이학박사 학위논문

Bayesian analysis of multivariate mixture models via factor analyzer

인자분해를 통한 다변량 혼합 모형의 베이지안 분석

2012년 8월

서울대학교 대학원

통계학과

김 재 석

Bayesian analysis of multivariate mixture models via factor analyzer

지도교수 김 용 대

이 논문을 이학박사 학위논문으로 제출함.

2012년 4월

서울대학교 대학원

통계학과

김 재 석

김재석의 이학박사 학위논문을 인준함.

2012년 6월

위 원 장 :	전	종	우 (인)
부 위원장 :	김	용	대 (인)
위 원 :	박	병	욱 (인)
위 원 :	임	요	한 (인)
위 원 :	이	태	영 (인)

Bayesian analysis of multivariate mixture models via factor analyzer

By

Jaeseok Kim

A Thesis

Submitted in fulfillment of the requirements

for the degree of

Doctor of Philosophy

in Statistics

Department of Statistics

College of Natural Sciences

Seoul National University

August, 2012

Abstract

We consider a new Bayesian finite mixture model for multivariate data. A problem is to estimate the covariance matrix since the number of parameters for the covariance matrix is squarely proportional to the dimension of data. Also, the inverting large dimensional covariate, which is necessary for MCMC algorithms, is very time consuming and practically almost prohibited. In this thesis, we propose a way of reducing the parameters in the covariance matrices by use of the factor model. That is, the dependence structure of each component is assumed to be represented by linear combinations of factors. To simply the model and improve interpretability further, we allow some factors can be shared across the components. From numerical studies, we confirmed that our method well perform with different component covariance structure.

Keywords: Bayesian, mixture model, factor analysis, clustering, RJMCMC

Student Number: 2005 − 20294

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Clustering or cluster analysis have been studied extensively by both a theoretical and a practical perspective for over 100 years. Clustering divides a collection of objects into groups (called cluster) such that those within each cluster are more similar related to one another than objects assigned to different clusters. There are many applications of clustering including pattern recognition, image segmentation, and document retrieval. The existing clustering approaches can be roughly classified into two types, distance based and model based clustering.

The most commonly discussed distinction among different types of distance

based clusterings is whether the set of clusters is nested or unnested, or in more traditional terminology, hierarchical or partitional. Partitional methods divide the data set into a number of clusters predesignated by the analyst that minimize a certain criterion function. The K-means algorithm is such a method. Hierarchical clustering approaches produce a hierarchy of clusters from small clusters of very similar items to large clusters that include more dissimilar items. Those usually create a graphical output known as a dendrogram that shows this hierarchical clustering structure.

Distance based clustering algorithms are not easy to decide a number of clusters. Even though various strategies for simultaneous determination of the number of clusters have been proposed by Bock [1996], Engelman and Hartigan [1969], optimally deciding the number of clusters is still open question.

Model-based clustering algorithms attempt to optimize the fit between the given data and some statistical model. In model-based clustering, it is assumed that objects are generated by a mixture of underlying probability distributions in which each distribution represents a different cluster. The most prominent approach in model-based clustering is known as finite mixture models.

Generally, finite mixture models are useful for modeling of a heterogeneous data into homogeneous clusters. In addition, due to the large class of functions that can be approximated by a mixture model, they are attractive for describ-

ing nonstandard distributions. Finite mixture models date back to the work of Newcomb [1886] and Pearson [1894], but methodological advances in computational methods such as maximum likelihood approach Baum et al. [1970] and the expectation-maximization(EM) algorithm(Dempster et al. [1977]) have amply expanded the areas of their applications.

However, difficulties often arise in the application of finite mixture models. Deciding the optimal number of components is important to ensure an efficient and accurate estimation. To estimate the number of components, there have been several literature such as Akaike [1973], Schwarz [1978] and Spiegelhalter et al. [2002], but is not fully satisfactory. In some situations, the likelihood function may be unbounded(Aitkin [2001]). Also, the EM algorithm is sensitive to the initialization and is a local optimization procedure.

A Bayesian method of finite mixture models usually leads to intractable calculations, before Markov chain Monte Carlo(MCMC) methods were available. Diebolt and Robert [1994] suggested a data augmentation Gibbs sampler approach for finite mixture models with known number of components. When the number of components is unknown, several models and algorithms such as Richardson and Green [1997](reversible jump MCMC), Stephens [2000](birth and death MCMC) and Escobar and West [1995](Mixture of Dirichlet process) have been proposed. Among these, the reversible jump MCMC has received

much attention.

The reversible jump Markov chain Monte carlo (RJMCMC) sampler introduced by Green [1995], provides a general framework for MCMC simulation in which the dimension of the parameter space can vary between iterated of the Markov chain. The reversible jump methodology has been applied to a wide range of model choice problems, including change point analysis(Green [1995], Fan and Brooks [2000]) , variable selection(Smith and Kohn [1996], Nott and Leonte [2004]), time series models(Brooks et al. [2003], Vermaak et al. [2004]) and finite mixture model (Richardson and Green [1997], Dellaportas and Papageorgiou [2006]).

The RJMCMC algorithm of Richardson and Green [1997] is based on a series of split-combine and birth-death moves. The split move suggests splitting a randomly chosen component into two new components, whereas the combine move picks up randomly two components and proposes to merge them to one. But its implementation for multivariate data is not straightforward because of the mathematical complexity of the split and combine moves. In particular, the calculation of the Jacobian for the proposed move requires messy algebraic calculations. Dellaportas and Papageorgiou [2006] proposed split-combine moves using the eigenvalue decomposition of the component covariance matrix of multivariate data. However, the performance of their approach is degraded

rapidly when the dimension of data increase due to over-parametrization.

The number of parameters to be estimated is squarely proportional to the dimension of data due to the covariance matrices. One way of reducing the number of parameters is to delete noisy variables. Methods of variable selection with finite mixture models can be found in Tadesse et al. [2005], Kim et al. [2006], Pan and Shen [2007], Xie et al. [2008].

An alternative way of reducing the parameters in multivariate finite mixture models is to make a model for the covariance matrices. Factor models are popularly used to model the covariance matrices. See Ghahramani and Hinton [1996], McLachlan and Peel [2000] and Baek et al. [2010] for related approaches.

In this thesis, we propose a new Bayesian finite mixture model for multivariate data. We apply a factor model to the covariance matrices. Advantages of our model compared to other finite mixture models with factor models are (1) the number of factors in each component is estimated automatically, and (2) clusters can share factors so we can reduce the number of parameters further and (3) the range of the model is very large so that it covers from the simplest model (the equal diagonal covariance matrices) to the most complex one (unspecified covariance matrices). By choosing the prior parameters accordingly, we can find a best parsimonious model in between the simplest and most complex finite mixture models.

1.2 Outline of the thesis

The thesis is organized as follows. We briefly review various methods for finite mixture models in chapter 2. Our proposed Bayesian model is explained in chapter 3, and results of numerical studies are given in chapter 4. Concluding remarks follow in chapter 5.

Chapter 2

Reviews

In this chapter, we review various methods and algorithms for finite mixture models.

2.1 Univariate case

Mixture models are useful in describing a wide variety of random phenomena because of their inherent flexibility Titterington et al. [1985]. The probability density function of a mixture model with K components is given as

$$f(y) = \sum_{k=1}^{K} w_k f_k(y),$$

where the densities f_k are known up to parameters and the proportions $0 < w_k < 1$ satisfy $\sum_{k=1}^{K} w_k = 1$. If the components $f_k(\cdot)$ are normal densities with

mean μ_k and variance σ_k^2, the finite normal mixture model is given by

$$f(y|\boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \phi(y|\mu_k, \sigma_k^2),$$

where

$$\phi(y|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\{-\frac{1}{2}\frac{(y-\mu_k)^2}{\sigma_k^2}\},$$

and $\boldsymbol{\theta} = \{\theta_k = (w_k, \mu_k, \sigma_k^2), k = 1, \ldots, K\}$.

2.1.1 EM algorithm

For given data $y_1, \ldots, y_n$ from $f(y)$, the likelihood of the normal mixture model with K components is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} w_k \phi(y_i|\mu_k, \sigma_k^2).$$

The maximum likelihood estimator of $\boldsymbol{\theta}$ is defined to be

$$\hat{\boldsymbol{\theta}}^{ML} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}),$$

when this exists. A number of numerical algorithms have been developed for maximizing the log-likelihood function, because the explicit expression for the maximum likelihood estimator is typically not available. Among them, it has received much attention to use the EM algorithm of Dempster et al. [1977]

by casting the problem in the framework of incomplete data. Define z_{ik} as an indicator variable whether y_i is from the k-th cluster, that is

$$z_{ik} = \begin{cases} 1 & \text{if } y_i \text{ belongs to the } k\text{th cluster} \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that $\mathbf{z}_i = \{z_{i1}, \ldots, z_{iK}\}$ are independent and identically distributed according to a multinomial distribution of one draw from K categories with probabilities $w_1, \ldots, w_K$, and that the density of y_i given $\mathbf{z}_i$ is given by

$$\prod_{k=1}^{K} \phi(y_i|\mu_k, \sigma_k^2)^{z_{ik}},$$

the resulting complete data log-likelihood is

$$\log L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \Big\{ \log w_k + \log \phi(y_i|\mu_k, \sigma_k^2) \Big\}.$$

