# New Advancements of Scalable Statistical Methods for Learning Latent Structures in Big Data

by

Shiwen Zhao

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: _____

Approved:

_____
Sayan Mukherjee, Supervisor

_____
David B Dunson

_____
Barbara E Engelhardt

_____
Alexander J Hartemink

_____
Tim E Reddy

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program in Computational Biology and Bioinformatics
in the Graduate School of Duke University
2016

# Abstract

## New Advancements of Scalable Statistical Methods for Learning Latent Structures in Big Data

by

Shiwen Zhao

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: _____

Approved:

_____
Sayan Mukherjee, Supervisor

_____
David B Dunson

_____
Barbara E Engelhardt

_____
Alexander J Hartemink

_____
Tim E Reddy

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program in Computational Biology and Bioinformatics
in the Graduate School of Duke University
2016

# Abstract

Constant technology advances have caused data explosion in recent years. Accordingly modern statistical and machine learning methods must be adapted to deal with complex and heterogeneous data types. This phenomenon is particularly true for analyzing biological data. For example DNA sequence data can be viewed as categorical variables with each nucleotide taking four different categories. The gene expression data, depending on the quantitative technology, could be continuous numbers or counts. With the advancement of high-throughput technology, the abundance of such data becomes unprecedentedly rich. Therefore efficient statistical approaches are crucial in this big data era.

Previous statistical methods for big data often aim to find low dimensional structures in the observed data. For example in a factor analysis model a latent Gaussian distributed multivariate vector is assumed. With this assumption a factor model produces a low rank estimation of the covariance of the observed variables. Another example is the latent Dirichlet allocation model for documents. The mixture proportions of topics, represented by a Dirichlet distributed variable, is assumed. This dissertation proposes several novel extensions to the previous statistical methods that are developed to address challenges in big data. Those novel methods are applied in multiple real world applications including construction of condition specific gene co-expression networks, estimating shared topics among newsgroups, analysis of promoter sequences, analysis of political-economics risk data and estimating population

structure from genotype data.

To my family

# Contents

# List of Tables

# List of Figures

# List of Abbreviations and Symbols

## Symbols

| | |
|---|---|
| $\boldsymbol{X}$ | A matrix or a multi-way tensor |
| $\boldsymbol{x}$ | A column vector |
| $x$ | A scalar |
| $\mathbb{E}$ | Expectation operator |

## Abbreviations

| | |
|---|---|
| BCCA | Bayesian canonical correlation analysis |
| BMF | Bingham-von Mises-Fisher |
| CCA | Canonical correlation analysis |
| EM | Expectation maximization |
| FA | Factor analysis |
| GFA | Group factor analysis |
| GMM | Generalized method of moments |
| HMC | Hamiltonian Monte Carlo |
| MAP | Maximum a posteriori |
| MCMC | Markov chain Monte Carlo |
| PX | Parameter expansion |
| SVD | Singular value decomposition |

# Acknowledgements

I would like to express my sincere gratitude to my three main advisors on my committee: Prof. Barbara Engelhardt, Prof. Sayan Mukherjee and Prof. David Dunson. My thesis could not be finished without their support and encouragement.

I am deeply grateful to Prof. Barbara Engelhardt. Her enthusiasm in science and energetic attitude has been inspiring me since the first time I met her. I am very thankful for that she kept funding my research for a year even she was at Princeton. Her continual support helped me overcome many difficulties and finish my Ph.D. dissertation.

My deepest gratitude is to Prof. Sayan Mukherjee. He is a person with countless new ideas and at the same time has a profound understanding of the problem. He gave me many invaluable guidance of doing scientific research. In addition, his patience gave me tremendous courage to overcome my communication disadvantages. I could not survive without his support during my graduate life.

I am also sincerely grateful to Prof. David Dunson. David is a person who serves as a role model during my graduate life. Anyone would be impressed by his broad knowledge and profound vision. His advice in scientific problems stimulates my thinking to a deeper level. It has been my greatest privilege to have David on my committee and work with him and learn from him.

This dissertation could not be finished without the support from the members on my committee including Prof. Alex Hartemink, Prof. Tim Reddy. I thank them

for their comments and advice in general. I am particularly grateful to Prof. Scott Schmidler, who has advised my rotation projects once I arrived at Duke. Also great thanks to Prof. Terry Oas, who gave me an opportunity to experience experimental techniques in molecular biology and Prof. Jeff Thorne, who allowed me to work with him on a rotation project.

I would like to thank faculty members in statistics department and computational biology program for teaching me different courses. Thanks to Prof. Robert Wolpert, Prof. Merlise Clyde, Prof. Li Ma, Prof. Surya Tokdar, Prof. Fred Dietrich. I would like to thank my friends and colleagues. Thanks to Dr. Chuan Gao and Dr. Sanvesh Srivastava for numerous discussions. Thanks to Dr. Tingran Gao and Rujie Yin for being helpful when I have questions in mathematical derivations. Thanks to Amanda Lea for discussion of biology problems. Thanks to Dr. Douglas VanDerwerken, Dr. Ashlee Benjamin, Vivian Zhang, Shaobo Han, Yezhou Huang, Jincheng Li, Yang Qi, Hui Kang, Weiwei Li for being my friends and making my graduate life colorful.

I want to save my greatest gratitude to my family. I am especially grateful to my wife and my best friend Ruo He. Her support has been making my life much easier. I also want to thank my parents for their unconditional love. This dissertation could not be finished without their support.

# 1

# Introduction

Constant technology advances have induced data explosion in recent years. This phenomenon is particularly pronounced in biology. Such a big data era comes with the strong needs of efficient and scalable statistical approaches to extract useful information from massive data sets. This dissertation addresses several important practical problems by developing theoretically supported and computationally efficient models. The practical potential of the developed models are demonstrated by real world applications.

Chapter 2 will develop a new Bayesian factor model for multiple coupled observations. This model is named BASS. BASS is motivated by the fact that in real world applications, people often encounter paired or multiple coupled observations. For example expression profiles of a number of genes for a particular individual are measured under different conditions. Researchers are particularly interested in the covariance specific to each observation as well as the covariance among different combinations of observations. Built on the latest innovations in Bayesian shrinkage priors, a structured prior that combines element-wise sparsity with column-wise sparsity on the factor loading matrix is developed. In addition the prior allows

mixture of sparse and dense columns in the loading, generating sparse + low rank decompositions of the covariance matrix. To efficiently perform maximum a posteriori (MAP) parameter estimation, a parameter expanded expectation maximization (PX-EM) algorithm is proposed. The PX-EM algorithm introduces an additional rotation parameter into the factor model. The additional rotation parameter connects posterior modes in the original space by equal likelihood curves in the expanded space, therefore it facilitates efficient posterior mode search. The performance of BASS is evaluated by comparisons with other existing methods through simulation studies. Results suggest BASS achieves best parameter estimation and prediction accuracy in most cases. In the end BASS is applied to real data sets with the aim of multivariate response prediction, constructing condition specific gene co-expression networks and inferring topics that are shared by different newsgroups.

Chapter 3 will propose a new model, called generalized latent Dirichlet variable model. Such a model assumes each multivariate observation partially belongs to $k$ latent components. Moreover different coordinates of the observation could take different distributions. Previous parameter estimation methods for such a model have relied on EM or Markov chain Monte Carlo (MCMC) algorithms with initiations of latent variables. To perform parameter estimation efficiently a generalized methods of moment (GMM) approach is developed to estimate component parameters of the model. The new approach does not require initiations of latent variables. This is achieved by constructing moment functions from second and third order cross moments among variables. The moment functions have expectation of zero at true values of parameters. By minimizing quadratic forms of the moment functions parameters could be estimated using a coordinate descent algorithm. Using GMM theories the asymptotic properties and efficiency of the estimator are shown. We name the new approach MELD. MELD is orders of magnitude faster than alternative estimation methods such as EM or MCMC algorithms, and at the same time it

achieves higher estimation accuracy. The performance of MELD is evaluated by comparisons with competing methods through simulation studies. To demonstrate the utility of MELD in real world applications we apply it to public available data sets including a promoter sequence data, a political-economic risk data and a genotype + gene expression data in human HapMap phase 3 data set.

