bigtriangledown_{\mathbf{x}} \hat{f}^T_{\mathbf{GMM}} d\mathbf{x} = \sum_{l=1}^{k}\sum_{m=1}^{k} \pi_l \pi_m \phi_p(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m; 0, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)$$

$$\times \left[(\boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)^{-1} + (\boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)^{-1}(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m)(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m)^T(\boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)^{-1}\right],$$

$$\int \mathbf{x}\mathbf{x}^T \hat{f}^2_{\mathbf{GMM}}(\mathbf{x})d\mathbf{x} = \sum_{l=1}^{k}\sum_{m=1}^{k} \pi_l \pi_m \phi_p(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m; 0, (\boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)\left[\boldsymbol{\Sigma}_c + \boldsymbol{\mu}_c\boldsymbol{\mu}_c^T\right],$$

$$\int \mathbf{x} \hat{f}^2_{\mathbf{GMM}}(\mathbf{x})d\mathbf{x} = \sum_{l=1}^{k}\sum_{m=1}^{k} \pi_l \pi_m \phi_p(\boldsymbol{\mu}_l - \boldsymbol{\mu}_m; 0, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_m)\boldsymbol{\mu}_c,$$

where $\boldsymbol{\Sigma}_c = (\boldsymbol{\Sigma}_l^{-1} + \boldsymbol{\Sigma}_m^{-1})^{-1}$ and

$$\boldsymbol{\mu}_c = (\boldsymbol{\Sigma}_l^{-1} + \boldsymbol{\Sigma}_m^{-1})^{-1}(\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l + \boldsymbol{\Sigma}_m^{-1}\boldsymbol{\mu}_m).$$

Plugging in the above equations to (3.6), we can get the estimate of $\mathbf{J}_{f2}$. We will call the above ICA method *DIM-GMM* that uses Gaussian mixture model and *f2* transformation to estimate DIM.

3.3 Simulation study

In this section, a simulation study is conducted to investigate the finite sample performance of *DIM-KDE* and *DIM-GMM*. The performance is evaluated via Amari Distance

$$E = \sum_{i=1}^{p}(\sum_{j=1}^{p}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{i=j}^{p}(\sum_{i=1}^{p}\frac{|p_{ij}|}{\max_k |p_{kj}|} - 1), \tag{3.10}$$

where $\mathbf{P} = (p_{ij}) = \mathbf{WA}$ with the range $[0, p-1]$ (Amari *et al.* , 1996). If $\mathbf{W} = \mathbf{A}^{-1}$, then $E = 0$. Thus, we want E as small as possible.

Figure 3.1 displays 9 distributions, introduced by Bach & Jordan (2002), from which the original source signals are generated. In this simulation study, all pairs of independent components are drawn from different distributions. We simulate two dimensional

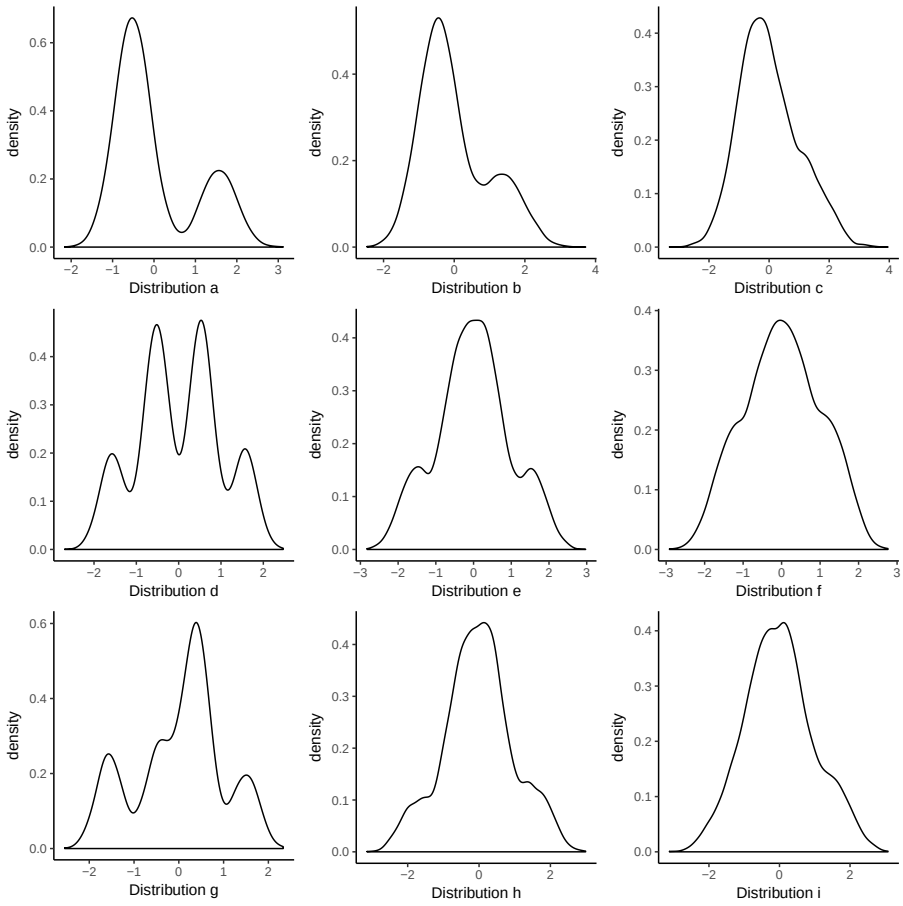

Figure 3.1: The 9 distributions proposed by Bach & Jordan (2002) are used to generate the original source signal $\mathbf{s}(t)$.

($p = 2$) independent components $\mathbf{s}(t)$ with sample size $n = 200$ and 1000 and a random mixing matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$. The observations are generated by $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$. We compared *DIM-KDE* and *DIM-GMM* with *FastICA*, *Infomax*, and *SteadyICA* (Bell & Sejnowski,

1995; Hyvärinen *et al.* , 2004; Matteson & Tsay, 2017). The following table lists the ICA algorithms and the corresponding R Packages used.

ICA Algorithm	R package
FastICA	a fast fixed-point ICA algorithm (R Package 'fastICA')
Infomax	Information maximization ICA algorithm (R Package 'ica')
SteadyICA	ICA via distance covariance (R Package 'steadyICA')

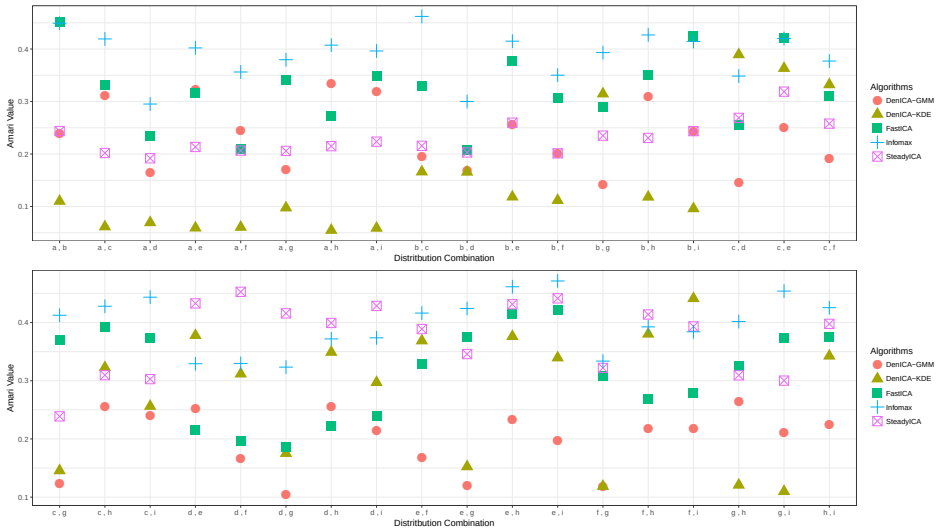

Figure 3.2: The comparison results for two dimensional source signals with sample size 200 over 100 replications and *DIM-KDE*, *DIM-GMM* are compared with three existing ICA algorithms.