The EM algorithm is an iterative procedure consisting of two alternating steps. With data and initial guess $\boldsymbol{\theta}^{(0)}$ for $\hat{\boldsymbol{\theta}}^{ML}$, a sequence $\{\boldsymbol{\theta}^{(t)}\}$ are generated from the following double-step that creates $\boldsymbol{\theta}^{(t+1)}$ from $\boldsymbol{\theta}^{(t)}$. First, for the E-step of the EM algorithm all that we have to compute are the expectations of the indicator variables, z_{ik}. Given $\boldsymbol{\theta}^{(t)}$, we obtain

$$\gamma_{ik}^{(t+1)} = \frac{w_k^{(t)} \phi(y_i|\mu_k^{(t)}, \sigma_k^{2(t)})}{\sum_{k=1}^{K} w_k^{(t)} \phi(y_i|\mu_k^{(t)}, \sigma_k^{2(t)})}, \tag{2.1}$$

where $\gamma_{ik} = E(z_{ik}|y_i, \mu_k, \sigma_k^2, w_k)$ for each i and k. In the M-step of the EM algorithm, we update the parameters as

$$w_k^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)}}{n}, \tag{2.2}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t+1)} y_i}{\sum_{i=1}^n \gamma_{ik}^{(t+1)}}, \sigma_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t+1)} (y_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^n \gamma_{ik}^{(t+1)}}. \tag{2.3}$$

The EM algorithm requires two inputs, the number of components and initialization of the parameters. To estimate the number of components, there have been several literature including Akaike [1973], Schwarz [1978] and Spiegelhalter et al. [2002]. For initialization of the parameters, see Yeung et al. [2001], Fraley and Raftery [2002] and Fraley and Raftery [2006].

2.1.2 Bayesian method

Parameters, $\boldsymbol{\theta}$, for finite mixture model can be estimated with Bayesian approaches. Let $L_n(y_1, \ldots, y_n|\boldsymbol{\theta})$ be the likelihood function of $\boldsymbol{\theta}$. Assuming that a prior distribution $p(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$ is available, the posterior density $p(\boldsymbol{\theta}|y_1, \ldots, y_n)$ can be obtained by

$$p(\boldsymbol{\theta}|y_1, \ldots, y_n) \propto L_n(y_1, \ldots, y_n|\boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

using the Bayes' theorem.

Like the EM algorithm, we introduce a latent variable $M_i \in \{1, \ldots, K\}$ such that

$$P(M_i = k) = w_k, \ i = 1, \ldots, n, \ k = 1, \ldots, K.$$

Then, the joint density function can be written as

$$P(\mathbf{y}, \mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, M) = W(\mathbf{w}) \prod_{i=1}^{n} w_{M_i} \prod_{k=1}^{K} \left\{ \pi_\mu(\mu) \pi_{\sigma^2}(\sigma^2) \prod_{M_i=k} \phi(y_i|\mu_k, \sigma_k) \right\},$$

where $\mathbf{y} = (y_1, \ldots, y_n)$, $\mathbf{w} = (w_1, \ldots, w_K)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_K^2)$, $M = (M_1, \ldots, M_n)$ and $W(\cdot), \pi_\mu(\cdot), \pi_{\sigma^2}(\cdot)$ are prior densities of the parameter $\mathbf{w}$, μ_k, σ_k^2, respectively. We review the algorithms of Diebolt and Robert [1994] for fixed components number and Richardson and Green [1997] for unknown components number.

Let prior distributions be that $\mathbf{w}, \mu_k, \sigma_k^2$ are drawn independently from

$$\mathbf{w} \sim Dir(\alpha_w, \ldots, \alpha_w), \ \mu_k \sim N(0, \sigma_\mu^2) \text{ and } \sigma_k^2 \sim IG(\alpha_e, \beta_e),$$

where $Dir(\alpha_1, \ldots, \alpha_K)$ denotes Dirichlet distribution with parameter $\alpha_1, \ldots, \alpha_K$, $N(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, and $IG(a, b)$ is inverse gamma distribution with shape parameter a and scale parameter b.

The Gibbs sampling algorithm of Diebolt and Robert [1994], which is widely used in practice, is based on simulation of $\mathbf{w}, M, \boldsymbol{\mu}, \boldsymbol{\sigma}$ iteratively from their conditional posteriors. The Gibbs sampling approach can be summarized as follows

- The configuration M_i is randomly chosen from $\{1, \ldots, K\}$ with probability

$$P(M_i|\text{other}) \propto w_k \phi(y_i|\mu_k, \sigma_k^2)$$

- The mean parameter μ_k is generated

$$\mu_k|\text{other} \sim N(\eta_k^2 \frac{\sum_{i \in N_k} y_i}{\sigma_k^2}, \eta_k^2)$$

where $N_k = \{i : M_i = k\}$, $n_k = |N_k|$ and $\eta_k^2 = (\frac{n_k}{\sigma_k} + \frac{1}{\sigma_\mu^2})^{-1}$.

- The variance parameter σ_k^2 is generated

$$\sigma_k^2|\text{other} \sim IG(\alpha_e + \frac{n_k}{2}, \beta_e + \frac{1}{2}\sum_{i \in N_k}(y_i - \mu_k)^2).$$

- The weight is generated

$$\mathbf{w}|\text{other} \sim Dir(\alpha_w + n_1, \ldots, \alpha_w + n_k).$$

When K is unknown, the dimension of the parameter space is random and trans-dimensional methods are needed. Green [1995] proposed a reversible jump MCMC which is an extension of the Metropolis-Hastings algorithm that jumps between the parameter spaces of differing dimensions. Richardson and Green [1997] used this RJMCMC algorithm for the normal mixture model with unknown number of components. The RJMCMC for the normal mixture model mainly consists of the splitting/combining of components. Among the existing components a randomly chosen one is a candidate to split into two in the split move. The resulting mixture will be increased by one in the total number of components. the RJMCMC for the normal mixture model can be summarized as follows.

- Parameters M_i, μ_k, σ_k^2, w_k are generated by the Gibbs sampling algorithm with fixed components number.

- K is updated by

  - Splitting one component into two or combining two into one

  - Birth or death of an empty component

For details of the RJMCMC for unknown K see Richardson and Green [1997].

2.2 Multivariate case

Given data $\mathbf{y}_1, \ldots, \mathbf{y}_n$ where $\mathbf{y}_i \in R^p$, the likelihood for a normal mixture model is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} w_k \phi_p(\mathbf{y}_i | \boldsymbol{\mu}_k, \Sigma_k),$$

where ϕ_p is the p dimension normal density with mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k = (w_k, \boldsymbol{\mu}_k, \Sigma_k), k = 1, \ldots, K\}$. Here, the geometric features of the clusters are determined by the covariances Σ_k. For example, when $\Sigma_k = \lambda I$, all cluster are spherical and of the same size, or all cluster have the same geometry but need not be spherical when $\Sigma_k = \Sigma$. Banfield and Raftery [1993] proposed a general framework for covariance matrices through

the eigenvalue decomposition in the following form

$$\Sigma_k = \lambda_k D_k A_k D_k^T,$$

where D_k is the matrix of eigenvectors, A_k is a diagonal matrix whose elements are proportional to the eigenvalues, and λ_k is an associated constant of proportionality. In the EM algorithm for the multivariate normal mixture model, the E-step is the same as that for the univariate normal mixture model except ϕ being replaced by ϕ_k in (2.1). For the M-step, estimates of the means and proportions have the same forms as (2.2) and (2.3). Computation of the covariance estimate $\hat{\Sigma}_k$ depends on their constraint. For example, we can let D_k are the same but A_k are different to reduce the number of parameters. For details of the M-step for Σ_k, see Celeux and Govaert [1995].

Full Bayesian estimation of the multivariate normal mixture with unknown number of components has been attempted by Stephens [2000] who proposed an approach based on continuous time Markov birth-death processes and applied it to a bivariate setting. Some limitations of this method, and comparisons with the RJMCMC algorithm can be found in Cappé et al. [2003]. Zhang et al. [2004] suggested a RJMCMC algorithm with all covariance matrices of the mixtures are restricted to share the same eigenvector matrix. Dellaportas and Papageorgiou [2006] proposed a RJMCMC algorithm using the spectral decomposition of a covariance matrix. Here, we briefly review split move of

Dellaportas and Papageorgiou [2006] with two dimensions.

- Let w_{k^*}, $\boldsymbol{\mu}_{k^*}$, Σ_{k^*} be weight, mean and covariance matrix of component to be split.

- New weights w_{k_1} and w_{k_2} are generated by

$$w_{k_1} = u_1 w_{k^*},\ w_{k_2} = (1-u_1)w_{k^*},$$

  where $u_1 \sim Beta(2,2)$.

- New means $\boldsymbol{\mu}_{k_1}$ and $\boldsymbol{\mu}_{k_2}$ are generated by

$$\boldsymbol{\mu}_{k_1} = \boldsymbol{\mu}_{k^*} - (u_2^1\sqrt{\lambda_{k^*}^1}V_{k^*}^1 + u_2^2\sqrt{\lambda_{k^*}^2}V_{k^*}^2)\sqrt{\frac{w_{k_2}}{w_{k_1}}}$$
$$\boldsymbol{\mu}_{k_2} = \boldsymbol{\mu}_{k^*} + (u_2^1\sqrt{\lambda_{k^*}^1}V_{k^*}^1 + u_2^2\sqrt{\lambda_{k^*}^2}V_{k^*}^2)\sqrt{\frac{w_{k_1}}{w_{k_2}}},$$

  where $u_2^1 \sim Beta(2,2)$,$u_2^2 \sim U(-1,1)$ and $\lambda_{k^*}^i$, $V_{k^*}^i$ are i-th eigenvalue and eigenvector of $\Sigma_{k^*} = V_{k^*}\Lambda_{k^*}V_{k^*}^T$, respectively.