Chapter 4 will develop a sampling method for distributions on Riemannian manifolds. One example of such distributions is the Bingham-von Mises-Fisher (BMF) distribution. The distribution is defined on the Stiefel manifold consisting of $p \times k$ orthonormal matrices. This distribution has been frequently encountered in problems such as orthogonal factor analysis and probabilistic singular value decomposition (SVD). Motivated by those problems, an efficient Monte Carlo method that could draw samples from those distributions is developed. The method combines the Hamiltonian Monte Carlo (HMC) algorithm with a geodesic integrator. The utility of the new method is demonstrated in two applications.

Chapter 5 will give a concluding remark.

# 2

# BASS - a scalable Bayesian group factor analysis model for structured latent space learning

Linear dimension reduction techniques play a very important role in modern data analysis. The basic idea of linear dimension reduction is to find a lower dimensional latent space that useful information in the original space can be kept. Examples of such techniques include principal component analysis (PCA) (Hotelling, 1933), factor analysis (FA) (Spearman, 1904) and canonical correlation analysis (CCA) (Hotelling, 1936). A comprehensive and elegant review can be found in (Cunningham and Ghahramani, 2014). In this chapter we investigate a new FA model called group factor analysis (GFA) model which can combinatorially model multiple data sets and can learn a latent space that is structured in a desirable way. This is achieved from the latest developments in Bayesian literature using sparsity inducing priors. The rest of this chapter is organized as follows. We introduce GFA in Section 2.1. In Section 2.2 we review recent strategies, mainly from Bayesian point of view, to structure the latent space. We also give a brief review about recent innovations in Bayesian shrinkage priors. In Section 2.3 we combine GFA with a particular Bayesian

shrinkage prior and develop a new Bayesian GFA model. The new model is called BASS standing for Bayesian group Analysis with Structured Sparsity. In Section 2.4 we propose a fast and accurate parameter estimation method with parameter expansion. Simulations and applications are demonstrated in Section 2.5 and 2.6 respectively. We conclude this chapter by a discussion in Section 2.7.

## 2.1 Group factor analysis

### *2.1.1 Factor analysis*

Before we come to the group factor analysis model, we first introduce factor analysis. A FA model finds a low dimensional latent variable $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$ from a high dimensional observation $\boldsymbol{y}_i \in \mathbb{R}^{p\times 1}$ for subject $i$ with $i = 1, \ldots, n$. Usually it is assumed the dimension of $\boldsymbol{y}_i$ is larger than the dimension of $\boldsymbol{x}_i$. A sample in the low dimensional space is linearly projected to the original high dimensional space through a *loading matrix* $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times k}$. The observation $\boldsymbol{y}_i$ is assumed to be a noisy version of the projection, with the noise denoted as $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p\times 1}$. Formally the factor model could be written as

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \tag{2.1}$$

for $i = 1, \ldots, n$. In a standard FA model, $\boldsymbol{x}_i$ is assumed to follow a $\mathrm{N}_k(\boldsymbol{0}, \boldsymbol{I})$ distribution and $\boldsymbol{\epsilon}_i \sim \mathrm{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p \times p$ diagonal covariance matrix with $\sigma_j^2$ for $j = 1, \ldots, p$ on the diagonal. We have assumed the $\boldsymbol{y}_i$ is centered in this case. The model can be easily extended to non-centered case where we first provide a sample estimate of the mean and then subtract the mean from $\boldsymbol{y}_i$. The resulting centered observations could be modeled by (2.1). Integrating over the factor $\boldsymbol{x}_i$, the model produces a low-rank (in the sense of the loading matrix) estimation of the covariance matrix of $\boldsymbol{y}_i$

$$\boldsymbol{\Omega} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Sigma} = \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}\boldsymbol{\lambda}_{\cdot h}^\top + \boldsymbol{\Sigma}, \tag{2.2}$$

where $\boldsymbol{\lambda}_{\cdot h}$ is the $h$th column of $\boldsymbol{\Lambda}$. This factorization suggests that each factor separately contributes to the covariance of the observation through its corresponding loading. Traditional exploratory data analysis methods such as principle component analysis (PCA) (Hotelling, 1933), independent component analysis (ICA) (Comon, 1994), and canonical correlation analysis (CCA) (Hotelling, 1936) all have interpretations as a FA model.

The parameter estimation in FA is usually conducted using expectation maximization (EM) (Dempster et al., 1977) or Markov chain Monte Carlo (MCMC) algorithms. In either way all the information for parameter estimation is coming from the sample covariance matrix

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^\top.$$

This result in fact reflects that $\boldsymbol{S}$ is the sufficient statistic for $\boldsymbol{\Omega}$. In the $n < p$ case, it is important to include regularization on the covariance matrix estimation due to sample covariance is singular. In the context of FA, this can be transferred to assigning regularization on the loading matrix, generating sparse structures in $\boldsymbol{\Lambda}$. For example, element-wise sparsity in the loading corresponds to *variable selection*. This achieves the effect that a latent factor contributes to the variation of a subset of the observed variables, generating interpretable results (West, 2003; Carvalho et al., 2008; Knowles and Ghahramani, 2011). For example, in gene expression analysis, sparse factor loadings are interpreted as clusters of genes and are used to identify sets of co-regulated genes (Pournara and Wernisch, 2007; Lucas et al., 2010; Gao et al., 2013).

### *2.1.2 Extension to paired vectors*

Factor model (2.1) provides a covariance estimation for single random vector $\boldsymbol{y}_i$. However in real world applications there are common cases that paired random vec-

tors or coupled multiple vectors are observed. Let $\boldsymbol{y}_i^{(1)}, \ldots \boldsymbol{y}_i^{(m)}$ denote the $m$ coupled vectors for the $i$th subject. For example, consider the case where gene expression profiles under $m$ different conditions are measured. Each condition characterizes a $n$ repeated measurements of a random vector, and researchers are interested in the covariance specific to each condition as well as the covariance among different combinations of conditions. Another example is that $m$ different sections of $n$ documents are observed. We let the matrix $\boldsymbol{Y}^{(v)} = (\boldsymbol{y}_1^{(v)}, \ldots, \boldsymbol{y}_n^{(v)})$ denote the $n$ independent subjects under condition $v$ with $v = 1, \ldots, m$. $\boldsymbol{Y}^{(v)}$ is known as a view.

When $m = 2$, canonical correlation analysis (CCA) identifies a linear latent space and projections (canonical directions) for which the correlations between the two views are mutually maximized (Hotelling, 1936). CCA has a probabilistic interpretation as a factor model by assuming a common latent factor $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$ for both $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$ (Bach and Jordan, 2005)

$$\begin{aligned} \boldsymbol{y}_i^{(1)} &= \boldsymbol{\Lambda}^{(1)}\boldsymbol{x}_i + \boldsymbol{e}_i^{(1)}, \\ \boldsymbol{y}_i^{(2)} &= \boldsymbol{\Lambda}^{(2)}\boldsymbol{x}_i + \boldsymbol{e}_i^{(2)}. \end{aligned} \tag{2.3}$$

The errors are distributed as $\boldsymbol{e}_i^{(1)} \sim \mathrm{N}_{p_1}(\boldsymbol{0}, \boldsymbol{\Psi}^{(1)})$ and $\boldsymbol{e}_i^{(2)} \sim \mathrm{N}_{p_2}(\boldsymbol{0}, \boldsymbol{\Psi}^{(2)})$, where $\boldsymbol{\Psi}^{(1)}$ and $\boldsymbol{\Psi}^{(2)}$ are positive semi-definite matrices. The model does not restrict $\boldsymbol{\Psi}^{(1)}$ and $\boldsymbol{\Psi}^{(2)}$ to be diagonal, allowing dependencies among residual errors within a view. The maximum likelihood estimators of the loading matrices, $\boldsymbol{\Lambda}^{(1)}$ and $\boldsymbol{\Lambda}^{(2)}$, are the first $k$ canonical directions up to linear transformations (Bach and Jordan, 2005).