In Figure 3.2 and 3.3, we compare our DIM ICA algorithms with the above three ICA algorithm under each sample size. The horizontal axis indicates the pairs of distribution, while the vertical axis presents the average Amari Distance over 100 replications. In Figure 3.2, *DIM-KDE* shows a good performance in the upper plot, while in lower plot the *DIM-GMM* works better. Overall, the DIM ICA algorithms have superior performance compared with other algorithms. According to Figure 3.3, when increasing sample size, all

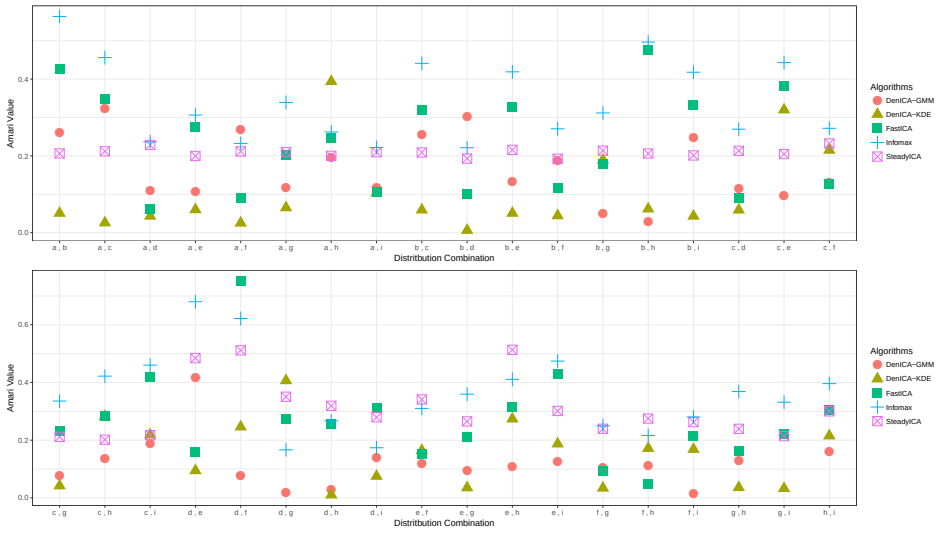

Figure 3.3: The comparison results for two dimensional source signals with sample size 1000 over 100 replications and *DIM-KDE*, *DIM-GMM* are compared with three existing ICA algorithms.

ICA algorithms achieve smaller average Amari Distance. In addition, *DIM-KDE* becomes stable and has the overall best performance.

3.4 Application

In independent component analysis problem, we are interested in recovering the original sources or finding "interesting" coordinates. ICA algorithm applies transformation to convert the raw dataset into sets of independent variables. In this section, we present four real data applications.

3.4.1 Cocktail party problem

In this application, the famous Blind Source Separation (BSS) problem "Cocktail Party Problem" is illustrated and resolved with *DIM-KDE* algorithm due to the large sample size. Suppose there are four speakers placed in one room; three speakers play different music and one speaker plays white noise sound. Figure 3.4 shows the shape of original sound sources. As the music begins to play, all four sound sources are mixed together. At the same time, four microphones which located in the same room record the mixed sounds. Figure 5 displays the shape of mixed sounds. In order to illustrate the BSS

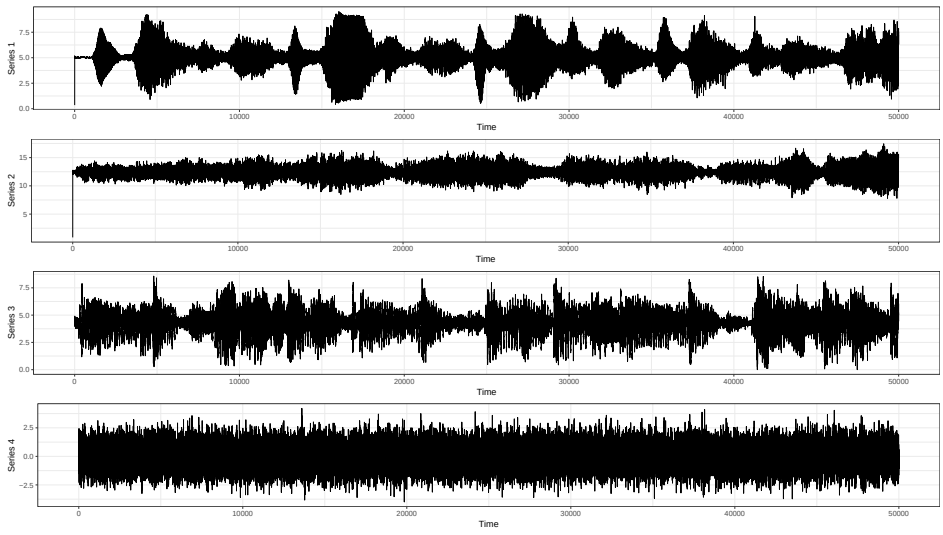

Figure 3.4: "Cocktail Party Problem" consists of four original sound sources with one white noise in the last position.

mechanism, we transform the music sound signals into a data matrix $\mathbf{S}$. In BSS problem, only the mixed sound recordings $\mathbf{x}(t)$ are observable from microphones. The target of ICA algorithm is to recover the original music only with data matrix $\mathbf{x}(t)$.

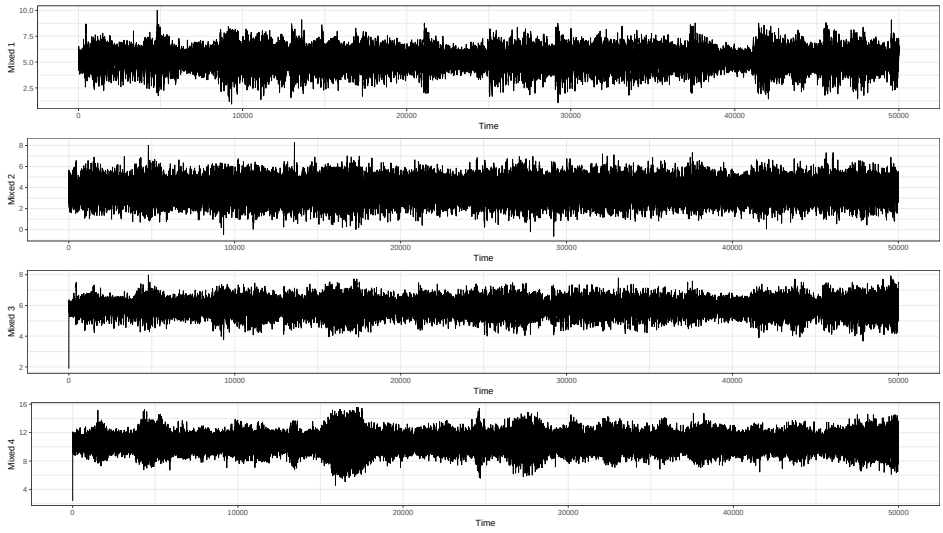

Figure 3.5: The plot shows the mixture of original sound sources with unknown mixing procedure matrix **A**.

Figure 3.6 presents the result from *DIM-KDE* algorithm. Clearly, the *DIM-KDE* algorithm recovers in great extend the shape of each the origin music sound. Due to the permutation ambiguity, the recovered sound sources have different ordering from the original one. Note, however, the proposed *DIM-KDE* algorithm ranks the four sound sources from top to bottom in terms of their density information. In addition, *DIM-KDE* algorithm successfully puts the white noise coordinate, that has the least information ($\lambda = 0.26$), in the last coordinate. From Figure 3.6, it seems that the top three estimated sound sources become closer to the noise sound when moving from top to bottom.