- Covariance matrices :

$$\Sigma_{k_1} = V_{k_1}\Lambda_{k_1}V_{k_1}^T$$

$$\Sigma_{k_2} = V_{k_2}\Lambda_{k_2}V_{k_2}^T,$$

  where

$$\Lambda_{k_1} = \begin{pmatrix} \lambda_{k_1}^1 & 0 \\ 0 & \lambda_{k_1}^2 \end{pmatrix} = \begin{pmatrix} u_3^1 & 0 \\ 0 & u_3^2 \end{pmatrix} \begin{pmatrix} 1-(u_2^1)^2 & 0 \\ 0 & 1-(u_2^2)^2 \end{pmatrix} \Lambda_{k^*}\frac{w_{k^*}}{w_{k_1}}$$

$$\Lambda_{k_2} = \begin{pmatrix} \lambda_{k_2}^1 & 0 \\ 0 & \lambda_{k_2}^2 \end{pmatrix} = \begin{pmatrix} 1-u_3^1 & 0 \\ 0 & 1-u_3^2 \end{pmatrix} \begin{pmatrix} 1-(u_2^1)^2 & 0 \\ 0 & 1-(u_2^2)^2 \end{pmatrix} \Lambda_{k^*} \frac{w_{k^*}}{w_{k_1}},$$

$u_3^1 \sim Beta(1,2)$, $u_3^2 \sim U(0,1)$,

$$V_{k_1} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} V_{k^*}$$

$$V_{k_2} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} V_{k^*}$$

with $\theta \sim U(0, \frac{\pi}{2})$.

To split mean vector, they proposed an approach to use eigenvalues and eigenvectors of covariance matrix. For covariance matrix, the splitting eigenvalues is presented on extending Richardson and Green [1997] approach and the splitting eigenvectors are produced by rotating the old eigenvector. A more detailed technical description is presented in Dellaportas and Papageorgiou [2006]

2.3 Mixture of factor analyzers

The K component multivariate normal mixture model with unrestricted component covariance matrices has $d = \frac{K}{2}p(p+1)$ parameters for the covariance

matrices Σ_k. Banfield and Raftery [1993] proposed the spectral decomposition for Σ_k. However, if p is large relative to the sample size n, it may not be possible to use this decomposition to infer an appropriate model for the component-covariance matrices. A common approach for reducing parameters in the forms for the component covariance matrices is to adopt the mixture of factor analyzer model, as considered in Ghahramani and Hinton [1996] and McLachlan and Peel [2000]. The distribution of $\mathbf{y}_i$ can be modelled as

$$\mathbf{y}_i = \boldsymbol{\mu}_k + G_k Z_i^k + \boldsymbol{\epsilon}_i^k, \quad \text{with probability, } w_k,$$

where G_k is $p \times q$ factor loading matrix, Z_i^k is q-dimensional factor and $\boldsymbol{\epsilon}_i^k$ are distributed independently by $N_p(0, \Sigma_{\epsilon k})$ and $\Sigma_{\epsilon k} = \text{diag}(\sigma_{\epsilon k1}^2, \ldots, \sigma_{\epsilon kp}^2)$. Thus the k-th covariance matrix Σ_k has the form

$$\Sigma_k = G_k G_k^T + \Sigma_{\epsilon k}$$

We shall refer to this approach as the MFA (Mixture of Factor Analyzers).

If p is large and/or K is not small, the number of parameters of the MFA model still might not be manageable. Baek et al. [2010] proposed the use of mixture of factor analyzer with common component factor loadings (MCFA). The MCFA model is considered as a multivariate normal mixture model with the restrictions

$$\boldsymbol{\mu}_k = A\xi_k, \ \Sigma_k = A\Omega_k A^T + \Sigma_{\epsilon k}$$

where A is $p \times q$ matrix, ξ_k is a q dimensional vector and Ω_k is a $q \times q$ positive definite symmetric matrix. Then, we can rewrite as

$$\begin{aligned} \mathbf{y}_i &= AZ_i^{k*} + \boldsymbol{\epsilon}_i^k = A\xi_k + A(Z_i^{k*} - \xi_k) + \boldsymbol{\epsilon}_i^k \\ &= \boldsymbol{\mu}_k + AK_kK_k^{-1}(Z_i^{k*} - \xi_k) + \boldsymbol{\epsilon}_i^k \\ &= \boldsymbol{\mu}_k + G_kZ_i^k + \boldsymbol{\epsilon}_i^k, \end{aligned}$$

where $Z_i^{k*} \sim N(\xi_k, \Omega_k)$, $G_k = AK_k$ and $Z_i^k = K_k^{-1}(Z_i^{k*} - \xi_k)$. The covariance matrix of Z_i^k is equal to I_q, since K_k can be chosen so that

$$K_k^{-1}\Omega_kK_k^{-1T} = I_q.$$

To determine the number of factor as well as the number of components, they used the Bayesian information criterion(BIC) of Schwarz [1978]. For details of mixture of factor analyzers see Baek et al. [2010].

Chapter 3

Bayesian factor clustering

3.1 Model

We consider a finite multivariate normal mixture model. Let $\mathbf{y}$ be a p-dimensional random vector. The density of mixture of K multivariate normal distributions is given by

$$f(\mathbf{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \phi_p(\mathbf{y}|\boldsymbol{\mu}_k, \Sigma_k).$$

For reducing parameters in the forms for the component covariance matrices, we adopt the MFA model, as considered in Ghahramani (1997) and McLachlan (2000). The distribution of $\mathbf{y}$ can be modelled as

$$\mathbf{y} = \boldsymbol{\mu}_k + G_k Z^k + \boldsymbol{\epsilon}^k, \quad \text{with probability, } w_k,$$

where G_k is $p \times q$ factor loading matrix, Z^k is q-dimensional factor and $\boldsymbol{\epsilon}^k$ are distributed independently by $N_p(0, \Sigma_{\epsilon k})$ and $\Sigma_{\epsilon k} = \text{diag}(\sigma^2_{\epsilon k1}, \ldots, \sigma^2_{\epsilon kp})$. Thus the k-th covariance matrix Σ_k has the form

$$\Sigma_k = G_k G_k^T + \Sigma_{\epsilon k}$$

The covariance matrices represent the geometric features of clusters such as volume, shape and orientation. Therefore, when Σ_k are completely unspecified, each cluster can have a different geometrical structure. But it requires the estimation of very large number of parameters. To reduce the number of parameters, we may assume that an eigenvector of the covariance matrix of some cluster may be similarly to that of other cluster. In other words, there are factor loading vectors which affect multiple clusters. To model this assumption, we allow some parameters in G_k can be shared by multiple clusters. For this purpose, we introduce latent variables $c_{kj} \in \{0, 1, \ldots, K\}$, $k = 1, \ldots, K$, $l = 1, \ldots, p$, that indicate which clusters share the j-th factor in G_k. Using c_{kj}, we construct $\mathbf{g}_k^j = (g_{j1}^k, \ldots, g_{jp}^k)$, the j-th column in G_k, as follwings

$$\begin{aligned}
\mathbf{g}_k^j &= \mathbf{0}, \qquad \text{if } c_{kj} = 0, \\
\mathbf{g}_k^j &\neq \mathbf{0}, \qquad \text{if } c_{kj} > 0, \\
\mathbf{g}_k^j &= \mathbf{g}_{k^*}^j, \qquad \text{if } c_{kj} = c_{k^*j} > 0.
\end{aligned}$$

Given $\mathbf{c}_k$, let $G_k = (\mathbf{g}_k^1, \ldots, \mathbf{g}_k^p)$ be $p \times q_k$ factor loading matrix, where $q_k =$

$|\{j; c_{kj} > 0\}|$. Thus the dimensions of G_k can differ across the clusters.

Let $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be p-dimensional observed samples where $\mathbf{y}_i = (y_{i1}, \ldots, y_{ip})$. Bayesian clustering algorithms can be formulated in terms of multivariate normal mixture model with K components as follows

$$\mathbf{y}_i = \sum_{k=1}^{K} (\boldsymbol{\mu}_k + G_k Z_i^k + \boldsymbol{\epsilon}_i^k) I(M_i = k),\ i = 1, \ldots, n,$$

Then the conditional probability of $\mathbf{y}_i$ given $\mathbf{R} = \{w_k, \boldsymbol{\mu}_k, G_k, Z_i^k, \sigma_{\epsilon kj}^2, \mathbf{c}_k = (c_{k1}, \ldots, c_{kp}), i = 1, \ldots, n, j = 1, \ldots, p, k = 1, \ldots, K\}$ with $M_i = k$ is the normal density as

$$P(\mathbf{y}_i | \mathbf{R}, M_i = k) = \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma_{\epsilon kj}^2}} \exp\left(-\frac{\epsilon_{ikj}^2}{2\sigma_{\epsilon kj}^2}\right),$$

where $\epsilon_{ijk} = y_{ij} - \mu_{kj} - \mathbf{g}_j^k Z_i^k$, $\mathbf{g}_j^k = (g_{jl}^k, l = 1, \ldots, q_k)$. Thus the full likelihood function is

$$P(\mathbf{y}|\mathbf{R}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left\{ \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma_{\epsilon kj}^2}} \exp\left(-\frac{\epsilon_{ikj}^2}{2\sigma_{\epsilon kj}^2}\right) \right\}^{M_i = k},$$

where $\mathbf{y} = (\mathbf{y}_1, \cdots \mathbf{y}_n)$.