Building on the probabilistic CCA model, a Bayesian CCA (BCCA) model is proposed by Klami et al. (2013). BCCA model assumes

$$\begin{aligned} \boldsymbol{y}_i^{(1)} &= \boldsymbol{A}^{(1)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(1)}\boldsymbol{x}_i^{(1)} + \boldsymbol{\epsilon}_i^{(1)}, \\ \boldsymbol{y}_i^{(2)} &= \boldsymbol{A}^{(2)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(2)}\boldsymbol{x}_i^{(2)} + \boldsymbol{\epsilon}_i^{(2)}, \end{aligned} \tag{2.4}$$

with $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^{k_0\times 1}$, $\boldsymbol{x}_i^{(1)} \in \mathbb{R}^{k_1\times 1}$ and $\boldsymbol{x}_i^{(2)} \in \mathbb{R}^{k_2\times 1}$ (Figure 2.1B). The latent vector

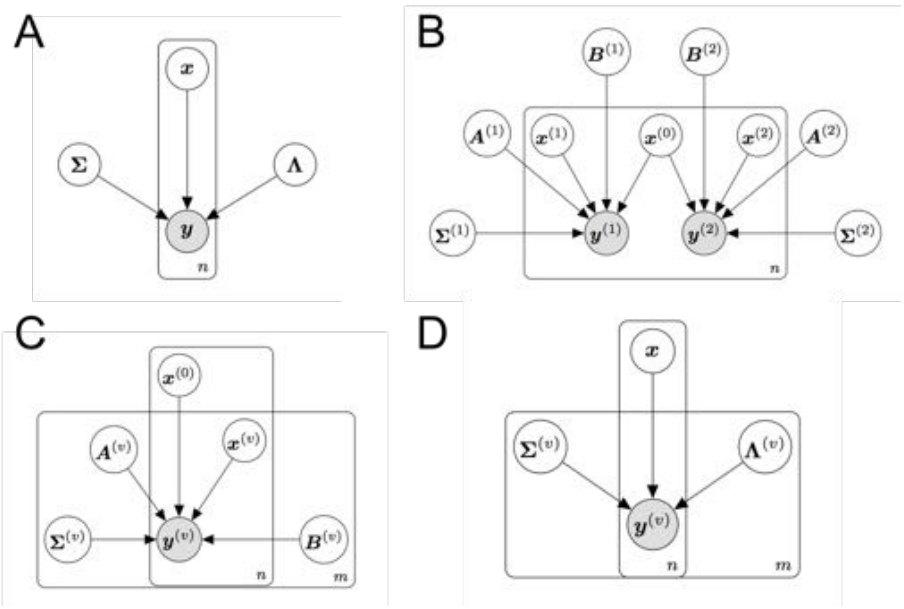


FIGURE 2.1: Graphical representations of different latent factor models. Panel A: Factor analysis model. Panel B: Bayesian canonical correlation analysis model (BCCA). Panel C: An extension of BCCA model to multiple views. Panel D: Bayesian group factor analysis model studied in current chapter.

$\boldsymbol{x}_i^{(0)}$ is shared by both $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$, and it captures their common variation through loading matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$. Two additional latent vectors, $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$, are specific to each view; they are multiplied by view specific loading matrices $\boldsymbol{B}^{(1)}$ and $\boldsymbol{B}^{(2)}$ respectively. The two residual error terms are $\boldsymbol{\epsilon}_i^{(1)} \sim \mathrm{N}_{p_1}(\mathbf{0}, \boldsymbol{\Sigma}^{(1)})$ and $\boldsymbol{\epsilon}_i^{(2)} \sim \mathrm{N}_{p_2}(\mathbf{0}, \boldsymbol{\Sigma}^{(2)})$, where $\boldsymbol{\Sigma}^{(1)}$ and $\boldsymbol{\Sigma}^{(2)}$ are two diagonal matrices. This model was originally called inter-battery factor analysis (IBFA) (Browne, 1979) and recently has been studied under a full Bayesian inference framework (Klami et al., 2013). It can be viewed as a probabilistic CCA model (2.3) with an additional low rank factorization of the error covariance matrices. In fact, we re-write the residual error term specific to view $v$ ($v = 1, 2$) from the probabilistic CCA model (2.3) as $\boldsymbol{e}_i^{(v)} = \boldsymbol{B}^{(v)}\boldsymbol{x}_i^{(v)} + \boldsymbol{\epsilon}_i^{(v)}$, then marginally $\boldsymbol{e}_i^{(v)} \sim \mathrm{N}_{p_v}(\mathbf{0}, \boldsymbol{\Psi}^{(v)})$ with $\boldsymbol{\Psi}^{(v)} = \boldsymbol{B}^{(v)}(\boldsymbol{B}^{(v)})^\top + \boldsymbol{\Sigma}^{(v)}$.

Klami et al. (2013) re-write (2.4) as a factor analysis model with *group-wise sparsity* in the loading matrix. Let $\boldsymbol{y}_i \in \mathbb{R}^{p\times 1}$ with $p = p_1 + p_2$ be the vertical

concatenation of $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$; let $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$ with $k = k_0 + k_1 + k_2$ be vertical concatenation of $\boldsymbol{x}_i^{(0)}$, $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$; and let $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p\times 1}$ be vertical concatenation of the two residual errors. Then, the BCCA model in (2.4) can be written as a factor analysis model

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i,$$

with $\boldsymbol{\epsilon}_i \sim \mathrm{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{0} \\ \boldsymbol{A}^{(2)} & \boldsymbol{0} & \boldsymbol{B}^{(2)} \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}^{(2)} \end{pmatrix}. \tag{2.5}$$

The structure in the loading matrix $\boldsymbol{\Lambda}$ has a specific meaning. The non-zero columns (those in $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$) project the shared latent factors (i.e., the first $k_0$ in $\boldsymbol{x}_i$) to $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$ respectively. These latent factors represent the covariance across the two views. The columns with zero blocks (those in $(\boldsymbol{B}^{(1)}; \boldsymbol{0})$ or $(\boldsymbol{0}; \boldsymbol{B}^{(2)})$) relate rest factors to only one of the two views. Those factors are used to model covariance specific to one view. Under this model, the structure of $\boldsymbol{\Lambda}$ could be fixed *a priori*, and the inference problem is to estimate the entries in the non-zero blocks of $\boldsymbol{\Lambda}$.

### *2.1.3 Multiple coupled vectors*

The extensions of the BCCA/IBFA model to allow multiple views ($m > 2$) have been developed recently. Examples include McDonald (1970); Browne (1980); Archambeau and Bach (2009); Qu and Chen (2011); Ray et al. (2014). Those extensions partition latent variables to shared and view specific ones through following equation

$$\boldsymbol{y}_i^{(v)} = \boldsymbol{A}^{(v)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(v)}\boldsymbol{x}_i^{(v)} + \boldsymbol{\epsilon}_i^{(v)} \quad \text{for} \quad v = 1, \cdots, m. \tag{2.6}$$

By vertical concatenation of $\boldsymbol{y}_i^{(v)}$, $\boldsymbol{x}_i^{(v)}$ and $\boldsymbol{\epsilon}_i^{(v)}$, this model can be viewed as a latent factor model with the joint loading matrix $\boldsymbol{\Lambda}$ having a group-wise sparsity pattern

similar as in the BCCA/IBFA model

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{B}^{(1)} & \cdots & \mathbf{0} \\ \boldsymbol{A}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(m)} & \mathbf{0} & \cdots & \boldsymbol{B}^{(m)} \end{pmatrix}. \tag{2.7}$$

Here, the first column of blocks ($\boldsymbol{A}^{(v)}$) is a non-zero loading matrix across all views, and the remaining columns have a block diagonal structure with view-specific loading matrices ($\boldsymbol{B}^{(v)}$) on the diagonal. However, those extensions are limited by the strict diagonal structure of the loading matrix. Structuring the loading matrix in this way prevents the model from capturing covariance among arbitrary combinations of views.