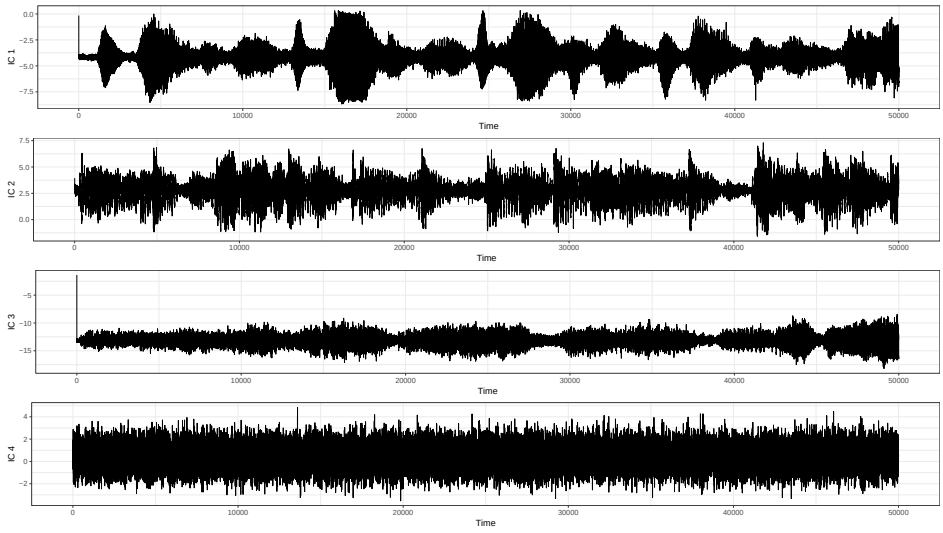

Figure 3.6: The plot displays the recovered sound sources via *DIM-KDE* algorithm and *DIM-KDE* algorithm automatically orders the recovered sound sources, also put the white noise in the last position.

3.4.2 Imaging Processing with ICA

In practice, ICA is also a popular technique of image processing. Consider a grey-scale image "*boat*" shown in Figure 3.7 a). We transform the image into a data matrix $\mathbf{S}_1$ (256×384). First, a white noise matrix $\mathbf{S}_2$ with the same dimension as $\mathbf{S}_1$ is generated. Second, these two data matrices, $\mathbf{S}_1$ and $\mathbf{S}_2$, are converted into one data matrix $\mathbf{S}$ (98304×2). With some unknown procedure $\mathbf{X} = \mathbf{AS}$, the observed image is shown in Figure 3.7 b) which combines both original image and the white noise.

Due to the large sample size, we apply *DIM-KDE* algorithm as image filter and the recovered image is presented in Figure 3.8. Obviously, *DIM-KDE* algorithm filters the observed image and move out the white noise.

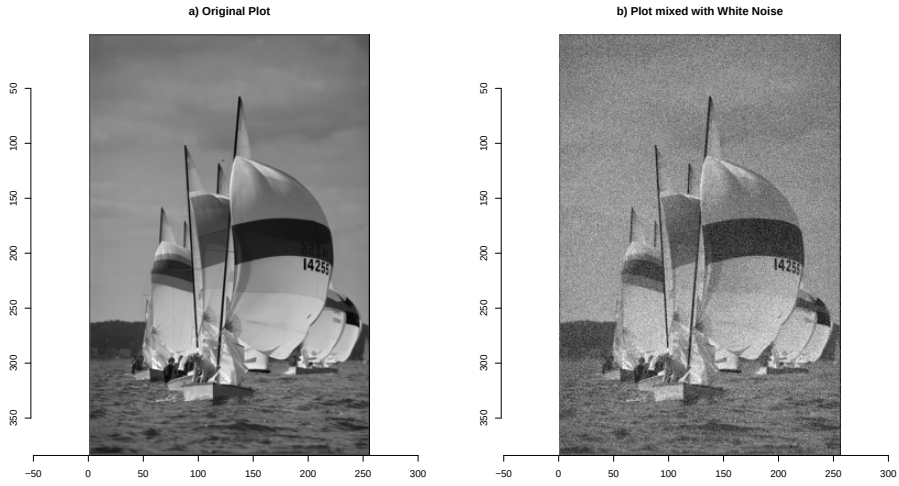

Figure 3.7: The plot shows an ICA application with image *boat* with the left plot a) showing the original plot and the right plot showing a combining plot via a white noise plot through unknown procedure **A**.

3.4.3 Fisher's Iris Flower

Also, as an unsupervised machine learning method, ICA algorithm is a widely used tool for clustering. The following two examples illustrate the clustering application of ICA algorithm. The Fisher's Iris flower data set contains 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor) with four variables, the length and the width of the sepals and petals. This dataset is popular as a benchmark in clustering algorithm and the data are available in the R *dataset* library. The data is treated as unlabeled with species with dimension 150×4. If we consider the observable four variables as mixed signals, the ICA algorithm recovers variables into distinct intrinsic characters (original signals), and then clusters flower samples with the first two intrinsic variables.

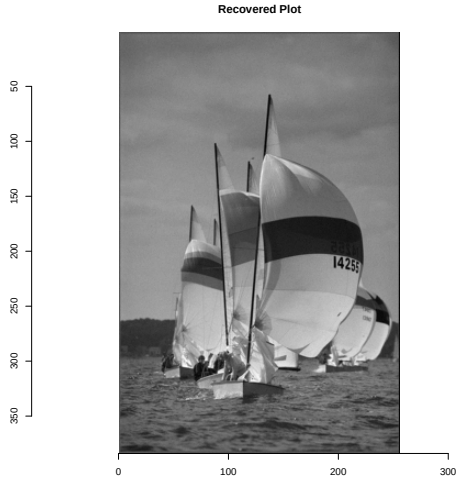

Figure 3.8: The plot shows the recovered image via *DIM-KDE* algorithm and the plot is generated by the first component in the estimated source matrix **S**. The second component in **S** contains the white noise estimate.

Figure 3.9 a) shows the clustering results based on the first two independent components. In comparison, we also present a similar plot using first two projections from principal component analysis (PCA) in Figure 3.9 b). From the Figure 3.9, we can see that compared with PCA method, the *DIM-KDE* algorithm has better performance as a clustering method. Note that the new method can also rank the transformed directions in terms of their density information. In Figure 3.9 a), it can be seen that the first direction is the most informative direction that contains the most clustering information and the second direction shows the heterogeneous variability among three clusters.

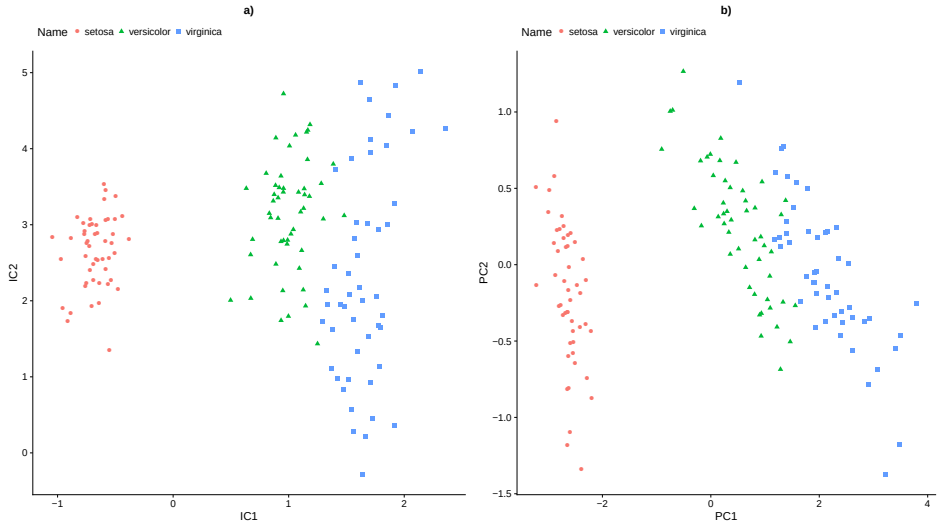

Figure 3.9: The plot shows ICA application of clustering with Fisher's Iris Flower data with the left plot a) presenting the clustering result based on the first two components of estimated sources $\mathbf{S}$ via *DIM-KDE* algorithm and the right plot b) presenting the clustering result based on the first two components with PCA method.

3.4.4 Leptograpsus Crabs

The Leptograpsus Crabs is another popular data set used for comparing various classifiers. The experimenter recorded five morphometric measurements on two crabs (blue and orange) with two genders (Female and Male) and the data are available in the R *MASS* library. Note that in this data there are four clusters. These five variables are highly correlated. Thus, it is difficult to cluster the crabs only relying on these variables. In Figure 3.10 a) and b), we plot the first two projections from *DIM-KDE* and PCA, respectively. Based on Figure 3.10, *DIM-KDE* has better clustering performance than PCA.