3.2 Priors

Bayesian formulation requires prior distributions for the model parameters. First, we assume a priori that K follows the Poisson distribution with mean

a_{nc}. We adopt the conjugate priors for w_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}^2_{\epsilon k} = (\sigma^2_{\epsilon k1}, \ldots, \sigma^2_{\epsilon kp})$, $k = 1, \ldots, K$.

Given K, the prior on the weight $\mathbf{w} = (w_1, \ldots, w_K)$ is taken as a Dirichlet distribution with the density function

$$p(\mathbf{w}|K) = \frac{\Gamma(K\alpha_w)}{\Gamma(\alpha_w)^K} \prod_{k=1}^{K} w_k^{\alpha_w - 1},$$

where $\alpha_w > 0$ is a prior parameter.

Given K and $\mathbf{w}$, the prior on the allocation vector $M = (M_1, \ldots, M_n)$ is given by

$$p(M|\mathbf{w}, K) = \prod_{i=1}^{n} w_{M_i}.$$

Given K, μ_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, p$, are independent normal random variables with mean 0 and variance σ^2_μ and σ^2_μ follows the inverse gamma distribution with shape parameter a_μ and scale parameter b_μ.

Given K, $\sigma^2_{\epsilon kj}$, $k = 1, \ldots, K$, $j = 1, \ldots, p$, follow independent the inverse gamma distribution with shape parameter a_ϵ and scale parameter b_ϵ.

For prior of $\mathbf{c}$, we propose that $\mathbf{c}^j = (c_{1j}, \ldots, c_{Kj})$, $j = 1, \ldots, p$, are a sample path of the modified Chinese restaurant process(MCRP), which is explained in the followings.

- There is a restaurant with $K + 1$ tables.

- The tables are chosen by customers according to the following random process.
  - The first customer chooses
    - the labeled 0 table with probability $\frac{a_0}{a_0+a_c}$, where a_0, a_c are MCRP parameters.
    - the first unoccupied table with probability $\frac{a_c}{a_0+a_c}$.
  - the new k-th customer chooses
    - the labeled 0 table with probability $\frac{a_0+m}{a_c+a_0+n-1}$
    - an occupied and labeled nonzero table with probability $\frac{m}{a_c+a_0+n-1}$.
    - the first unoccupied table with probability $\frac{a_c}{a_c+a_0+n-1}$,

    where m is the number of people sitting at that table.

The parameters in the MCRP are a_0 and a_c. By choosing them accordingly, we can control the structure of component covariance matrices. For example, if we let $a_c = 0$, the component covariance matrices become diagonal. If we let $a_0 = 0$ and a_c be large, the component covariance matrices are completely unspecified. If we let $a_0 = 0$ and a_c be small, the component covariance matrices become all equal.

Generally, Gibbs samplers with conjugate prior, $N(0, \sigma_g)$, for g_{jl}^k are poorly behaved. Here, we used a parameter expansion method on the factor loading

proposed by Ghosh and Dunson [2009]. Given K and $\mathbf{c}_k$, expansion factor loading g_{jl}^{k*} are drawn independently $N(0,1)$. Expansion factor Z_{il}^{k*}, $l = 1, \ldots, q_k$ are drawn independently from $N(0, \psi_l)$ and ψ_l follows an inverse gamma distribution with shape parameter a_g and scale parameter b_g. Then, we use the following transformation

$$g_{jl}^{k} = \text{sgn}(g_{ll}^{k*}) g_{jl}^{k*} \psi_l^{1/2}, \quad Z_{il}^{k} = \text{sgn}(g_{ll}^{k*}) \psi_l^{-1/2} Z_{il}^{k*},$$

where $\text{sgn}(x) = -1$ if $x < 0$ and $\text{sgn}(x) = 1$, otherwise. Then we obtain a t prior distribution for the off-diagonal elements of G_k and half-t prior distributions for the diagonal elements of G_k. A more detailed technical description is presented in Ghosh and Dunson [2009].

3.3 RJMCMC

For our multivariate normal mixture model, nine move types are involved in the RJMCMC:

(a) Update the allocation vector $M_1, \ldots, M_n$.

(b) Update the weight $w_1, \ldots, w_K$

(c) Update mean parameter μ_{kj}, $k = 1, ;K$, $j = 1, \ldots, p$

(d) Update variance parameter $\sigma^2_{\epsilon kj}$, $k = 1, ;K$, $j = 1, \ldots, p$

(e) Update latent variable c_{kj}, $k = 1, \dot{,} K$, $j = 1, \ldots, p$

(f) Update factor loading $\mathbf{g}_j^k$, $k = 1, \dot{,} K$, $j = 1, \ldots, p$

(g) Update factor Z_i^k, $k = 1, \dot{,} K$, $i = 1, \ldots, n$

(h) Split or Combine : Split a cluster into two or combine tow clusters into one.

(i) Birth or Death : Generate a new cluster or delete a cluster.

Move types (a), (b), (c) and (d) are the same as those in Diebolt and Robert [1994]. Move type (e) is easily done by using Chinese restaurant process. Move types (f) and (g) may use Ghosh and Dunson [2009] approach, but we update column of factor loading matrix instead row of factor loading matrix. The only randomness is the random choice between splitting or combining in move (h), and birth or death in move(i).

In move type (a) (the allocation vector), M_i is randomly chosen from $\{1, \ldots, K\}$ with probability

$$P(M_i = k|\mathbf{y}, \mathbf{R}\backslash\{M_i\}) \quad \propto \quad w_k \cdot N_p(\mathbf{y}_i; \boldsymbol{\mu}_k + \mathbf{G}_k \mathbf{Z}_i^k, \boldsymbol{\Sigma}_{\epsilon k}).$$

In move type (b) (weight), the full conditional distribution for the weight w is still a Dirichlet distribution as

$$\mathbf{w}|\mathbf{y}, \mathbf{R}\backslash\{\mathbf{w}\} \sim Dir(\alpha_w + n_1, \ldots, \alpha_w + n_K)$$

In move type (c) (mean parameter), we have

$$\mu_{kj}|\mathbf{y}, \mathbf{R}\backslash\{\mu_{kj}\} \sim N(\theta_{kj}, \eta_{kj}^2),$$

where

$$\theta_{kj} = \eta_{kj}^2 \left\{ \frac{1}{\sigma_{\epsilon j}^2} \sum_{i \in N_k} \left(x_{ij} - \mathbf{g}_j^k \mathbf{Z}_i^{k*} \right) \right\}, \ \eta_{kj}^2 = \left(\frac{n_k}{\sigma_{\epsilon j}^2} + \frac{1}{\sigma_\mu^2} \right)^{-1}.$$

The hyper parameter σ_μ^2 in the prior of μ_{kj} is updated via

$$1/\sigma_\mu^2|\mathbf{y}, \mathbf{R} \sim Gamma\left(\frac{pK}{2} + a_\mu, \frac{\sum_{j=1}^p \sum_{k=1}^K \mu_{kj}^2}{2} + b_\mu \right).$$

In move type (d) (variance parameter),

$$1/\sigma_{\epsilon kj}^2|\mathbf{y}, \mathbf{R}\backslash\{\sigma_{\epsilon kj}^2\} \sim Gamma\left(\frac{n_k}{2} + a_\epsilon, \frac{1}{2} \sum_{i \in N_k} \left(x_{ij} - \mu_{kj} - \mathbf{G}_k \mathbf{Z}_i^k \right)^2 + b_\epsilon \right).$$

In move type (e)(latent variable), by the definition of MCRP

$$P(c_{kj} = l|\mathbf{c}_{-kj}) = \begin{cases} \frac{\alpha_0 + m_{0j}}{\alpha_0 + \alpha_c + K - 1} & \text{if } l = 0 \\ \frac{m_{lj}}{\alpha_0 + \alpha_c + K - 1} & \text{if } 0 < l \leq L_{j+} \\ \frac{\alpha_c}{\alpha_0 + \alpha_c + K - 1} & \text{otherwise,} \end{cases}$$

where m_{lj}, $l \geq 0$, is the number of clusters currently assigned to class l in the j-th factor loading and L_{j+} is the number of classes for which $m_{lj} > 0$, $l > 0$. The latent variable is updated via

$$P(c_{kj} = l|\mathbf{y}, \mathbf{R}\backslash\{c_{kj}\}) \propto P(c_{kj} = l|\mathbf{c}_{-kj}) \prod_{i \in N_k} N_p(\mathbf{y}_i; \boldsymbol{\mu}_k - G_k Z_i^k, \boldsymbol{\Sigma}_{\epsilon k})$$

In move type (f) (factor loading),

$$\mathbf{g}^{kj_1}|\mathbf{y},\mathbf{R}\backslash\{\mathbf{g}^{kj_1}\} \sim N_p(\mathbf{g}^{kj_1} : \mathbf{m}_{j_1}, \Sigma_{gj_1}),$$

where

$$\mathbf{\Sigma}_{gj_1} = \left\{ \sum_{i \in C_{kj_1}} Z_{ij_1}^{M_i 2} \Sigma_{\epsilon M_i j_1}^{-1} + I_p \right\}^{-1},$$

$$\mathbf{m}_{j_1} = \mathbf{\Sigma}_{gj_1} \left\{ \sum_{i \in C_{kj}} Z_{ij_1}^{M_i} \mathbf{a}_i \right\},$$

$\mathbf{a}_i = (a_{i1}, \ldots, a_{ip})'$ and $a_{ij} = \frac{(x_{ij} - \mu_{M_i,j} - \sum_{l \neq j_1} g_{jl}^k Z_{il}^k)}{\sigma_{\epsilon M_i j}^2}$.