The structure of $\boldsymbol{\Lambda}$ in (2.7) has been relaxed to model covariance among combinations of views (Jia et al., 2010; Virtanen et al., 2012; Klami et al., 2014a). In the relaxed formulation, each view $\boldsymbol{y}_i^{(v)}$ is modeled by its own loading matrix $\boldsymbol{\Lambda}^{(v)}$ and a latent vector $\boldsymbol{x}_i$, and this latent vector $\boldsymbol{x}_i$ is shared by all views (Figure 2.1D)

$$\boldsymbol{y}_i^{(v)} = \boldsymbol{\Lambda}^{(v)}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i^{(v)} \quad \text{for} \quad v = 1, \ldots, m. \tag{2.8}$$

By allowing columns in $\boldsymbol{\Lambda}^{(v)}$ to be zero, the model decouples certain latent factors from certain views, achieving view selection. The covariance structure of an arbitrary combination of views is modeled by factors with non-zero loadings corresponding to the views in that combination. Factors that correspond to non-zero entries for only one view capture covariance specific to that view. The model in (2.8) is named group factor analysis (GFA) model (Virtanen et al., 2012).

## 2.2 Structures in the loading

The loading matrix $\boldsymbol{\Lambda}$ plays an important role in previous FA models and their extensions. As mentioned before, element-wise sparsity in loadings achieves variable

selection and generates interpretable results. When dealing with multiple views, group-wise sparsity decouples views from latent factors, achieving view selection. Imposing different levels of sparsity has been investigated through different strategies under various contexts, both using classical penalization or Bayesian shrinkage priors. For classical penalization, the elastic net (Zou and Hastie, 2005) and group Lasso (Yuan and Lin, 2006) penalties have been featured in regression models. Mixed matrix norms with $\ell_1$ norm penalizing each column and either $\ell_2$ or $\ell_\infty$ norms penalizing the elements have been used in GFA context (Jia et al., 2010). More sophisticated structured penalties have been studied. Examples include Kowalski and Torrésani (2009), Jenatton et al. (2011) and Huang et al. (2011) among others. In this thesis we focus on the approach using Bayesian methods.

### *2.2.1 Spike and slab prior in factor analysis*

A classic Bayesian approach to variable selection is to develop a two-component mixture prior, termed spike-and-slab prior, for the variables of interests (Mitchell and Beauchamp, 1988; West, 2003; Carvalho et al., 2008). The spike component corresponds to a probability mass at zero, and the slab component corresponds to a relatively diffuse prior on the parameter space. The spike component also has been formulated as a normal with a small variance. See George and McCulloch (1993) and Ročková and George (2014) for example. This prior has an elegant interpretability by estimating the probability that certain variables are excluded, modeled by the spike component or included, modeled by the slab component.

In the FA context, West (2003) develops a spike-and-slab prior on every element of the loading matrix, with mixture weight being shared cross the loading. Let $\lambda_{jh}$ denote the entry of $j$th row and $h$th column in the loading matrix $\mathbf{\Lambda}$, the prior assumes

$$\lambda_{jh}|\pi_h, \tau_h \sim (1-\pi_h)\delta_0(\cdot) + \pi_h \mathrm{N}(0, \tau_h). \tag{2.9}$$

When $p$ is large, we hope this prior could generate a large amount of zeros in the $h$th loading, therefore the hyper-prior of $\pi_h$ should have substantial mass around zero. Lucas et al. (2006) and Carvalho et al. (2008) extend this idea to allow each loading element has its own mixture weight, and the mixture weights in a loading is assumed to share a common population beta distribution

$$\lambda_{jh}|\pi_{jh}, \tau_h \sim (1 - \pi_{jh})\delta_0(\cdot) + \pi_{jh}\mathrm{N}(0, \tau_h). \tag{2.10}$$

Allowing each loading entry to have its own mixture weight $\pi_{jh}$ could overcome some limitations of the prior in equation (2.9): The strong informative prior assigned to $\pi_h$ in equation (2.9) could lead the posterior of $\lambda_{jh}$ being zero to be diffuse across the unit interval, therefore generating a large variance (Carvalho et al., 2008). Allowing $\lambda_{jh}$ to have its own mixture weight overcomes this limitation. It is possible that the entry specific weight $\pi_{jh} \to 0$, effectively setting $\lambda_{jh}$ to zero. This new prior controls the sparsity level of a loading and allows elements to borrow information within a loading. This idea is quite related to our GFA model in Section 2.3. Moreover, nonparametric methods like Indian buffet process (IBP) have been used to model the inclusion/exclusion of loading elements (Knowles and Ghahramani, 2011). Using nonparametric methods generates a conceptually infinite number of latent factors which allows the model itself to determine the dimension of latent space. However parameter estimations in those approaches often rely on Gibbs sampling methods which construct a Markov chain that performs stochastic search in an exponentially increasing configuration space (George and McCulloch, 1993). Recently an EM algorithm has been proposed to find posterior modes under the specification of the spike component being a tight normal in a linear regression model (Ročková and George, 2014). The approach allows fast parameter estimation and variable selection can be generated by posterior thresholding rules.

### *2.2.2 Continuous shrinkage priors*

Recently, scale mixtures of normal priors have been proposed as a computationally efficient alternative to the two component spike-and-slab prior (West, 1987; Carvalho et al., 2010; Polson and Scott, 2011; Armagan et al., 2011, 2013; Bhattacharya et al., 2015). A prior of such kind generally assumes a normal distribution with a mixed variance term. The mixing distribution of the variance determines the behavior of the prior. In some sense those priors could be viewed as extending the previous two components mixture to an infinite mixture of normals. Marginalizing over the variance, we would expect such priors have substantial probability mass around zero, which pushes small effects toward zero, and heavy tails, which allow large signals to escape from substantial shrinkage. For example, the inverse-gamma distribution on the variance term results in an automatic relevance determination (ARD) prior (Tipping, 2001). An exponential distribution on the variance term results in a Laplace prior (Park and Casella, 2008). The horseshoe prior, with a half Cauchy distribution on the standard deviation as the mixing distribution, has become popular due to its strong shrinkage around the origin and heavy tails (Carvalho et al., 2010). More general classes of priors including generalized beta mixtures of normals and double Pareto priors have been developed recently (Armagan et al., 2011, 2013). See Table 2.1 for a brief summary. More details can be found in the corresponding references. These various continuous shrinkage priors provide a one group answer to the original two groups question (Polson and Scott, 2011). Although such an answer can not provide an estimation of variable inclusion probability, it has many intriguing advantages as discussed by various researchers. See Polson and Scott (2011), Carvalho et al. (2010) and Bhattacharya et al. (2015) among others. For example, maximum a posteriori (MAP) estimator could provide exact zeros in the variables of interest, and continuous priors avoid to construct a Markov chain that has exponentially

Table 2.1: A brief summary of Bayesian shrinkage priors. More details can be found in Polson and Scott (2011) and references therein.

| Name | Prior for $\theta_j$ | Mixing density |
|---|---|---|
| ARD | $\theta_j \vert \tau_j \sim \mathrm{N}(0, \tau_j)$ | $\tau_j^{-1} \sim \mathrm{Ga}(a, b)$, small $a$ and $b$ |
| Laplace | $\theta_j \vert \tau_j \sim \mathrm{N}(0, \tau_j)$ | $\tau_j \sim \mathrm{Exp}(\lambda^2/2)$ |
| Strawderman -Berger | $\theta_j \vert \rho_j \sim \mathrm{N}(0, \rho_j^{-1} - 1)$ | $\rho_j \sim \mathrm{Be}(1/2, 1)$ |
| Horseshoe | $\theta_j \vert \tau_j \sim \mathrm{N}(0, \tau_j)$<br>$\theta_j \vert \rho_j \sim \mathrm{N}(0, \rho_j^{-1} - 1)$<br>$\theta_j \vert \tau_j \sim \mathrm{N}(0, \tau_j)$ | $\tau_j^{1/2} \sim \mathrm{C}^+(0, \phi^{1/2})$, $\phi = 1$, standard<br>$\rho_j \sim \rho_j^{1/2}(1-\rho_j)^{1/2}\frac{1}{1+(\phi-1)\rho_j}$<br>$\tau_j \sim \mathrm{Ga}(1/2, \lambda_j)$, $\lambda_j \sim \mathrm{Ga}(1/2, \phi)$ |
| Generalized double Pareto | $\theta_j \sim 1/(2\xi)(1 + \vert\tau_j\vert/\alpha\xi)^{-(\alpha+1)}$<br>$\theta_j \vert \tau_j \sim \mathrm{N}(0, \tau_j)$ | -<br>$\tau_j \sim \mathrm{Exp}(\lambda_j^2/2), \lambda_j \sim \mathrm{Ga}(a, \eta)$ |

increasing number of parameter configurations.