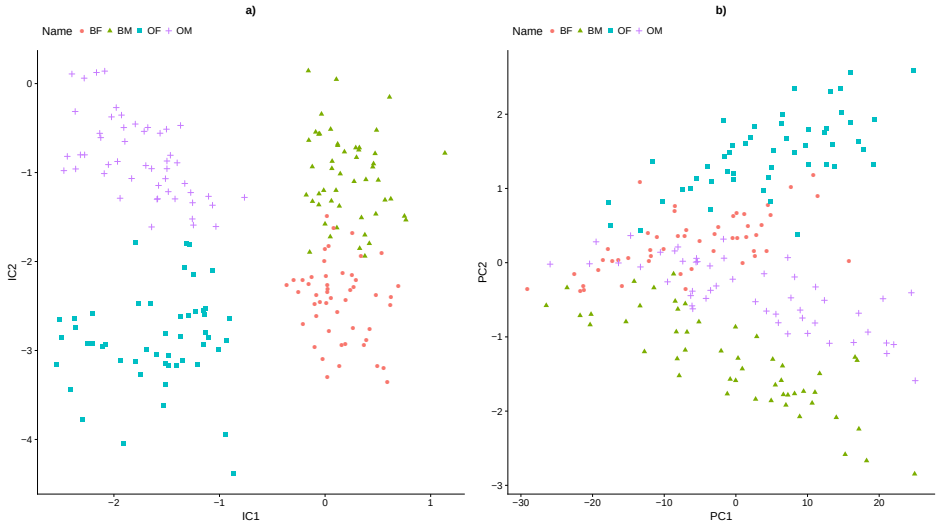

Figure 3.10: The plot shows ICA application of clustering with Leptograpsus Crabs data with The left plot a) presenting the result based on the first two components of estimated sources **S** via *DIM-KDE* algorithm and the right plot b) presenting the clustering result based on the first two components with PCA method.

Chapter 4

Discussion

In Chapter 2, we propose a mixed-effects segmented regression model motivated by forecasting the electric load in Southern California. When estimating unknown parameters, we propose a backfitting algorithm by combining the ideas of the penalized least square method for random-effects regression model and the linearization technique (Muggeo, 2003) for segmented regression. In addition, we extend the idea of LTS to the new mixed-effects segmented regression model to provide a robust model estimate. Both Simulation study and real data application demonstrate the effectiveness of the proposed new model and its estimation procedures.

Since the model was built up with hourly data, we could also aggregate the data and construct a daily electric load model. In this paper, we assume that the number of breakpoints is known. If the number of breakpoints is unknown, one could apply the selection techniques proposed by Ben Aïssa *et al.* (2004); Liu *et al.* (1997); Prodan (2008); Strikholm & Teräsvirta (2005) to our model. In addition, for LTS, although an conservation

α or serval α values can be used in practice, it requires more research to data adaptively choose the optimal α so that LTS can have both the robustness property and the high efficiency.

In Chapter 3, we introduce a new ICA method *DIM* based on a simple eigenanalysis of density information matrix. To estimate the density information matrix, we proposed two estimation methods: kernel density estimation and Gaussian mixture model.

The important advantages of *DIM* ICA method are that it has the ability to order recovered coordinates in terms of the density information and also identify the white noise coordinate which has the least density information. Simulation results demonstrate the effectiveness of the *DIM* method in recovering sources drawn from different distributions. As demonstrated by the simulations and real applications, *DIM* has overall superior performance across sample size. Moreover, the performance of *DIM-KDE* is more stable with large sample size.

There are several ways to extend the new ICA method. Note that one restriction of the *DIM* method is the assumption that the source signals have different distributions. It requires more research to relax this assumption for our new method. In addition, our method requires the source signal to have continuous distributions with continuous first order derivative due to the definition of DIM. Therefore, it is also interesting to extend our method to discrete distributions or continuous distribution that may not have continuous first order derivative.

The high dimensional dataset becomes more common nowadays. Extending the ICA method into the high dimensional setting is also very challenging due to the estimation

of DIM and covariance. In this situation, some sparsity assumption about DIM and the covariance matrix might be imposed to facilitate the estimation.

Bibliography

Adib, Rana, Murdock, HE, Appavou, F, Brown, A, Epp, B, Leidreiter, A, Lins, C, Murdock, HE, Musolino, E, Petrichenko, K, *et al.* . 2016. Renewables 2016 Global Status Report. *Global Status Report RENEWABLE ENERGY POLICY NETWORK FOR THE 21st CENTURY (REN21)*.

Amari, Shun-ichi, Cichocki, Andrzej, & Yang, Howard Hua. 1996. A new learning algorithm for blind signal separation. *Pages 757–763 of: Advances in neural information processing systems*.

Bach, Francis R, & Jordan, Michael I. 2002. Kernel independent component analysis. *Journal of machine learning research*, **3**(Jul), 1–48.

Bates, Douglas. 2011. Computational methods for mixed models. *Vignette for lme4*.

Bell, Anthony J, & Sejnowski, Terrence J. 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, **7**(6), 1129–1159.

Ben Aïssa, Mohamed Safouane, Boutahar, Mohamed, & Jouini, Jamel. 2004. Bai and Perron's and spectral density methods for structural change detection in the US inflation process. *Applied Economics Letters*, **11**(2), 109–115.

Bezdek, J. C., Hathaway, R. M., & Huggins, V. J. 1985. Parametric estimation for normal mixtures. *Pattern Recognition*, **3**, 79–84.

Böhning, D. 1999. *Computer-Assisted Analysis of Mixtures and Applications*. Boca Raton, FL: Chapman and Hall/CRC.

Bordes, Laurent, Mottelet, Stéphane, Vandekerkhove, Pierre, *et al.* . 2006. Semiparametric estimation of a two-component mixture model. *The Annals of Statistics*, **34**(3), 1204–1232.

Bowman, AW, & Foster, PJ. 1993. Adaptive smoothing and density-based tests of multivariate normality. *Journal of the American Statistical Association*, **88**(422), 529–537.

Brown, L. 1964. Sufficient statistics in the case of independent random variables. *Ann. Math. Statist.*, **35**, 1456.

Cardoso, Jean-François. 1999. High-order contrasts for independent component analysis. *Neural computation*, **11**(1), 157–192.

Cardoso, Jean-François, & Souloumiac, Antoine. 1993. Blind beamforming for non-Gaussian signals. *Pages 362–370 of: IEE proceedings F (radar and signal processing)*, vol. 140. IET.

Celeux, G. 1998. Bayesian inference for mixtures: The label switching problem. *Pages 227–232 of:* Payne, R., & Green, P.J. (eds), *In Compstat 98-Proc. in Computational Statistics*.

Celeux, G., Hurn, M., & Robert, C. P. 2000. Computational and inferential difficulties with mixture posterior distributions. *Journal of American Statistical Association*, **95**, 957–970.

Chen, Jiahua, & Kalbfleisch, J. D. 1996. Penalized minimum-distance estimates in finite mixture models. *Canadian Journal of Statistics*, **24**(2), 167–175.

Chen, Jiahua, & Li, Pengfei. 2009. Hypothesis test for normal mixture models: The EM approach. *The Annals of Statistics*, **37**, 2523–2542.

Chen, Jiahua, Li, Pengfei, & Fu, Yuejiao. 2012. Inference on the order of a normal mixture. *Journal of the American Statistical Association*, **107**(499), 1096–1105.

Choi, Seungjin, Cichocki, Andrzej, & Amari, Shun-Ichi. 2000. Flexible independent component analysis. *Journal of VLSI signal processing systems for signal, image and video technology*, **26**(1-2), 25–38.

Chung, H., Loken, E., & Schafer, J. L. 2004. Difficulties in drawing inferences with finite-mixture models: a simple example with a simple solution. *The American Statistician*, **58**, 152–158.

Comon, Pierre. 1994. Independent component analysis, a new concept? *Signal processing*, **36**(3), 287–314.

Cox, Christopher. 1987a. Threshold dose-response models in toxicology. *Biometrics*, 511–523.

Cox, Christopher. 1987b. Threshold dose-response models in toxicology. *Biometrics*, 511–523.

Crawford, S. L. 1994. An application of the Laplace method to finite mixture distributions. *Journal of American Statistical Association*, **89**, 259–267.