In move type (g) (factor),

$$\mathbf{Z}_i^k|\mathbf{y},\mathbf{R}\backslash\{\mathbf{Z}_i^k\} \sim N_{q_k}(\mathbf{Z}_i^k : \mathbf{m}_i, \mathbf{\Sigma}_z),$$

where

$$\mathbf{\Sigma}_z = (\mathbf{G}_k^T \mathbf{\Sigma}_{\epsilon,k}^{-1} \mathbf{G}_k + \Psi^{-1})^{-1}$$

and

$$\mathbf{m}_i = \mathbf{\Sigma}_z \mathbf{G}_k^T \mathbf{\Sigma}_{\epsilon,k}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k).$$

The hyper parameter ψ_l in the prior of G_k is updated via

$$1/\psi_l|\mathbf{y}, bR \sim Gamma\left(\frac{n}{2} + a_g, \frac{\sum_{i=1}^n Z_{il}^{M_i 2}}{2} + b_g \right),$$

where $\lambda_l = 1/\psi_l$.

In move type (h)(Splitting or combining), we start with a random choice between attempting to split or combine with probabilities b_k and $d_k = 1 - b_k$. Let $d_1 = 0$ and $d_k = 0.5$ for $k < k_{max}$ and $d_{k_{max}} = 1$. For splitting one cluster into two, we choose k^*-th cluster at random from $\{1, \ldots, K\}$ and split into two, labeled as $k_1 = k^*$ and $k_2 = K + 1$. The weights w_{k_1} and w_{k_2} of the two new components are generated by

$$w_{k_1} = w_{k^*} u_1, \; w_{k_2} = w_{k^*}(1 - u_1),$$

where $u_1 \sim Beta(2, 2)$. Set the mean parameters of the two new clusters at

$$\mu_{k_1 j} = \mu_{k^* j} - u_{2j} \sigma_{\epsilon k^* j} \sqrt{\frac{w_{k_2}}{w_{k_1}}},$$
$$\mu_{k_2 j} = \mu_{k^* j} + u_{2j} \sigma_{\epsilon k^* j} \sqrt{\frac{w_{k_1}}{w_{k_2}}},$$

and variance parameters at

$$\sigma^2_{\epsilon k_1 j} = u_{3j}(1 - u^2_{2j}) \sigma^2_{\epsilon k^* j} \frac{w_{k^*}}{w_{k_1}},$$
$$\sigma^2_{\epsilon k_2 j} = (1 - u_{3j})(1 - u^2_{2j}) \sigma^2_{\epsilon k^* j} \frac{w_{k^*}}{w_{k_2}},$$

where $u_{2j} \sim SBeta(-1, 1)$ and $u_{3j} \sim Beta(1, 1)$ for $j = 1, \ldots, p.$. For $(\mathbf{c}_{k1}, \mathbf{c}_{k2})$, we propose

$$c_{k_1 j} = c_{k^* j}, \; c_{k_2 j} = u_{4j} \sim MCRP | c_{0j}, \ldots, c_{Kj}, c^{(1)}_{k^* j}, \ldots, c^{(r)}_{k^* j},$$

where $c^{(r)}_{k^* j}$ denotes r many copies of $c_{k^* j}$. This is the easiest proposal for the algorithm to be reversible. Finally, the acceptance probability for splitting

move is $\min(1, A)$

$$
\begin{aligned}
A &= \frac{P(\mathbf{y}|\mathbf{R}^{new})}{P(\mathbf{y}|\mathbf{R}^{old})} \frac{a_{nc}}{K+1} \frac{w_{k_1}^{\alpha_w+n_{k_1}-1} w_{k2}^{\alpha_w+n_{k2}-1}}{w_{k^*}^{n_{k^*}} B(\alpha, K\alpha)} \\
&\times \left(\frac{1}{2\pi\sigma_\mu^2}\right)^{p/2} \exp\left\{-\frac{1}{2\sigma_\mu^2}\sum_{j=1}^{p}\left(\mu_{k_1 j}^2 + \mu_{k_2 j}^2 - \mu_{k^* j}^2\right)\right) \\
&\times \left(\frac{b_\epsilon^{a_\epsilon}}{\Gamma(a_\epsilon)}\right)^p \prod_{j=1}^{p}\left(\frac{\sigma_{\epsilon k^* j}^2}{\sigma_{\epsilon k_1 j}^2 \sigma_{\epsilon k_2 j}^2}\right)^{a_\epsilon+1} \exp\left\{-b_\epsilon \sum_{j=1}^{p}(\frac{1}{\sigma_{\epsilon k_{1\mathrm{J}}}^2} + \frac{1}{\sigma_{\epsilon k_{2\mathrm{J}}}^2} - \frac{1}{\sigma_{\epsilon k^*_{\mathrm{J}}}^2})\right\} \\
&\times \frac{d_{K+1}}{b_K P_{alloc}} \left\{\phi_{2,2}(u_1)\prod_{j=1}^{p}\phi_{2,2}^s(u_{2j})\phi_{1,1}(u_{3j})\right\}^{-1} \\
&\times w_{k^*}\prod_{j=1}^{p}\frac{|\mu_{k_1 j} - \mu_{k_2 j}|\sigma_{\epsilon k_1 j}^2 \sigma_{\epsilon k_2 j}^2}{u_{2j}(1-u_{2j}^2)u_{3j}(1-u_{3j})\sigma_{\epsilon k^* j}^2}.
\end{aligned}
$$

If splitting is accepted, K is updated as $K+1$ and parameters are rearranged.

For combining two cluster into one, we randomly choose k_1 and k_2. Set $k^* = k_1$ with probability 1/2, otherwise $k^* = k_2$. All those observations $\mathbf{y}_i$ with $M_i = k_1$ or $M_i = k_2$ are reallocated in the new combined component labeled k^*. Set the weight

$$w_{k^*} = w_{k_1} + w_{k_2},$$

and set mean parameters $\mu_{\mu_k^*}$ to satisfy

$$w_{k^*}\mu_{k^*} = w_{k_1}\mu_{k_1} + w_{k_2}\mu_{k_2},$$

and set variance parameters $\sigma_{\epsilon k^*}^2$ to satisfy

$$w_{k^*}(\mu_{k^*}^2 + \sigma_{\epsilon k^*}^2) = w_{k_1^*}(\mu_{k_1^*}^2 + \sigma_{\epsilon k_1^*}^2) + w_{k_2^*}(\mu_{k_2^*}^2 + \sigma_{\epsilon k_2^*}^2).$$

The acceptance probability for combining move is $\min(1, A^{-1})$. If combining is accepted, K is updated as $K-1$ and parameters are rearranged.

This proposal for splitting and combining move is proposed by Richardson and Green [1997]. Dellaportas and Papageorgiou [2006] proposed to perturb μ_{k^*} according to the eigenvector directions of covariance matrix and eigenvector rotating the old eigenvector. In addition, their approach remains mathematical complexity of the Jacobian in high dimension. In our model, μ_{k^*} and $\sigma^2_{\epsilon k^* j}$ are independent with respect to the conditional posterior, and so we expect that coordinatewise perturbation works. This would be an advantage of the factor model.

In move type (i) (Birth or death), we first make a random choice between birth or death with probabilities b^*_k and $d^*_k = 1 - b^*_k$. Let $d^*_1 = 0$ and $b^*_k = d^*_k = 1/2$ for $k < k_{max}$, and $d^*_{k_{max}} = 1$. For a birth, parameters for a new component are drawn from

$$w^* \sim Beta(1, K), \mu_{(K+1)} \sim N_p(0, \sigma^2_\mu I_p), \sigma^2_{\epsilon(K+1)} \sim IG(a_\epsilon, b_\epsilon).$$

Set latent variable

$$c_{K+1,j} \sim MCRP | c_{0j}, \ldots, c_{Kj}, j = 1, \ldots, p.$$

Rescaling $w_k = w_k(1 - w^*)$ and let $w_{K+1} = w^*$.

The acceptance probability of birth is $\min(1, B)$

$$\begin{aligned} B &= \frac{a_{nc}}{K+1}(1-w^*)^n \frac{w^{*(\alpha_w-1)} \cdot (1-w^*)^{(K\alpha_w-1)}}{B(\alpha_w, K\alpha_w)} \\ &\times \frac{d_{K+1}}{b_K(k_0+1)}\phi_{1,k}(w^*)^{-1}, \end{aligned}$$

where k_0 is the number of empty cluster.

For a death, we randomly choose a cluster k^* at random among k with $n_k = 0$. The chosen component is deleted and remaining weights are rescaled to sum to 1. The acceptance probability of death is $\min(1, B^{-1})$.

Chapter 4

Numerical studies

In this chater, we investigate the performance of the proposed Bayesian factor clustering(BFC) method. In particular, we compare the proposed BFC with other methods, EM approach(EM) and common factor analyzer(CFA) in terms of clustering performance. Here, EM approach is estimated by the 'mclust' algorithm available in R. CFA is estimated by the 'mcfa' package by Geoff McLachla(http://www.maths.uq.edu.au/ gjm/). We analyzed various types synthetic and two data sets.