In the FA context, Bhattacharya and Dunson (2011) propose an infinite factor model using a multiplicative gamma process shrinkage prior on the loading matrix. The variance of the loading element is composed by a product of a global shrinkage parameter specified for each loading and a local shrinkage parameter for that element. This structure is similar to the global/local shrinkage discussed in linear regression contexts (Polson and Scott, 2011) and is conceptually related to the spike-and-slab prior developed by Carvalho et al. (2008) in terms of allowing loading element to have its own shrinkage parameter. One distinct feature of the multiplicative gamma process prior is it generates the effect that loadings are shrunk more heavily as their column indices increasing. With the size of latent space growing, the factors become less important in contributing to the covariance (see (2.2)) therefore can be deleted in downstream analysis. Gao et al. (2013) use another prior called three parameter beta (TPB) prior in FA context. Their prior is equivalent to the horseshoe prior formulated in a hierarchy with three levels. The hierarchy introduces three levels of shrinkage. We are going to introduce the prior in next section in detail and use the

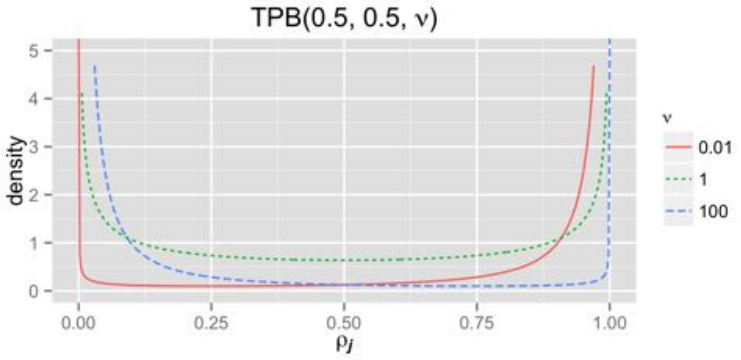


FIGURE 2.2: Density of three parameter beta (TPB) distribution with $a = b = 1/2$ with different values of $\nu$.

prior in the GFA context.

## 2.3 A new Bayesian group factor analysis model

The three parameter beta (TPB) distribution for a random variable $0 < Z < 1$ has the following density (Armagan et al., 2011)

$$f(z; a, b, \nu) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \nu^b z^{b-1} (1-z)^{a-1} [1 + (\nu - 1)z]^{-(a+b)}, \tag{2.11}$$

where $a > 0, b > 0$ and $\phi > 0$. We denote this distribution as TPB$(a, b, \nu)$. When $0 < a < 1$ and $0 < b < 1$, the distribution is bimodal, with two modes at 0 and 1 respectively. When $a = b = 1/2$, this prior becomes the class of horseshoe priors studied by Carvalho et al. (2010). We call $\nu$ the *variance parameter*. With fixed $a$ and $b$, smaller values of $\nu$ put greater probability on $z = 1$, while larger values of $\nu$ move the probability mass towards $z = 0$ (Armagan et al., 2011) (Figure 2.2). With $\nu = 1$, this distribution becomes a beta distribution Be$(b, a)$.

Let $\lambda$ denote the parameter to which we are performing variable selection. We assign the following TPB normal (TPBN) scale mixture prior to $\lambda$

$$\lambda|\varphi \sim \mathrm{N}(0, 1/\varphi - 1), \qquad \text{with} \qquad \varphi \sim \mathrm{TPB}(a, b, \nu),$$

where the *shrinkage parameter* $\varphi$ follows a TPB distribution. With $a = b = 1/2$ and $\nu = 1$ above prior becomes the second formulation of the standard horseshoe prior in Table 2.1. The bimodal property of $\varphi$ induces two distinct shrinkage behaviors. The mode near one encourages $1/\varphi - 1$ towards zero and induces strong shrinkage on $\lambda$. The mode near zero encourages $1/\varphi - 1$ towards infinity and generates a diffuse prior. Further decreasing the variance parameter $\nu$ puts more support on stronger shrinkage (Armagan et al., 2011). If we let $\theta = 1/\varphi - 1$, then this mixture prior has the following hierarchical representation

$$\lambda \sim \mathrm{N}(0, \theta), \quad \theta \sim \mathrm{Ga}(a, \delta), \quad \delta \sim \mathrm{Ga}(b, \nu).$$

In previous work, Gao et al. (2013) extend the prior to three levels of a hierarchical structure and apply it to a FA model. The three levels are formularized as

$$\begin{aligned}
\varrho &\sim \mathrm{TPB}(1/2, 1/2, \nu), && \text{Level 1} \\
\zeta_h &\sim \mathrm{TPB}(1/2, 1/2, 1/\varrho - 1), && \text{Level 2} \\
\varphi_{jh} &\sim \mathrm{TPB}(1/2, 1/2, 1/\zeta_h - 1), && \text{Level 3} \\
\lambda_{jh} &\sim \mathrm{N}(0, 1/\varphi_{jh} - 1). && (2.12)
\end{aligned}$$

At each of the three levels, a TPB distribution with horseshoe parameterization is used to induce shrinkage with its own variance parameter ($\nu$ in (2.11)), which has a further TPB distribution on previous hierarchy. We set the variance parameter in the first level to 1, generating a standard horseshoe prior in the first level. Specifically, in the first level the shrinkage parameter $\varrho$ applies horseshoe shrinkage across all columns of the loading matrix, and jointly adjusts the support of $\zeta_h$ at either zero or one. This can be interpreted as inducing sufficient shrinkage across loading columns to identify the number of factors supported by the observed data. In particular, when $\zeta_h$ is close to one, all elements in the loading are zero, inducing column-wise shrinkage. The shrinkage parameter $\zeta_h$ in second level adjusts the shrinkage applied

to each element of the $h$th loading, estimating the column-wise shrinkage by borrowing strength across all elements in that loading. The last shrinkage parameter $\varphi_{jh}$ creates element-wise sparsity in the loading matrix through a TPBN.