Crawford, S. L., Degroot, M. H., Kadane, J. B., & Small, M. J. 1992. Modeling lake-chemistry distributions-approximate Bayesian methods for estimating a finite-mixture model. *Techometrics*, **34**, 441–453.

Cron, A., & West, M. 2011. Efficient Classification-Based Relabeling in Mixture Models. *The American Statistician*, **65**, 16–20.

de Jong, P. 1987. A central limit theorem for generalized quadratic forms. *Probab. Theory Related Fields*, **75**, 261–277.

Dellaportas, P., & Papageorgious, I. 2006. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, **16**(1), 57–68.

Dellaportas, P., Stephens, D. A., Smith, A. F. M., & Guttman, I. 1996. A comparative study of perinatal mortality using a two-component mixture model. *Pages 601–616 of:* Berry, D.A., & Stangl, D.K. (eds), *Bayesian Biostatistics*. CRC Press.

Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of Royal Statistical Association: Series B*, **39**, 1–38.

Diebolt, J., & Robert, C. P. 1994. Estimation of finite mixture distributions through Bayesian sampling. *Journal of Royal Statistical Association: Series B*, **56**, 363–375.

Feder, Paul I. 1975a. The log likelihood ratio in segmented regression. *The Annals of Statistics*, 84–97.

Feder, Paul I. 1975b. The log likelihood ratio in segmented regression. *The Annals of Statistics*, 84–97.

Friedman, Jerome H, & Tukey, John W. 1974. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on computers*, **100**(9), 881–890.

Frühwirth-Schnatter, S. 2001. Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. *Journal of American Statistical Association*, **96**, 194–209.

Frühwirth-Schnatter, S. 2006. *Finite Mixture and Markov Switching Models*. Springer.

Garel, Bernard. 2005. Asymptotic theory of the likelihood ratio test for the identification of a mixture. *Journal of statistical planning and inference*, **131**(2), 271–296.

Geweke, J. 2007. Interpretation and inference in mixture models: Simple MCMC works. *Computational Statistics and Data Analysis*, **51**, 3529–3550.

Ghosh, Jayanta K, & Sen, Pranab Kumar. 1984. On the asymptotic performance of the log likelihood ratio statistic for the mixture model and related results.

Gössl, Christoff, & Küchenhoff, Helmut. 2001a. Bayesian analysis of logistic regression with an unknown change point and covariate measurement error. *Statistics in medicine*, **20**(20), 3109–3121.

Gössl, Christoff, & Küchenhoff, Helmut. 2001b. Bayesian analysis of logistic regression with an unknown change point and covariate measurement error. *Statistics in medicine*, **20**(20), 3109–3121.

Grün, B., & Leisch, F. 2009. Dealing with label switching in mixture models under genuine multimodality. *Journal of Multivariate Analysis*, **100**, 851–861.

Hastie, Trevor, & Tibshirani, Rob. 2003. Independent components analysis through product density estimation. *Pages 665–672 of: Advances in neural information processing systems*.

Hastie, Trevor, & Tibshirani, Robert. 1990a. *Generalized additive models*. Wiley Online Library.

Hastie, Trevor, & Tibshirani, Robert. 1990b. *Generalized additive models*. Wiley Online Library.

Hathaway, R. J. 1983. *Constrained maximum likelihood estimation for a mixture of m univariate normal distribbutions*. Tech. rept. University of South Carolina, Columbia, South Carolina.

Hathaway, R. J. 1985. A constrained formulation of maximum-likelihood estimation for nomral mixture distributions. *Annals of Statistics*, **13**, 795–800.

Hathaway, R. J. 1986. A constrained EM algorithm for univariate mixutres. *Journal of Statistical Computation and Simulation*, **23**, 211–230.

Hendricks, Wallace, & Koenker, Roger. 1992a. Hierarchical spline models for conditional quantiles and the demand for electricity. *Journal of the American statistical Association*, **87**(417), 58–68.

Hendricks, Wallace, & Koenker, Roger. 1992b. Hierarchical spline models for conditional quantiles and the demand for electricity. *Journal of the American statistical Association*, **87**(417), 58–68.

Huang, M., Li, R., & Wang, S. 2013. Nonparametric mixture of regression models. *Journal of the American Statistical Association*, **108**, 929–941.

Huang, Mian, & Yao, Weixin. 2012. Mixture of Regression Models With Varying Mixing Proportions: A Semiparametric Approach. *Journal of the American Statistical Association*, **107**(498), 711–724.

Huber, Peter J. 1985. Projection pursuit. *The annals of Statistics*, 435–475.

Hui, Guodong, & Lindsay, Bruce G. 2010. Projection pursuit via white noise matrices. *Sankhya B*, **72**(2), 123–153.

Hunter, David R, Wang, Shaoli, & Hettmansperger, Thomas P. 2007. Inference for mixtures of symmetric distributions. *The Annals of Statistics*, 224–251.

Hurn, M., Justel, A., & Robert, C. P. 2003. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, **12**, 55–79.

Hyvärinen, Aapo, Karhunen, Juha, & Oja, Erkki. 2004. *Independent component analysis*. Vol. 46. John Wiley & Sons.

James, Lancelot F, Priebe, Carey E, & Marchette, David J. 2001. Consistent estimation of mixture complexity. *Annals of Statistics*, 1281–1296.

Jasra, A, Holmes, C. C., & A., Stephens D. 2005. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, **20**, 50–67.

Jutten, Christian, & Herault, Jeanny. 1991. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, **24**(1), 1–10.

Kagan, Abram. 2001. Another look at the Cramer-Rao inequality. *The American Statistician*, **55**(3), 211–212.

Kagan, Abram M, Rao, Calyampudi Radhakrishna, & Linnik, Yurij Vladimirovich. 1973. Characterization problems in mathematical statistics.

Kim, Hyune-Ju, Fay, Michael P, Feuer, Eric J, Midthune, Douglas N, *et al.* . 2000a. Permutation tests for joinpoint regression with applications to cancer rates. *Statistics in medicine*, **19**(3), 335–351.

Kim, Hyune-Ju, Fay, Michael P, Feuer, Eric J, Midthune, Douglas N, *et al.* . 2000b. Permutation tests for joinpoint regression with applications to cancer rates. *Statistics in medicine*, **19**(3), 335–351.

Laird, Nan M, & Ware, James H. 1982a. Random-effects models for longitudinal data. *Biometrics*, 963–974.

Laird, Nan M, & Ware, James H. 1982b. Random-effects models for longitudinal data. *Biometrics*, 963–974.

Lee, Te-Won, Girolami, Mark, & Sejnowski, Terrence J. 1999. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural computation*, **11**(2), 417–441.

Leroux, Brian G. 1992. Consistent estimation of a mixing distribution. *The Annals of Statistics*, **20**(3), 1350–1360.

Li, J., Ray, S., & Lindsay, B. G. 2007. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, **8**, 1687–1723.

Li, P, Chen, J, & Marriott, P. 2009. Non-finite Fisher information and homogeneity: an EM approach. *Biometrika*, 411–426.

Li, Pengfei, & Chen, Jiahua. 2010. Testing the order of a finite mixture. *Journal of the American Statistical Association*, **105**, 1084–1092.

Lindsay, B. G. 1995. Mixture Models: Theory, Geometry, and Applications. *In: NSF-CBMS Regional Conference Series in Probability and Statistics v 5*. Institure of Mathematical Statistics, Hayward, CA.

Lindsay, Bruce G, & Yao, Weixin. 2012. Fisher information matrix: A tool for dimension reduction, projection pursuit, independent component analysis, and more. *Canadian Journal of Statistics*, **40**(4), 712–730.

Liu, Jian, Wu, Shiying, & Zidek, James V. 1997. On segmented multivariate regression. *Statistica Sinica*, 497–525.

Lu, Zhaohua, & Song, Xinyuan. 2012. Finite mixture varying coefficient models for analyzing longitudinal heterogenous data. *Statistics in Medicine*, **31**(6), 544–560.

Marin, J.-M., Mengersen, K. L., & Robert, C. P. 2005. Bayesian modelling and inference on mixtures of distributions. *Pages 459–507 of:* Dey, D., & Rao, C.R. (eds), *Handbook of Statistics: Volume 25*. Amsterdam: North Holland.