4.1 Simulation studies

In this section, we considered six cases of simulations. For each experimental setting, we replicated the simulation 10 times.

4.1.1 Simulation 1(Covariance matrix structures)

In this subsection, we have investigated the model performances in terms of four covariance matrix structures which are diagonal, unspecified, (diagonal or unspecified) and (diagonal or factor), respectively. Here, (diagonal or unspecified) means that component covariance matrix is a diagonal or unspecified. Similarly, (diagonal or factor) means that component covariance matrix is a diagonal or factor model. Therefore, covariance matrix structures in the last two scenarios are different for each cluster. For this, we fixed the number of clusters, the number of observations and mean vectors in each cluster. Consider a 5-cluster example, with the observation vector in a 3-dimensional space. We generate the same number, 50, of observations in each cluster. For each scenario, mean vectors are

$$\boldsymbol{\mu}_1 = (0,0,0)^T, \boldsymbol{\mu}_2 = (-3,-3,-3)^T,$$

$$\boldsymbol{\mu}_3 = (3,3,3)^T, \boldsymbol{\mu}_4 = (-6,-6,-6)^T, \boldsymbol{\mu}_5 = (6,6,6)^T,$$

and covariance matrixes are as follows

- Diagonal covariance matrix(Diag) scenario

$$\Sigma_1 = \text{diag}(1,1,1), \Sigma_2 = \text{diag}(2,2,2),$$

$$\Sigma_3 = \text{diag}(1,1,1), \Sigma_4 = \text{diag}(1,1,1), \Sigma_5 = \text{diag}(2,2,2)$$

- Unspecified covariance matrix(Unsp) scenario

$$\Sigma_1 = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 4 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 2 & -1.5 & 1 \\ -1.5 & 5 & 2 \\ 1 & 2 & 3 \end{pmatrix},$$

$$\Sigma_3 = \begin{pmatrix} 5 & -1 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 3 \end{pmatrix}, \Sigma_4 = \begin{pmatrix} 4 & 1.5 & 0 \\ 1.5 & 3 & 2 \\ 0 & 2 & 2 \end{pmatrix}, \Sigma_5 = \begin{pmatrix} 1.5 & -2 & 1.5 \\ -2 & 3 & -2.5 \\ 1.5 & 2.5 & 4 \end{pmatrix}$$

- Diagonal or unspecified covariance matrix(DiUn) scenario

$$\Sigma_1 = \text{diag}(1,1,1), \Sigma_3 = \text{diag}(2,2,2), \Sigma_4 = \text{diag}(2,2,2)$$

$$\Sigma_2 = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 4 \end{pmatrix}, \Sigma_5 = \begin{pmatrix} 2 & -1.5 & 1 \\ -1.5 & 5 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

- Diagonal or factor covariance matrix(DiFa) scenario

$$\Sigma_1 = \mathbf{g}_1\mathbf{g}_1^T + \text{diag}(1,1,1), \Sigma_2 = \mathbf{g}_2\mathbf{g}_2^T + \text{diag}(1,1,1)$$

$$\Sigma_3 = \text{diag}(2,2,2), \Sigma_4 = \mathbf{g}_2\mathbf{g}_2^T + \text{diag}(1,1,1), \Sigma_5 = \text{diag}(2,2,2),$$

where $\mathbf{g}_1 = (1, 0.9, 0.8)^T, \mathbf{g}_2 = (1.2, -0.7, 0.4)^T$.

We choose the hyper parameters, $k_{max} = 20$, $a_\mu = b_\mu = 1$, $a_g = b_g = 1$, $a_w = 1$ and the hyper parameters of MCRP, $a_0 = 0.5$ and $a_c = 5.0$. The choice of hyper parameters of MCRP may be to cover covariance structure in between the diagonal and unspecified covariance. We ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

Table 4.1 summaries means of Error Rates(ER) for number(K) and means of Error Rates(ER), Rand index(RI)(Rand [1971]) , Adjusted Rand Index(ARI)(Hubert and Arabie [1985]), Jaccard Index(JI)(Jaccard [1903]) and Fowlkes and Mallows index(FMI)(Fowlkes and Mallows [1983]). Here, we use unspecified covariance matrix for EM method and bold font represents best accuracy between the 3 models. We see that our method have a good performance than other methods without Unsp scenario. Since the estimated covariance matrix via EM and CFA methods is overfitting in Diag scenario, their have a poor performance. In Unsp scenario, it is not surprising that the EM method is better in performance accuracy than BFC, because true covariance structure is unspecified. Since proposed method consider various component covariance matrix unlike EM and CFA, our model is well suited for the last two scenario.

Table 4.1: Result of simulation 1

Scenario	Diag			Unsp		
Method	EM	CFA	BFC	EM	CFA	BFC
ER of K	1.000	0.900	**0.000**	0.200	1.000	**0.100**
ER	0.402	0.505	**0.108**	**0.127**	0.443	0.206
RI	0.812	0.727	**0.923**	**0.921**	0.765	0.874
ARI	0.556	0.447	**0.756**	**0.749**	0.450	0.603
JI	0.507	0.442	**0.674**	**0.666**	0.422	0.519
FMI	0.699	0.652	**0.805**	**0.799**	0.621	0.682
Scenario	DiUn			DiFa		
Method	EM	CFA	BFC	EM	CFA	BFC
ER of K	0.900	1.000	**0.000**	0.800	0.300	**0.000**
ER	0.338	0.444	**0.132**	0.363	0.166	**0.085**
RI	0.835	0.755	**0.908**	0.803	0.905	**0.939**
ARI	0.570	0.465	**0.713**	0.555	0.758	**0.808**
JI	0.509	0.442	**0.628**	0.522	0.706	**0.735**
FMI	0.690	0.642	**0.770**	0.699	0.824	**0.846**

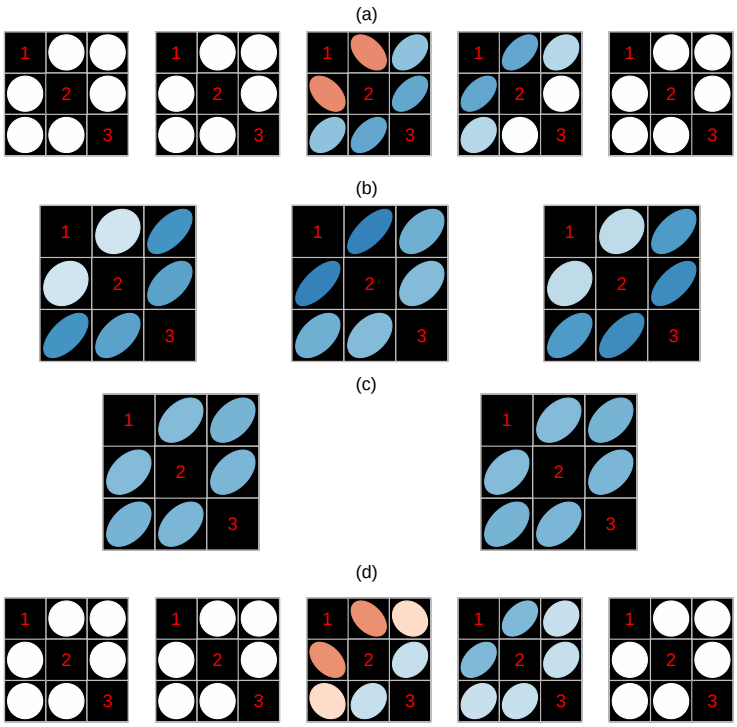

Figure 4.1: Estimated correlation matrix plot in Unsp scenario

Figure 4.1 shows true correlation(a), estimated correlation using EM(b), CFA(c) and BFC(d). For details of the correlation plot see the 'corrplot' package in R. In particular, proposed method is well estimated to correlation matrix with different component covariance structure.

In order to investigate feature of our method, we consider DiUn scenario in other simulations.