The three levels are further extended to jointly model sparse and dense components (Gao et al., 2013). This is achieved by assigning a two component mixture to the third level shrinkage parameter

$$\varphi_{jh} \sim \pi \cdot \text{TPB}(1/2, 1/2, 1/\zeta_h - 1) + (1 - \pi) \cdot \delta_{\zeta_h}(\cdot), \tag{2.13}$$

where $\delta_{\zeta_h}(\cdot)$ is the Dirac delta function concentrated at $\zeta_h$. The effect of the two component mixture is that the local shrinkage parameter $\varphi_{jh}$ in (2.12) could select between the third level or the second level. When it is from the second level, in which case $\varphi_{jh} = \zeta_h$, the elements in the $h$th loading follows a shared normal prior $\lambda_{jh} \sim \text{N}(0, 1/\zeta_h - 1)$. Depending on the shrinkage parameter $\zeta_h$, two effects will be generated for the $h$th loading. When $\zeta_h$ is close to zero, the whole loading is assigned a diffuse normal prior and with probability one no elements will become zero. In contrast, when $\zeta_h$ is close to one, elements in that loading are heavily pushed toward zero, generating column-wise sparsity. We call factors corresponding to such loadings *dense* factors. The motivation is that, in applications such as the analysis of gene expression data, it has been shown that much of the variation in the observation is due to technical (e.g., batch) or biological effects (e.g., sex, ethnicity), which impact a large number of genes (Leek et al., 2010). Therefore, the loadings corresponding to these effects will not be sparse. Equation (2.13) allows the local sparsity on the loading to select between element-wise sparsity or column-wise sparsity. Jointly modeling sparse and dense factors combines the idea of low-rank covariance estimation with interpretability of factors (Zou et al., 2006; Parkhomenko et al., 2009). Those dense factors capture the low-rank approximation of the covariance matrix. They usually explain a large proportion of variance in the model. The sparse factors describe

the impulse signals in observations and facilitate the interpretation of latent factors (Chandrasekaran et al., 2011).

Using the third formulation of the horseshoe prior in Table 2.1, we write 2.12 as

$$\begin{aligned}
\gamma &\sim \mathrm{Ga}(1/2, \nu),\ \eta \sim \mathrm{Ga}(1/2, \gamma), \\
\tau_h &\sim \mathrm{Ga}(1/2, \eta),\ \phi_h \sim \mathrm{Ga}(1/2, \tau_h), \\
\delta_{jh} &\sim \mathrm{Ga}(1/2, \phi_h),\ \theta_{jh} \sim \mathrm{Ga}(1/2, \delta_{jh}), \\
\lambda_{jh} &\sim \mathrm{N}(0, \theta_{jh}),
\end{aligned} \tag{2.14}$$

with sparse/dense mixture

$$\theta_{jh} \sim \pi \cdot \mathrm{Ga}(1/2, \delta_{jh}) + (1 - \pi) \cdot \delta_{\phi_h}(\cdot). \tag{2.15}$$

We assign the prior in (2.14) and (2.15) to the view specific loading $\mathbf{\Lambda}^{(v)}$ in (2.8) to develop a new GFA model. We call our model BASS standing for Bayesian group factor Analysis with Structured Sparsity. We summarize BASS as follows. The generative model for $m$ coupled views $\boldsymbol{y}_i^{(v)}$ is

$$\boldsymbol{y}_i^{(v)} = \mathbf{\Lambda}^{(v)} \boldsymbol{x}_i + \boldsymbol{\epsilon}_i^{(v)}, \quad \text{for} \quad v = 1, \ldots, m, i = 1, \ldots, n.$$

We re-write this model as a factor model by concatenating the $m$ vectors for subject $i$ into vector $\boldsymbol{y}_i$

$$\boldsymbol{y}_i = \mathbf{\Lambda} \boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \text{ with } \boldsymbol{x}_i \sim \mathrm{N}_k(0, \boldsymbol{I}), \text{ and } \boldsymbol{\epsilon}_i \sim \mathrm{N}_p(0, \mathbf{\Sigma}), \tag{2.16}$$

where $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2)$. We assign following priors on the parameters in the model

$$\begin{aligned}
&\mathbf{\Lambda}^{(v)} \sim (2.14), (2.15),\ \pi^{(v)} \sim \mathrm{Be}(1, 1), \text{ for } v = 1, \ldots, m; \\
&\sigma_j^{-2} \sim \mathrm{Ga}(a_\sigma, b_\sigma) \text{ for } j = 1, \ldots, p.
\end{aligned}$$

$a_\sigma$ and $b_\sigma$ are set to 1 and 0.3 respectively to allow a relatively wide support of variances (Bhattacharya and Dunson, 2011). These values correspond to a prior with mean of 3.3 and variance of 11 of the precision parameter $\sigma_j^{-2}$.

## 2.4 Fast parameter estimation via parameter expanded EM

Given our setup, the full joint distribution of BASS factorizes as

$$\begin{aligned} & p(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{Z}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ & \quad = p(\boldsymbol{Y}|\boldsymbol{\Lambda}, \boldsymbol{X}, \boldsymbol{\Sigma}) p(\boldsymbol{X}) \\ & \qquad \times p(\boldsymbol{\Lambda}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{\Delta}, \boldsymbol{Z}, \boldsymbol{\Phi}) p(\boldsymbol{\Delta}|\boldsymbol{\Phi}) p(\boldsymbol{\Phi}|\boldsymbol{T}) p(\boldsymbol{T}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\boldsymbol{\gamma}) \\ & \qquad \times p(\boldsymbol{\Sigma}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}), \end{aligned} \tag{2.17}$$

where $\boldsymbol{\Theta} = \{\theta_{jh}^{(v)}\}$, $\boldsymbol{\Delta} = \{\delta_{jh}^{(v)}\}$, $\boldsymbol{\Phi} = \{\phi_h^{(v)}\}$, $\boldsymbol{T} = \{\tau_h^{(v)}\}$, $\boldsymbol{\eta} = \{\eta^{(v)}\}$ and $\boldsymbol{\gamma} = \{\gamma^{(v)}\}$ are the collections of the prior parameters in (2.14). The posterior distributions of the model parameters could be either simulated through MCMC algorithms or approximated using variational Bayes inference. We propose an MCMC algorithm using block update of the loading matrix (Bhattacharya and Dunson, 2011). The algorithm updates the loading matrix row by row, enabling fast mixing behavior. Details of the algorithm can be found in Appendix A.1.

### *2.4.1 Standard EM*

In this study, we are interested in a structured solution of the loading matrix. Therefore we find a maximum a posteriori (MAP) estimator of the model using an expectation maximization (EM) algorithm (Dempster et al., 1977). The latent factors $\boldsymbol{X}$ and the indicator variables $\boldsymbol{Z}$ are treated as missing data and are estimated in E step, and the rest parameters are estimated in M step. Let $\boldsymbol{\Xi} = \{\boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$ be the collection of the parameters optimized in M step. The complete log likelihood ($Q$ function) could be written as

$$Q(\boldsymbol{\Xi}|\boldsymbol{\Xi}_{(s)}) = \mathbb{E}_{\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Xi}_{(s)}, \boldsymbol{Y}} \log \big(p(\boldsymbol{\Xi}, \boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{Y})\big). \tag{2.18}$$

Since $\boldsymbol{X}$ and $\boldsymbol{Z}$ are conditional independent given $\boldsymbol{\Xi}$, the expectation can be easily calculated using the full conditionals of $\boldsymbol{X}$ and $\boldsymbol{Z}$ derived in the MCMC algorithm.

In M step when estimating $\mathbf{\Lambda}$, the loadings specific to each view are estimated jointly. We summarize our EM algorithm as follows.

*Expectation Step* Given model parameters, the distribution of latent factor $\boldsymbol{X}$ has shown in Appendix A.1. The expectation of first and second moments of $\boldsymbol{X}$ can be derived as

$$\langle \boldsymbol{x}_{\cdot i} \rangle = (\mathbf{\Lambda}^\top \mathbf{\Sigma}^{-1} \mathbf{\Lambda} + \boldsymbol{I})^{-1} \mathbf{\Lambda}^\top \mathbf{\Sigma}^{-1} \boldsymbol{y}_{\cdot i},$$
$$\langle \boldsymbol{x}_{\cdot i} \boldsymbol{x}_{\cdot i}^\top \rangle = \langle \boldsymbol{x}_{\cdot i} \rangle \langle \boldsymbol{x}_{\cdot i} \rangle^\top + (\mathbf{\Lambda}^\top \mathbf{\Sigma}^{-1} \mathbf{\Lambda} + \boldsymbol{I})^{-1}.$$