Matteson, David S, & Tsay, Ruey S. 2017. Independent component analysis via distance covariance. *Journal of the American Statistical Association*, 1–16.

McLachlan, G. J., & Peel, D. 2000. *Finite Mixture Models*. New York: Wiley.

McLachlan, Geoffrey J. 1987. On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture. *Applied statistics*, 318–324.

Muggeo, Vito MR. 2003. Estimating regression models with unknown break-points. *Statistics in medicine*, **22**(19), 3055–3071.

Naik, Ganesh R, & Kumar, Dinesh K. 2011. An overview of independent component analysis and its applications. *Informatica*, **35**(1).

Papaioannou, Takis, & Ferentinos, Kosmas. 2005. On two forms of Fisher's measure of information. *Communications in Statistics-Theory and Methods*, **34**(7), 1461–1470.

Papastamoulis, P., & Iliopoulos, G. 2010. An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, **19**, 313–331.

Pappas, S Sp, Ekonomou, L, Karampelas, P, Karamousantas, DC, Katsikas, SK, Chatzarakis, GE, & Skafidas, PD. 2010a. Electricity demand load forecasting of the Hellenic power system using an ARMA model. *Electric Power Systems Research*, **80**(3), 256–264.

Pappas, S Sp, Ekonomou, L, Karampelas, P, Karamousantas, DC, Katsikas, SK, Chatzarakis, GE, & Skafidas, PD. 2010b. Electricity demand load forecasting of the Hellenic power system using an ARMA model. *Electric Power Systems Research*, **80**(3), 256–264.

Park, Dong C, El-Sharkawi, MA, Marks, RJ, Atlas, LE, & Damborg, MJ. 1991a. Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems*, **6**(2), 442–449.

Park, Dong C, El-Sharkawi, MA, Marks, RJ, Atlas, LE, & Damborg, MJ. 1991b. Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems*, **6**(2), 442–449.

Pastor, Roberto, & Guallar, Eliseo. 1998a. Use of two-segmented logistic regression to estimate change-points in epidemiologic studies. *American journal of epidemiology*, **148**(7), 631–642.

Pastor, Roberto, & Guallar, Eliseo. 1998b. Use of two-segmented logistic regression to estimate change-points in epidemiologic studies. *American journal of epidemiology*, **148**(7), 631–642.

Pearson, Karl. 1894. Contribution to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, **185**, 71–110.

Prodan, Ruxandra. 2008. Potential pitfalls in determining multiple structural changes with an application to purchasing power parity. *Journal of Business & Economic Statistics*, **26**(1), 50–65.

Ray, Surajit, & Lindsay, Bruce G. 2008. Model selection in high dimensions: a quadratic-risk-based approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**(1), 95–118.

Redner, R. A., & Walker, H. F. 1984. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.*, **26**, 195–239.

Richardson, S., & Green, P. J. 1997. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of Royal Statistical Association: Series B*, **59**, 731–792.

Rodríguez, C. E., & Walker, S. G. 2012. Label switching in Bayesian mixture models: Deterministic relabeling strategies. *Journal of Computational and Graphical Statistics*, **23**(1), 25–45.

Rousseeuw, Peter J. 1984. Least Median of Squares Regression. *Journal of the American Statistical Association*, **79**(388), 871–880.

Rousseeuw, Peter J, & Van Driessen, Katrien. 2006. Computing LTS regression for large data sets. *Data mining and knowledge discovery*, **12**(1), 29–45.

Scott, D. W. 1992. *Multivariate Density Estimation.* John Wiley & Sons, New York, Chichester.

Shao, Quanxi, & Campbell, NA. 2002a. Applications: Modelling trends in groundwater levels by segmented regression with constraints. *Australian & New Zealand Journal of Statistics*, **44**(2), 129–141.

Shao, Quanxi, & Campbell, NA. 2002b. Applications: Modelling trends in groundwater levels by segmented regression with constraints. *Australian & New Zealand Journal of Statistics*, **44**(2), 129–141.

Simpson, DG. 1997. Introduction to Rousseeuw (1984) Least Median of Squares Regression. *Pages 433–461 of: Breakthroughs in Statistics*. Springer.

Sperrin, M, Jaki, T, & Wit, E. 2010. Probabilistic relabelling strategies for the label switching problem in Bayesian mixture models. *Statistics and Computing*, **20**(3), 357–366.

Stephens, M. 1997. *Bayesian methods for mixtures of normal distributions*. Ph.D. thesis, Department of Statistics, University of Oxford.

Stephens, M. 2000. Dealing with label switching in mixture models. *Journal of Royal Statistical Association: Series B*, **62**, 795–809.

Strikholm, Birgit, & Teräsvirta, Timo. 2005. *Determing the number of regimes in a threshold autoregressive model using smooth transition autoregressions*. Tech. rept. SSE/EFI Working Paper Series in Economics and Finance.

Tan, Xianming, Shiyko, Mariya P, Li, Runze, Li, Yuelin, & Dierker, Lisa. 2012. A time-varying effect model for intensive longitudinal data. *Psychological methods*, **17**(1), 61.

Terrell, George R. 1995. A Fisher information test for Pearson-family membership. *Pages 230–234 of: Proceedings of the statistical computing section, joint statistical meetings, Orlando, Florida*.

Titterington, D. M., Smith, A. F. M., & Makov, U.E. 1985. *Statistical Analysis of Finite Mixture Distributions*. Wiley.

Toms, Judith D, & Lesperance, Mary L. 2003a. Piecewise regression: a tool for identifying ecological thresholds. *Ecology*, **84**(8), 2034–2041.

Toms, Judith D, & Lesperance, Mary L. 2003b. Piecewise regression: a tool for identifying ecological thresholds. *Ecology*, **84**(8), 2034–2041.

Valenzuela, Olga, Rojas, Ignacio, Rojas, Fernando, Pomares, Héctor, Herrera, Luis Javier, Guillén, Alberto, Marquez, Luisa, & Pasadas, Miguel. 2008a. Hybridization of intelligent techniques and ARIMA models for time series prediction. *Fuzzy Sets and Systems*, **159**(7), 821–845.

Valenzuela, Olga, Rojas, Ignacio, Rojas, Fernando, Pomares, Héctor, Herrera, Luis Javier, Guillén, Alberto, Marquez, Luisa, & Pasadas, Miguel. 2008b. Hybridization of intelligent techniques and ARIMA models for time series prediction. *Fuzzy Sets and Systems*, **159**(7), 821–845.

Vermont, J, Bosson, JL, François, P, Robert, C, Rueff, A, & Demongeot, J. 1991a. Strategies for graphical threshold determination. *Computer methods and programs in biomedicine*, **35**(2), 141–150.

Vermont, J, Bosson, JL, François, P, Robert, C, Rueff, A, & Demongeot, J. 1991b. Strategies for graphical threshold determination. *Computer methods and programs in biomedicine*, **35**(2), 141–150.

Wagner, Anita K, Soumerai, Stephen B, Zhang, Fang, & Ross-Degnan, Dennis. 2002a. Segmented regression analysis of interrupted time series studies in medication use research. *Journal of clinical pharmacy and therapeutics*, **27**(4), 299–309.

Wagner, Anita K, Soumerai, Stephen B, Zhang, Fang, & Ross-Degnan, Dennis. 2002b. Segmented regression analysis of interrupted time series studies in medication use research. *Journal of clinical pharmacy and therapeutics*, **27**(4), 299–309.

Wand, M. P., & Jones. 1995. *Kernel Smoothing*. Monographs on Statistics and Applied Probability, vol. 60. Chapman and Hall, London.

Wang, Shaoli, Yao, Weixin, & Huang, Mian. 2014. A note on the identifiability of nonparametric and semiparametric mixtures of GLMs. *Statistics & Probability Letters*, **93**, 41–45.

Wei, Tianshu, Zhu, Qi, & Yu, Nanpeng. 2016. Proactive demand participation of smart buildings in smart grid. *IEEE Transactions on Computers*, **65**(5), 1392–1406.

Wu, Xing, & Yu, Conglian. 2015. Estimation of the Mixtures of GLMs with Covariate-Dependent Mixing Proportions. To appear in*Communications in Statistics*.