4.1.2 Simulation 2(The number of cluster)

In this simulation example, we set to almost DiUn scenario setting in section 4.1.1. But, the number of components are different. For this, we fixed the number of observations in each cluster and the dimension of observation vector. The observation vector in a 3-dimensional space is generate the same number, 50, of observations in each cluster. We consider 3-cases of the number of component such that $K = 3, 5, 10$. Here, setting of scenario with $K = 5$ corresponds setting of DiUn scenario in section 4.1.1, and setting of scenario with $K = 3$ is equal to elimination the last two components of DiUn scenario in section 4.1.1. For scenario with $K = 10$, mean vectors are

$$\boldsymbol{\mu}_1 = (0,0,0)^T, \boldsymbol{\mu}_2 = (-3,-3,-3)^T, \boldsymbol{\mu}_3 = (3,3,3)^T$$

$$\boldsymbol{\mu}_4 = (-6,-6,-6)^T, \boldsymbol{\mu}_5 = (6,6,6)^T, \boldsymbol{\mu}_6 = (-9,-9,-9)^T,$$

$$\boldsymbol{\mu}_7 = (9,9,9)^T, \boldsymbol{\mu}_8 = (-12,-12,-12)^T, \boldsymbol{\mu}_9 = (12,12,12)^T, \boldsymbol{\mu}_{10} = (15,15,15)^T,$$

covariance matrices are as follows

$$\Sigma_1 = \text{diag}(1,1,1),\ \Sigma_3 = \text{diag}(2,2,2),\ \Sigma_4 = \text{diag}(2,2,2)$$

$$\Sigma_6 = \text{diag}(2,2,2),\ \Sigma_9 = \text{diag}(2,2,2),\ \Sigma_{10} = \text{diag}(1,1,1)$$

$$\Sigma_2 = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 4 \end{pmatrix},\ \Sigma_5 = \begin{pmatrix} 2 & -1.5 & 1 \\ -1.5 & 5 & 2 \\ 1 & 2 & 3 \end{pmatrix},$$

Table 4.2: Result of simulation 2

Scenario	Method	ER of K	ER	RI	ARI	JI	FMI
$K = 3$	EM	0.300	0.181	0.820	0.631	0.642	0.777
	CFA	0.200	0.155	0.839	0.700	0.717	0.832
	BFC	**0.100**	**0.075**	**0.908**	**0.792**	**0.758**	**0.861**
$K = 5$	EM	0.900	0.338	0.835	0.570	0.509	0.690
	CFA	1.000	0.444	0.755	0.465	0.442	0.642
	BFC	**0.000**	**0.132**	**0.908**	**0.713**	**0.628**	**0.770**
$K = 10$	EM	1.000	0.559	0.816	0.397	0.321	0.532
	CFA	1.000	0.700	0.741	0.302	0.256	0.489
	BFC	**0.300**	**0.173**	**0.945**	**0.696**	**0.571**	**0.727**

$$\Sigma_7 = \begin{pmatrix} 5 & -1 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 3 \end{pmatrix}, \ \Sigma_8 = \begin{pmatrix} 4 & 1.5 & 0 \\ 1.5 & 3 & 2 \\ 0 & 2 & 2 \end{pmatrix}$$

We choose the hyper parameters similar to those of section 4.4.1, and we ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

The results are summarized in table 4.2. We find that the performance of proposed method is better than other methods. EM and CFA methods have

a poor performance as increasing the number of components. In particular, their may trend to form one from multiple components. However, our method is a robust for increasing the number of components than other approaches.

4.1.3 Simulation 3(The dimension)

In this subsection, we check the effect of the dimension of observation. Hence, we fixed the number of clusters and the number of observations in each cluster. We consider 3-cases of the dimension of observation such that $p = 3, 5, 10$. Here, setting of scenario with $p = 3$ corresponds setting of DiUn scenario in section 4.1.1. Mean vectors of scenarios with $p = 5$ and $p = 10$ are

$$\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kp})^T$$

where

$$\mu_{1j} = 0,\ \mu_{2j} = -3,\ \mu_{3j} = 3,\ \mu_{4j} = -6,\ \mu_{5j} = 6,$$

$j = 1, \ldots, p$. Covariance matrices are generated using inverse wishart distribution.

We choose the hyper parameters similar to those of section 4.4.1, and we ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

The clustering results are detailed in Table 4.3. We see that proposed method performed better than other methods.

Table 4.3: Result of simulation 3

Scenario	Method	ER of K	ER	RI	ARI	JI	FMI
$p = 3$	EM	0.900	0.338	0.835	0.570	0.509	0.690
	CFA	1.000	0.444	0.755	0.465	0.442	0.642
	BFC	**0.000**	**0.132**	**0.908**	**0.713**	**0.628**	**0.770**
$p = 5$	EM	0.400	0.181	0.906	0.777	0.743	0.846
	CFA	0.900	0.200	0.903	0.765	0.725	0.837
	BFC	**0.000**	**0.024**	**0.982**	**0.942**	**0.912**	**0.954**
$p = 10$	EM	0.800	0.481	0.700	0.404	0.443	0.616
	CFA	0.600	0.129	0.947	0.850	0.800	0.890
	BFC	**0.400**	**0.062**	**0.975**	**0.930**	**0.907**	**0.950**

4.1.4 Simulation 4(The number of observations)

In this simulation example, we set to almost the DiUn scenario setting in section 4.1.1. But, the number of observations are different. Hence, we fixed the number of clusters, and mean vectors and covariance matrices in each cluster. We consider 3-cases of the number of observation such that $n_k = 25, 50, 100$. Here, setting of scenario with $n_k = 50$ corresponds setting of DiUn scenario in section 4.1.1.

We choose the hyper parameters similar to those of section 4.4.1, and we ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

The results are summarized in table 4.4. We find that the performance of proposed method is better than other methods. All methods have a good performance as increasing the number of observations. However, proposed method have a good performance with small sample size than other methods.

4.1.5 Simulation 5(The unbalanced data)

In this subsection, we set to almost the DiUn scenario setting in section 4.1.1. But, the number of observations in each cluster are different. We consider the number of observation in each cluster such that

$$n_1 = n_2 = n_5 = 20, \ n_3 = n_4 = 100$$

Table 4.4: Result of simulation 4

Scenario	Method	ER of K	ER	RI	ARI	JI	FMI
$n_k = 25$	EM	1.000	0.586	0.638	0.300	0.333	0.556
	CFA	1.000	0.541	0.659	0.332	0.355	0.574
	BFC	**0.800**	**0.290**	**0.848**	**0.586**	**0.519**	**0.693**
$n_k = 50$	EM	0.900	0.338	0.835	0.570	0.509	0.690
	CFA	1.000	0.444	0.755	0.465	0.442	0.642
	BFC	**0.000**	**0.132**	**0.908**	**0.713**	**0.628**	**0.770**
$n_k = 100$	EM	0.300	0.190	0.885	0.667	0.590	0.743
	CFA	0.100	0.134	0.915	0.747	**0.671**	**0.803**
	BFC	**0.000**	**0.114**	**0.920**	**0.750**	0.668	0.801

Table 4.5: Result of simulation 5

Method	EM	CFA	BFC
ER of K	1.000	1.000	**0.500**
ER	0.291	0.333	**0.260**
RI	0.747	0.736	**0.801**
ARI	0.463	0.470	**0.522**
JI	0.488	**0.507**	0.497
FMI	0.659	**0.680**	0.663

.

We choose the hyper parameters similar to those of section 4.4.1, and we ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

Table 4.5 present the results of simulation 5. The proposed method performed better than other methods. Therefore, when we have different cluster size, the our method could provide accurate clustering results than other methods.

4.1.6 Simulation 6(The hyper parameter of MCRP)

In this simulation example, we check that the range of our model is very large so that it covers from the simplest model(the equal diagonal covariance matrices) to the most complex one(unspecified covariance matrices). Hence, we consider 3-cases of the hyper parameters for MCRP of DiUn scenario in section 4.1.1. For this, we chosen that the hyper parameters for simplest(Simp) and the most complex(MoCo) model are $a_0 = 10000$ and $a_c = 0.0001$ and $a_0 = 0.0001$ and $a_c = 10000$, respectively. Here, the last scenario corresponds setting of DiUn scenario in section 4.1.1.

We choose the hyper parameters similar to those of section 4.4.1 without the hyper parameter for MCRP, and we ran RJMCMC 70,000 iterations and 20,000 iterations as a burn-in.

The results are summarized in Table 4.6, and Figure 4.2 depicts true correlation(a), estimated correlation using Simp(b), DiUn(c) and MoCo(d). Proposed method via MoCo have a poor performance and incorrect correlation matrix. In particular, our method via Simp have a similar performance to DiUn, but incorrect correlation matrix. Therefore, by choosing the prior parameters accordingly, we can find a best parsimonious model in between the simplest and most complex finite mixture models.

Table 4.6: Result of simulation 6

Method	Simp	DiUn	MoCo
ER of K	0.000	**0.000**	1.000
ER	0.143	**0.132**	0.430
RI	0.902	**0.908**	0.766
ARI	0.695	**0.713**	0.426
JI	0.609	**0.628**	0.400
FMI	0.757	**0.770**	0.593

4.2 Real data analysis

We summarize the real data in table 4.7. We analyzed two data sets, Iris and Wine. The results are obtained by 10 random partitions of the data set into 70% training data and 30% test data. We choose the hyper parameters similar to those of section 4.4.1, and we ran RJMCMC 70,000 iterations and 20,000

Table 4.7: The information of two real data

Data set	Cluster	Cluster size	Total size	p
Iris	3	50-50-50	150	4
Wine	3	59-71-48	178	13

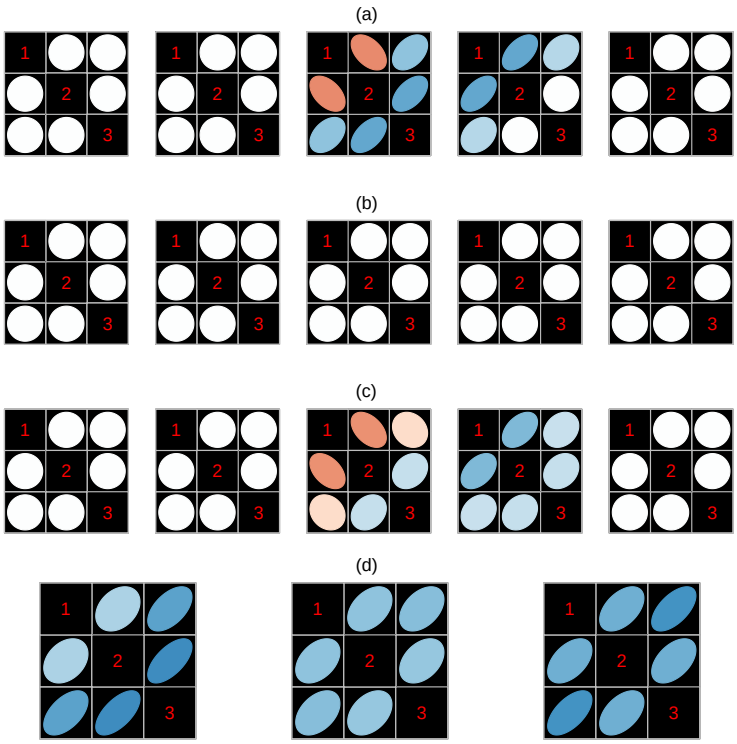

Figure 4.2: Estimated correlation matrix plot in simulation 6

iterations as a burn-in.