The expectation of indicator variable $\rho_h^{(v)} = \langle z_h^{(v)} \rangle$ is

$$\rho_h^{(v)} = \frac{\pi^{(v)} \prod_{j=1}^{p_v} \mathrm{N}(\lambda_{jh}^{(v)}; 0, \theta_{jh}^{(v)}) \mathrm{Ga}(\theta_{jh}^{(v)}; a, \delta_{jh}^{(v)}) \mathrm{Ga}(\delta_{jh}^{(v)}; b, \phi_h^{(v)})}{(1-\pi^{(v)}) \prod_{j=1}^{p_w} \mathrm{N}(\lambda_{jh}^{(v)}; 0, \phi_h^{(v)}) + \pi^{(v)} \prod_{j=1}^{p_w} \mathrm{N}(\lambda_{jh}^{(v)}; 0, \theta_{jh}^{(v)}) \mathrm{Ga}(\theta_{jh}^{(v)}; a, \delta_{jh}^{(v)}) \mathrm{Ga}(\delta_{jh}^{(v)}; b, \phi_h^{(v)})}.$$

*Maximization Step* The log posterior of $\mathbf{\Lambda}$ can be written as

$$\log\left(p(\mathbf{\Lambda}|-)\right) \propto \mathrm{tr}\left(\mathbf{\Sigma}^{-1} \mathbf{\Lambda} \boldsymbol{S}^{XY}\right) - \frac{1}{2} \mathrm{tr}\left(\mathbf{\Lambda}^\top \mathbf{\Sigma}^{-1} \mathbf{\Lambda} \boldsymbol{S}^{XX}\right) - \frac{1}{2} \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}^\top \boldsymbol{D}_h \boldsymbol{\lambda}_{\cdot h},$$

where

$$\boldsymbol{D}_h = \mathrm{diag}\left( \frac{\rho_h^{(1)}}{\theta_{1h}^{(1)}} + \frac{1-\rho_h^{(1)}}{\phi_h^{(1)}}, \cdots, \frac{\rho_h^{(m)}}{\theta_{p_m h}^{(m)}} + \frac{1-\rho_h^{(m)}}{\phi_h^{(m)}} \right),$$
$$\boldsymbol{S}^{XY} = \sum_{i=1}^{n} \langle \boldsymbol{x}_{\cdot i} \rangle \boldsymbol{y}_{\cdot i}^\top, \text{ and } \boldsymbol{S}^{XX} = \sum_{i=1}^{n} \langle \boldsymbol{x}_{\cdot i} \boldsymbol{x}_{\cdot i}^\top \rangle.$$

We take the derivative with respect to the loading column $\boldsymbol{\lambda}_{\cdot h}$ to get the MAP solution. For the first part in the right side of the proportion,

$$\begin{aligned} \frac{\partial \mathrm{tr}\left(\mathbf{\Sigma}^{-1} \mathbf{\Lambda} \boldsymbol{S}^{XY}\right)}{\partial \boldsymbol{\lambda}_{\cdot h}} &= (\mathbf{1}_k^h \otimes \boldsymbol{I}_p) \times \mathrm{vec}\left(\mathbf{\Sigma}^{-1} \boldsymbol{S}^{YX}\right) = \mathrm{vec}\left(\mathbf{\Sigma}^{-1} \boldsymbol{S}^{YX} \mathbf{1}_k^h\right) \\ &= \mathbf{\Sigma}^{-1} \boldsymbol{S}^{YX} \mathbf{1}_k^h, \end{aligned}$$

where vec is vectorization operation of a matrix, $\otimes$ is the Kronecker product, $\mathbf{1}_k^h \in \mathbb{R}^{k\times 1}$ is a zero vector except $h$th row being 1, and $\boldsymbol{S}^{YX} = (\boldsymbol{S}^{XY})^\top$. For the second part,

$$\begin{aligned}\frac{\partial \text{tr}\left(\boldsymbol{\Lambda}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX}\right)}{\partial \boldsymbol{\lambda}_{\cdot h}} &= 2(\mathbf{1}_k^h \otimes \boldsymbol{I}_p) \times \text{vec}\left(\boldsymbol{\Sigma}^{-1}\Lambda\boldsymbol{S}^{XX}\right) = 2\text{vec}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX}\mathbf{1}_k^h\right) \\ &= 2\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX}\mathbf{1}_k^h.\end{aligned}$$

For the third part, the derivative is $\boldsymbol{D}_h\boldsymbol{\lambda}_{\cdot h}$. The MAP of $\boldsymbol{\lambda}_{\cdot h}$ can be obtained by setting the derivative to zero, resulting

$$\widehat{\boldsymbol{\lambda}}_{\cdot h} = \left(s_{hh}^{XX}\boldsymbol{I}_p + \boldsymbol{\Sigma}\boldsymbol{D}_h\right)^{-1}\left(\boldsymbol{s}_{\cdot h}^{YX} - \sum_{h'\neq h}\boldsymbol{\lambda}_{\cdot h'}s_{h'h}^{XX}\right), \tag{2.19}$$

where $s_{ij}^{XX}$ is the $(i,j)$th element of $\boldsymbol{S}^{XX}$, and $\boldsymbol{s}_{\cdot h}^{YX}$ is the $h$th column of $\boldsymbol{S}^{YX}$. The matrix needed inverse is a diagonal matrix. Therefore $\widehat{\boldsymbol{\lambda}}_{\cdot h}$ can be calculated efficiently. The MAP of other model parameters can be obtained straightforwardly from their full conditional distributions. Results are listed in Appendix A.2

### *2.4.2 Parameter expanded EM*

The standard latent factor model in (2.1) is unidentifiable up to orthonormal transformations. For any orthogonal matrix $\boldsymbol{P}$ with $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}$, the new parameter with $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda}\boldsymbol{P}^\top$ and $\boldsymbol{x}' = \boldsymbol{P}\boldsymbol{x}$ produces identical likelihood. When FA is used for prediction or covariance estimation, the identifiability problem does not pose particular ambiguities. However it does cause difficulties in factor interpretation. One traditional solution is to restrict the loading matrix to be lower-triangular and the diagonal elements to be positive (West, 2003; Carvalho et al., 2008). This approach gives special roles to the first $k$ variables in $\boldsymbol{y}_i$, therefore they must be selected carefully (Carvalho et al., 2008). To handle this undesirable behavior, we propose an parameter expanded EM algorithm (Liu et al., 1998) that favors loading orientations

with desirable structures that match our prior. This idea comes from the recent work of Ročková and George (2015). We now introduce our algorithm in detail.

The unidentifiability problem comes from the rotation invariance. Once the model parameters are initialized, the original EM algorithm becomes difficult to escape from local suboptimal regions with undesirable loading orientations. It is due to the strong coupling effects between the updates of loading matrix and latent factors, making the algorithm converge poorly. Parameter expansion (PX) has been shown to reduce this coupling effect by introducing expansion parameters (Liu et al., 1998; van Dyk and Meng, 2001; Liu and Wu, 1999). PX has been studied under both deterministic (Liu et al., 1998) and stochastic (Liu and Wu, 1999) optimization algorithms. Usually those expansion parameters are chosen such that observed data likelihood dose not depend on them after latent variables integrated out (Liu and Wu, 1999). However they must be identifiable under the complete data likelihood with latent variables introduced. When the expansion parameters are independent of the prior assigned to parameters of interest, the posterior is invariant under parameter expansion (Liu and Wu, 1999).

We extend our model in (2.16) to a parameter expanded version as

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{A}_L^{-1}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i,\ \boldsymbol{x}_i \sim \mathrm{N}_k(\boldsymbol{0}, \boldsymbol{A}),\ \boldsymbol{\epsilon}_i \sim \mathrm{N}_k(\boldsymbol{0}, \boldsymbol{\Sigma}), \tag{2.20}$$

where $\boldsymbol{A}_L$ is the lower triangular part of Cholesky decomposition of $\boldsymbol{A}$. The covariance of $\boldsymbol{y}_i$ is invariant under this expansion, therefore generating the same likelihood. Note $\boldsymbol{A}_L^{-1}$ is not an orthogonal matrix, however it contains a orthogonal transformation through polar decomposition as discussed by Ročková and George (2015). We let $\boldsymbol{\Lambda}^\star = \boldsymbol{\Lambda}\boldsymbol{A}_L^{-1}$ and assign our prior on this *rotated* loading matrix. One way of thinking this parameter expansion is while keeping likelihood invariant, we are trying to find an orientation that best fits our prior structure of the joint loading matrix by rotating it to desired sparsity while sacrificing the independent assumption

of factors. However, we must emphasize that the posterior of $\boldsymbol{\Lambda}$ is not the same as the posterior in our original model. This is because the prior assigned to $\boldsymbol{\Lambda}$ depends on the expansion parameter $\boldsymbol{A}$, generating the dependence between the posterior of $\boldsymbol{\Lambda}$ and $\boldsymbol{A}$.