Yao, W. 2012a. Bayesian mixture labeling and clustering. *Communications in Statistics - Theory and Methods*, **41**, 403–421.

Yao, W. 2012b. Model based labeling for mixture models. *Statistics and Computing*, **22**, 337–347.

Yao, W., & Lindsay, B. G. 2009. Bayesian mixture labeling by highest posterior density. *Journal of American Statistical Association*, **104**, 758–767.

Young, Derek S, & Hunter, David R. 2010. Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis*, **54**(10), 2253–2266.

Yu, Nanpeng, & Jie, Shi. 2017. Spatio-temporal modeling of electric loads. *Pages 1–6 of: 49th North American Power Symposium*. IEEE.

Yu, Nanpeng, Wei, Tianshu, & Zhu, Qi. 2015. From passive demand response to proactive demand participation. *Pages 1300–1306 of: Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. IEEE.

Appendix A

Robust Mixed-effects Segmented Regression Model

A.1 MLE Algorithm

```
library(mvtnorm)
library(lme4)
library(segmented)
library(matrixcalc)
mixed.MLE <- function(input.data,
initial, alpha, tol, max.iter){
    ## Inside function
    ran.eff <- function(b.vector, input.data){
```

```
    rep.num <- as.matrix(table(input.data$Subject))
    S.matrix <- cbind(rep(1,
    nrow(input.data)), input.data$S)
    random <- matrix(NA, nrow(input.data),1)
    sum <- 0
    for(i in 1:nrow(rep.num)){
        random[(sum + 1):(sum+rep.num[i]),] <-
        S.matrix[(sum + 1):(sum+rep.num[i]),]
        %*%(b.vector[i,])
        sum <- sum + rep.num[i]
    }
    return(random)
}
break.initial1 <- pmax(input.data$Z-
quantile(input.data$Z, initial[1]), 0)
break.initial2 <- pmax(input.data$Z-
quantile(input.data$Z, initial[2]), 0)
input.data <- data.frame(input.data,
Z.change1 = break.initial1,
Z.change2 = break.initial2,
Subject = Subject)
mixed.output.REML <- lmer(Y ~ X + Z +
```

```
Z.change1 + Z.change2 + (1 + S|Subject),
input.data, REML = FALSE)
psi1 <- initial[1]*max(input.data$Z)
psi2 <- initial[2]*max(input.data$Z)
convergence1 <- 10 ## converge of break-point
convergence2 <- 10
convergence3 <- 10 ## converge of random-effect
convergence4 <- 10
track <- 0
result <- list()
seg <- list()
index <- sample(seq(1, nrow(input.data), 1),
nrow(input.data)*alpha, replace = FALSE)
while (max(convergence1,
convergence2, convergence4) >
tol && track < max.iter){
    std.corr <- as.data.frame(
    VarCorr(mixed.output.REML))$sdcor
    std.corr[is.nan(std.corr)] = 0.5
    sigma.matrix <- matrix(c(std.corr[1]^2,
    std.corr[1]*std.corr[2]*std.corr[3],
    std.corr[1]*std.corr[2]*
```

```
std.corr[3], std.corr[2]^2), 2, 2)
b.vector <- as.matrix(
ranef(mixed.output.REML)$Subject)
random.effect <- ran.eff(b.vector, input.data)
## Create a new column to store the random effect
input.data1 <- data.frame
(input.data, random.effect)[index,]
## Use new b.vector to obtain new break.points
seg.regression <- segmented(lm
((Y-random.effect) ~
X + Z, data = input.data1),
seg.Z=~Z, psi=c(psi1, psi2),
control=seg.control(display=FALSE))
convergence1 <- abs(
seg.regression$psi[1,2] - psi1)
convergence2 <- abs(
seg.regression$psi[2,2] - psi2)
## Use new psi to replace the old psi
psi1 <- seg.regression$psi[1,2]
psi2 <- seg.regression$psi[2,2]
input.data$Z.change1 <- pmax(input.data$Z-psi1, 0)
input.data$Z.change2 <- pmax(input.data$Z-psi2, 0)
```

```
input.data1 <- input.data[index,]
## Calculate the random effect again: b.vector
mixed.output.REML <- lmer(Y ~ X + Z +
Z.change1 + Z.change2 +
(1 + S|Subject), input.data1, REML = FALSE)
std.corr <- as.data.frame(
VarCorr(mixed.output.REML))$sdcor
convergence3 <- max(abs(matrix
(c(std.corr[1]^2, std.corr[1]*std.corr[2]*std.corr[3],
std.corr[1]*std.corr[2]*std.corr[3],
std.corr[2]^2), 2, 2)
%*%solve(sigma.matrix) - diag(2)))
resid <- cbind(sort
((predict(mixed.output.REML,
input.data)-input.data$Y)^2,
decreasing = FALSE, index.return = TRUE)$x,
sort((predict(mixed.output.REML,
input.data)-input.data$Y)^2,
decreasing = FALSE, index.return = TRUE)$ix)
convergence4 <- 1 - length(intersect
(resid[1:(nrow(input.data)*alpha), 2], index))
/(nrow(input.data)*alpha)
```

```
        index <- resid[1:(nrow(input.data)*alpha), 2]
        ## Keep track
        track <- track + 1
        seg[[track]] <- seg.regression
        result[[track]] <- mixed.output.REML
    }
    return(list(seg[[track]], result[[track]]))
}
```

A.2 TLE Algorithm

```
library(mvtnorm)
library(lme4)
library(segmented)
library(matrixcalc)
mixed.MLE <- function(input.data,
initial, alpha, tol, max.iter){
    ## Inside function
    ran.eff <- function(b.vector, input.data){

        rep.num <- as.matrix(table(input.data$Subject))
        S.matrix <- cbind(rep(1,
        nrow(input.data)), input.data$S)
```

```
    random <- matrix(NA, nrow(input.data),1)
    sum <- 0
    for(i in 1:nrow(rep.num)){
        random[(sum + 1):(sum+rep.num[i]),] <-
        S.matrix[(sum + 1):(sum+rep.num[i]),]
        %*%(b.vector[i,])
        sum <- sum + rep.num[i]
    }
    return(random)
}
break.initial1 <- pmax(input.data$Z-
quantile(input.data$Z, initial[1]), 0)
break.initial2 <- pmax(input.data$Z-
quantile(input.data$Z, initial[2]), 0)
input.data <- data.frame(input.data,
Z.change1 = break.initial1,
Z.change2 = break.initial2,
Subject = Subject)
mixed.output.REML <- lmer(Y ~ X + Z +
Z.change1 + Z.change2 + (1 + S|Subject),
input.data, REML = FALSE)
psi1 <- initial[1]*max(input.data$Z)
```

```
psi2 <- initial[2]*max(input.data$Z)
convergence1 <- 10 ## converge of break-point
convergence2 <- 10
convergence3 <- 10 ## converge of random-effect
convergence4 <- 10
track <- 0
result <- list()
seg <- list()
index <- sample(seq(1, nrow(input.data), 1),
nrow(input.data)*alpha, replace = FALSE)
while (max(convergence1,
convergence2, convergence4) >
tol && track < max.iter){
    std.corr <- as.data.frame(
    VarCorr(mixed.output.REML))$sdcor
    std.corr[is.nan(std.corr)] = 0.5
    sigma.matrix <- matrix(c(std.corr[1]^2,
    std.corr[1]*std.corr[2]*std.corr[3],
    std.corr[1]*std.corr[2]*
    std.corr[3], std.corr[2]^2), 2, 2)
    b.vector <- as.matrix(
    ranef(mixed.output.REML)$Subject)
```

```
random.effect <- ran.eff(b.vector, input.data)
## Create a new column to store the random effect
input.data1 <- data.frame
(input.data, random.effect)[index,]
## Use new b.vector to obtain new break.points
seg.regression <- segmented(lm
((Y-random.effect) ~
X + Z, data = input.data1),
seg.Z=~Z, psi=c(psi1, psi2),
control=seg.control(display=FALSE))
convergence1 <- abs(
seg.regression$psi[1,2] - psi1)
convergence2 <- abs(
seg.regression$psi[2,2] - psi2)
## Use new psi to replace the old psi
psi1 <- seg.regression$psi[1,2]
psi2 <- seg.regression$psi[2,2]
input.data$Z.change1 <- pmax(input.data$Z-psi1, 0)
input.data$Z.change2 <- pmax(input.data$Z-psi2, 0)
input.data1 <- input.data[index,]
## Calculate the random effect again: b.vector
mixed.output.REML <- lmer(Y ~ X + Z +
```

```
Z.change1 + Z.change2 +
(1 + S|Subject), input.data1, REML = FALSE)
std.corr <- as.data.frame(
VarCorr(mixed.output.REML))$sdcor
convergence3 <- max(abs(matrix
(c(std.corr[1]^2, std.corr[1]*std.corr[2]*std.corr[3],
std.corr[1]*std.corr[2]*std.corr[3],
std.corr[2]^2), 2, 2)
%*%solve(sigma.matrix) - diag(2)))
resid <- cbind(sort
((predict(mixed.output.REML,
input.data)-input.data$Y)^2,
decreasing = FALSE, index.return = TRUE)$x,
sort((predict(mixed.output.REML,
input.data)-input.data$Y)^2,
decreasing = FALSE, index.return = TRUE)$ix)
convergence4 <- 1 - length(intersect
(resid[1:(nrow(input.data)*alpha), 2], index))
/(nrow(input.data)*alpha)
index <- resid[1:(nrow(input.data)*alpha), 2]
## Keep track
track <- track + 1
```

```
        seg[[track]] <- seg.regression
        result[[track]] <- mixed.output.REML
    }
    return(list(seg[[track]], result[[track]]))
}
```