Table 4.8 present the results of Iris data. We see that our method have a good performance than other methods. ER of our method was 0.218, RI was 0.820 and ARI was 0.634.

The clustering results of Wine data are detailed in Table 4.9. The proposed method performed better than other methods. ER of our method was 0.081, RI was 0.913 and ARI was 0.804.

Table 4.8: Result of Iris data

Method	EM	CFA	BFC
ER of K	0.900	0.900	**0.000**
ER	0.280	0.267	**0.218**
RI	0.785	0.801	**0.820**
ARI	0.580	0.615	**0.634**
JI	0.599	0.637	**0.638**
FMI	0.771	**0.795**	0.788

Table 4.9: Result of Wine data

Method	EM	CFA	BFC
ER of K	1.000	0.300	**0.200**
ER	0.319	0.141	**0.081**
RI	0.833	0.857	**0.913**
ARI	0.581	0.707	**0.804**
JI	0.527	0.712	**0.770**
FMI	0.708	0.827	**0.869**

From the experimental results, we can conclude that proposed method provides an improvement in terms of clustering performances. In particular, proposed method outperforms with different component covariance structure.

Chapter 5

Concluding remarks

In this thesis, for clustering problem, we suggested Bayesian multivariate mixture model with unknown number of components. The proposed method has several advantages over existing model-based clustering. Advantages of our model are (1) the number of factors in each component is estimated automatically, and (2) clusters can share factors so we can reduce the number of parameters further and (3) the range of the model is very large so that it covers from the simplest model (the equal diagonal covariance matrices) to the most complex one (unspecified covariance matrices). By choosing the prior parameters accordingly, we can find a best parsimonious model in between the simplest and most complex finite mixture models. From numerical studies, we confirmed that our method well perform with different covariance structure

in clusters. In the near future, we will extend binary data using auxiliary variable.

Bibliography

M. Aitkin. Likelihood and bayesian analysis of mixtures. *Statistical Modelling*, 1(4):287–304, 2001.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, volume 1, pages 267–281. Springer Verlag, 1973.

J. Baek, G.J. McLachlan, and L.K. Flack. Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1298–1309, 2010.

J.D. Banfield and A.E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique oc-

curring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

H.H. Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

S.P. Brooks, P. Giudici, and G.O. Roberts. Efficient construction of reversible jump markov chain monte carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–39, 2003.

O. Cappé, C.P. Robert, and T. Rydén. Reversible jump, birth-and-death and more general continuous time markov chain monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3): 679–700, 2003.

G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

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

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from in-

complete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

J. Diebolt and C.P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.

L. Engelman and J.A. Hartigan. Percentage points of a test for clusters. *Journal of the American Statistical Association*, 64(328):1647–1648, 1969.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, pages 577–588, 1995.

Y. Fan and SP Brooks. Bayesian modelling of prehistoric corbelled domes. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3): 339–354, 2000.

E.B. Fowlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, pages 553–569, 1983.

C. Fraley and A.E. Raftery. Model-based clustering, discriminant analysis,

and density estimation. *Journal of the american statistical association*, 97 (458):611–631, 2002.

C. Fraley and A.E. Raftery. Mclust version 3 for r: Normal mixture modeling and model-based clustering. Technical report, Technical report, 2006.

Z. Ghahramani and G.E. Hinton. The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

J. Ghosh and D.B. Dunson. Default prior distributions and efficient posterior computation in bayesian factor analysis. *Journal of Computational and Graphical Statistics*, 18(2):306–320, 2009.

P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2 (1):193–218, 1985.

P. Jaccard. Distribution comparée de la flore alpine dans quelques régions des alpes occidentales et orientales. *year [ca. 1903]*, 1903.

S. Kim, M.G. Tadesse, and M. Vannucci. Variable selection in clustering via dirichlet process mixture models. *Biometrika*, 93(4):877–893, 2006.

G. McLachlan and D. Peel. Mixtures of factor analyzers. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

S. Newcomb. A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, 8(4):343–366, 1886.

D.J. Nott and D. Leonte. Sampling schemes for bayesian variable selection in generalized linear models. *Journal of Computational and Graphical Statistics*, 13(2):362–382, 2004.

W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8:1145–1164, 2007.

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

W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, pages 846–850, 1971.

S. Richardson and P.J. Green. On bayesian analysis of mixtures with an

unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6 (2):461–464, 1978.

M. Smith and R. Kohn. Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75(2):317–343, 1996.

D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

M. Stephens. Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of Statistics*, pages 40–74, 2000.

M.G. Tadesse, N. Sha, and M. Vannucci. Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, 100(470):602–617, 2005.

D.M. Titterington, A.F.M. Smith, U.E. Makov, et al. *Statistical analysis of finite mixture distributions*, volume 38. Wiley New York, 1985.

J. Vermaak, C. Andrieu, A. Doucet, and SJ Godsill. Reversible jump markov chain monte carlo strategies for bayesian model selection in autoregressive processes. *Journal of time series analysis*, 25(6):785–809, 2004.

B. Xie, W. Pan, and X. Shen. Variable selection in penalized model-based clustering via regularization on grouped parameters. *Biometrics*, 64(3):921–930, 2008.

K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

Z. Zhang, K.L. Chan, Y. Wu, and C. Chen. Learning a multivariate gaussian mixture model with the reversible jump mcmc algorithm. *Statistics and Computing*, 14(4):343–355, 2004.

국문초록

다변량자료에서 혼합모형을 고려할 때 공분산 행렬의 추정은 많은 어려움을 가지고 있다. 특히 자료의 차원이 커지면 공분산 행렬에서 추정해야 할 모수의 수가 자료의 차원보다 제곱으로 커지게 된다. 또한 베이지안 분석의 MCMC 알고리즘은 공분산 행렬의 추정에 많은 시간과 노력들이 필요하다.

본 학위논문에서는 인자모형을 이용하여 공분산 행렬의 모수들을 줄이는 방법을 연구하였다. 각 군집의 공분산 행렬 구조는 인자들의 선형결합으로 재표현 할 수 있다는 가정을 통해 우리는 가장 간단한 공분산 구조(대각행렬)부터 복잡한 공분산 구조들을 표현 할 수 있다. 또한 특정 인자들은 각 군집에 공유될 수 있게 공분산 행렬의 잠재변수에 Modified Chinese restaurant process를 적용 하였다. 마지막으로 모의실험과 실제자료를 통해 제안된 방법이 기존의 혼합모형보다 성능이 더 우수하다는 것을 보였다.

주요어 : 베이지안 분석, 혼합모형, 인자모형, 군집분석, RJMCMC LASSO

학 번 : 2005 – 20294

감사의 글

학업에 뜻을 품고 대학원을 진학한 지 7년이라는 시간이 지났습니다. 학위를 받기까지 얼마나 많은 분들의 도움을 받았는지 모르겠습니다. 소중한 분들의 도움이 없었다면 제가 이 논문을 완성할수 없었을 것입니다. 변변치 않은 감사의 글로나마 고마운 마음을 올립니다.

먼저 많이 부족한 저에게 훌륭한 가르침을 주셨던 통계학과 교수님들에게 진정 감사드립니다. 특히 부모님처럼 보살펴 주시고 인생과 학문을 가르쳐 주신 전종우 교수님과 제가 학문적으로 더 성숙할 수 있도록 따뜻한 조언과 세심한 지도로 단련해 주신 김용대 교수님께 정말 감사드립니다. 또한 귀중한 시간을 내시어 저의 부족한 논문을 심사해 주신 박병욱 교수님, 임요한 교수님, 이태영 박사님께도 감사드립니다.

대학원 생활동안 연구실에서 함께 생활하며 공부했던 선후배님들께도 감사를 드립니다. 큰 형님이신 광수형님, 동화형, 호식이형, 인재형, 성훈이형, 도현이형, 상준이형, 범수형, 길영이를 비롯한 여러 연구실 선배들과 동기인 재연이와 연하 그리고 같이 졸업하는 종준이형, 미애, 상미, 주유, 민우, 승환, 우성, 효원, 원준, 재성, 지선이를 비롯한 많은 후배들에게도 고맙다고 말하고 싶습니다. 그리고 오랜 시간동안 함께 고민도 하고 서로 의지하며 공부했던 병엽, 상인에게 고맙다는 마음을 전합니다.

마지막으로 저를 위해 항상 애쓰시고 사랑해 주시는 부모님께 감사의 마음을 드립니다. 앞으로도 열심히 생활하여 자랑스런 아들이 되도록 노력하겠습니다. 또한 항상 든든한 후원자가 되어준 누나들과 자형들에게 감사의 마음을 드리며, 사랑스러운 조카들에게 고마움을 전합니다.

2012년 8월 김 재 석