Appendix B

Density Information Matrix with ICA Application

B.1 DIM-KDE

```
##Construct the empty list and A matrix##
library(fastICA)
library(ProDenICA)
library(MASS)
library(mvtnorm)
library(expm)
ppreduceyao<-function(X, nrow, ncol)  #jf2 code
  {
  nrow=dim(X)[1]; ncol=dim(X)[2]
```

```
cov<-cov(X) #Calculate sample covariance#
SqrtSigma<-eigen(cov)$vectors %*%
diag(sqrt(eigen(cov)$values)) %*% t(eigen(cov)$vectors)
Hopt<-((4/(ncol+2))^(1/(ncol+4)))*
SqrtSigma*(nrow^(-1/(ncol+4)))
###Search the circle around the area of Hopt##
Hopt.power.2<-Hopt%*%Hopt
Hopt.power.neg2<-solve(Hopt.power.2)
non.diag.one<-0 # set initial value of non-diagonal elements#
non.diag.two<-0
non.diag.three<-0
non.diag.four<-0
# Calculate the summation of non-diagonal part#
for (i in 1:(nrow-2)) {
  temp=dmvnorm(X[(i+1):nrow,],X[i,],2*Hopt.power.2)
  temp1=sum(temp)
  non.diag.one=non.diag.one+temp1
  non.diag.two=non.diag.two+(1/2)*
  Hopt.power.neg2*temp1-(1/4)*Hopt.power.neg2%*%
  ((X[i,]-t(X[(i+1):nrow,]))%*%
  (temp*t(X[i,]-t(X[(i+1):nrow,]))))
  %*%Hopt.power.neg2
```

```
    non.diag.three=non.diag.three+(1/2)
    *Hopt.power.2*temp1+(1/4)*(X[i,]+t(X[(i+1):nrow,]))
    %*%(temp*t(X[i,]+t(X[(i+1):nrow,])))
    non.diag.four=non.diag.four+(1/2)*temp%*%
    t(X[i,]+t(X[(i+1):nrow,]))
  }
    i=nrow-1
    temp=dmvnorm(X[(i+1):nrow,],X[i,],2*Hopt.power.2)
    non.diag.one=non.diag.one+temp
    non.diag.two=non.diag.two+(1/2)*
    Hopt.power.neg2*temp-(1/4)*temp*
    Hopt.power.neg2%*%
    ((X[i,]-X[(i+1):nrow,])%*%
    t(X[i,]-X[(i+1):nrow,]))%*%Hopt.power.neg2
    non.diag.three=non.diag.three+(1/2)*Hopt.power.2*
    temp+(1/4)*temp*(X[i,]+X[(i+1):nrow,])%*%
    t(X[i,]+X[(i+1):nrow,])
    non.diag.four=non.diag.four+(1/2)*
    temp*(X[i,]+X[(i+1):nrow,])
# Calculate the summation of diagonal part
kernel.part<-dmvnorm(rep(0,ncol),rep(0,ncol),2*Hopt.power.2)
diag.one<-nrow*kernel.part
```

```
deriv.part<-kernel.part*((1/2)*
Hopt.power.neg2-(1/4)*Hopt.power.neg2%*%
(rep(0,ncol))%*%t(rep(0,ncol))%*%Hopt.power.neg2)
diag.two<-nrow*deriv.part
# set initial value of diagonla elements#
diag.three<-0
diag.four<-0
diag.three=kernel.part*((1/2)*nrow*Hopt.power.2+t(X)%*%X)
diag.four=kernel.part*colSums(X)
part1<-(2*non.diag.one+diag.one)/(nrow^2)
part2<-(2*non.diag.two+diag.two)/(nrow^2)
part3<-(2*non.diag.three+diag.three)/(nrow^2)
part4<-(2*non.diag.four+diag.four)/(nrow^2)
Vf2<-part3/part1-t(part4/part1)%*%(part4/part1)
SqrtVf2<-eigen(Vf2)$vectors %*% diag(sqrt(eigen(Vf2)$values))
%*% t(eigen(Vf2)$vectors)
Jf2<-(SqrtVf2%*%part2%*%SqrtVf2)/part1
W<- solve(SqrtSigma)%*%eigen(Jf2)$vectors
output<-list(Jf2=Jf2,
SqrtSigma = SqrtSigma,
W = W, Vf2 = Vf2, Int = (part2/part1))
return(output)
```

```
}
```

B.2 DIM-GMM

```
DIMGMM <- function(X, nrow, ncol){
  cov<-cov(X) #Calculate sample covariance#
  SqrtSigma<-eigen(cov)$vectors %*%
  diag(sqrt(eigen(cov)$values)) %*%
  t(eigen(cov)$vectors)
  Comp <- tail(
  mclustBootstrapLRT(X, model = "EEI")$G, n = 1)
  dens <- densityMclust
  (X, G = Comp, modelNames = "EEI")
  Lambda <- dens$parameters$pro
  Mu <- dens$parameters$mean
  Sigma <- dens$parameters$variance$Sigma
  ########## Use f2 transformation method ##########
  equation1 <- 0
  equation2 <- 0
  equation3 <- 0
  equation4 <- 0
  Sigma.c <- Sigma/2
  for (i in 1:Comp) {
```

```
  for (j in 1:Comp){
    mu.c <- (Mu[, i] + Mu[, j])
    fx.square <- Lambda[i]*Lambda[j]*
    dmvnorm(Mu[, i], Mu[, j], 2*Sigma)
    equation1 <- equation1 + fx.square
    fx.der <- fx.square*(
    solve(Sigma)/2 + (1/4)*solve(Sigma)
                          %*%(Mu[, i] - Mu[, j])
                          %*%t(Mu[, i] - Mu[, j])
                          %*%solve(Sigma))
    equation2 <- equation2 + fx.der
    Vf1 <- fx.square*(Sigma.c + mu.c%*%t(mu.c))
    equation3 <- equation3 + Vf1
    Vf2 <- fx.square*(mu.c)
    equation4 <- equation4 + Vf2
  }
}
Vf.f2 <- equation3/equation1 -
(equation4/equation1)%*%t(equation4/equation1)
sqrtVf.f2 <- eigen(Vf.f2)$vectors
%*% diag(sqrt(eigen(Vf.f2)$values)) %*%
t(eigen(Vf.f2)$vectors)
```

```
  ############ Output matrix #########
   Jf2 <- (sqrtVf.f2%*%equation2%*%sqrtVf.f2)/equation1
  W.f2 <- solve(SqrtSigma)%*%eigen(Jf2)$vectors
   output<-list(Jf2 = Jf2,
   SqrtSigma = SqrtSigma, W.f2 =W.f2,
   Vf.f2 = Vf.f2, mixest = mixest)
}
```