We let $\boldsymbol{\Xi}^\star = \{\boldsymbol{\Lambda}^\star, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$ and the parameters of our expanded model are $\{\boldsymbol{\Xi}^\star \cup \boldsymbol{A}\}$. The EM algorithm in this expanded parameter space generates a sequence $\{{\boldsymbol{\Xi}^\star}_{(1)} \cup \boldsymbol{A}_{(1)}, {\boldsymbol{\Xi}^\star}_{(2)} \cup \boldsymbol{A}_{(2)}, \cdots\}$. This sequence corresponds to a sequence of parameter estimations in original space $\{\boldsymbol{\Xi}_{(1)}, \boldsymbol{\Xi}_{(2)}, \cdots\}$ with $\boldsymbol{\Lambda}$ in the original space being $\boldsymbol{\Lambda}^\star \boldsymbol{A}_L$ (Ročková and George, 2015). At every iteration we initialize $\boldsymbol{A}_{(s)} = \boldsymbol{I}_k$. Then the new $Q$ function could be written as

$$Q(\boldsymbol{\Xi}^\star, \boldsymbol{A}|\boldsymbol{\Xi}_{(s)}) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Xi}_{(s)},\boldsymbol{Y},\boldsymbol{A}_{(s)}} \log\big(p(\boldsymbol{\Xi}^\star, \boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{Y})\big). \tag{2.21}$$

We call our new EM algorithm PX-EM. The conditional distributions of $\boldsymbol{X}$ and $\boldsymbol{Z}$ still factorizes in the expectation. However the distribution of $\boldsymbol{x}_i$ depends on expansion parameter $\boldsymbol{A}$. The full joint distribution in (2.16) only changes $p(\boldsymbol{X})$, with $\boldsymbol{\Lambda}^\star$ substituting $\boldsymbol{\Lambda}$. Therefore the M step for $\boldsymbol{\Xi}^\star$ does not change. The only term involving $\boldsymbol{A}$ is $p(\boldsymbol{X})$. Thus the value of $\boldsymbol{A}$ that maximizes (2.21) can be solved by finding

$$\boldsymbol{A}_{(s+1)} = \text{argmax}_{\boldsymbol{A}} Q(\boldsymbol{\Xi}^\star, \boldsymbol{A}|\boldsymbol{\Xi}_{(s)}) = \text{argmax}_{\boldsymbol{A}} \bigg(\text{const} - \frac{n}{2}\log|\boldsymbol{A}| - \frac{1}{2}\text{tr}\big(\boldsymbol{A}^{-1}\boldsymbol{S}^{XX}\big)\bigg),$$

where $\boldsymbol{S}^{XX}$ is defined in the original EM algorithm. The solution is simply $\boldsymbol{A}_{(s+1)} = \frac{1}{n}\boldsymbol{S}^{XX}$. For the E step, the $\boldsymbol{\Lambda}$ in the original space is first calculated and the expectation is taken in the original model. The details of the updates of PX-EM algorithm are shown in Appendix A.3.

As discussed before, the proposed PX-EM only keeps the likelihood invariant but does not leave the prior invariant under transformation. Therefore it differs from the PX-EM studied by Liu et al. (1998), as discussed in Ročková and George

(2015). Therefore we run our PX-EM only for a first small number of iterations and then switch to our original EM algorithm targeting desired posterior modes of $\boldsymbol{\Lambda}$. The first couple runs of PX-EM greatly facilitate our model to escape from bad loading orientations, as shown in simulation studies. By introducing extra parameter $\boldsymbol{A}$, the posterior modes in original space are intersected with equal likelihood curves indexed by $\boldsymbol{A}$ in expanded space. Those curves serve to facilitate the traverse between posterior modes in original space and generate prior favorable orientations in the loading matrix (Ročková and George, 2015).

### *2.4.3 Computation complexity*

The computational complexity of our block Gibbs sampler is relative demanding. Updating each loading row requires first inversion of a $k \times k$ matrix with $O(k^3)$ complexity and then calculating the mean with $O(k^2 n)$ complexity. The complexity of updating whole joint loading matrix requires $p$ times this calculation. Other updates are in lower order compared to updating loading. Therefore our Gibbs sampler has $O(k^3 p + k^2 pn)$ complexity per iteration. In our EM algorithm, E step requires $O(k^3)$ for a matrix inversion, $O(k^2 p + kpn)$ for calculating first moment, and $O(k^2 n)$ for calculating second moment. Calculations in M step are in lower order. Therefore the original EM algorithm has $O(k^3 + k^2 p + k^2 n + kpn)$ complexity per iteration. Our PX-EM algorithm introduce an additional Cholesky decomposition with $O(k^3)$ and a matrix multiplication with $O(k^2 p)$. The total complexity is in the same order as the original EM algorithm.

## 2.5 Simulations

This this section we evaluate the performance of BASS in six simulation studies. Their details are provided in Table 2.2.

Table 2.2: Summary of six simulation studies to test the performance of BASS

| Simulations | Views | Dimensions |
|---|---|---|
| *Sim1* | 2 | $p_1 = 100, p_2 = 120$ |
| *Sim2* | 2 | $p_1 = 100, p_2 = 120$ |
| *Sim3* | 4 | $p_1 = 70, p_2 = 60, p_3 = 50, p_4 = 40$ |
| *Sim4* | 4 | $p_1 = 70, p_2 = 60, p_3 = 50, p_4 = 40$ |
| *Sim5* | 10 | each 50 |
| *Sim6* | 10 | each 50 |
| Simulations | Samples | Factors |
| *Sim1* | $n = \{20, 30, 40, 50\}$ | $k = 6$ all sparse |
| *Sim2* | $n = \{20, 30, 40, 50\}$ | $k = 8$ sparse and dense |
| *Sim3* | $n = \{20, 30, 40, 50\}$ | $k = 6$ all sparse |
| *Sim4* | $n = \{20, 30, 40, 50\}$ | $k = 8$ sparse and dense |
| *Sim5* | $n = \{20, 30, 40, 50\}$ | $k = 8$ all sparse |
| *Sim6* | $n = \{20, 30, 40, 50\}$ | $k = 10$ sparse and dense |

### *2.5.1 Simulating data*

*Paired views* We perform two simulations in the context of two paired views with $p_1 = 100$, $p_2 = 120$. The number of samples in these simulations is $n = \{20, 30, 40, 50\}$. The number of samples is chosen to be smaller than both $p_1$ and $p_2$ to reflect the large $p$ small $n$ problem that motivates our structured approach. In *Sim1* we simulate data with only sparse latent factors. We set $k = 6$, where two sparse factors are shared by both views (factor 1 and 2; Table 2.3), two sparse factors are specific to $\boldsymbol{y}^{(1)}$ (factor 3 and 4; Table 2.3), and two sparse factors are specific to $\boldsymbol{y}^{(2)}$ (factor 5 and 6; Table 2.3). The elements in the sparse loading matrix are randomly generated from a $\mathrm{N}(0, 4)$ Gaussian distribution, and sparsity is induced by setting 90% of the elements in each loading to zero at random. We make sure the absolute values of sparse loadings are greater than 0.5. Latent factors $\boldsymbol{x}_i$ are generated from $\mathrm{N}(\mathbf{0}, \boldsymbol{I})$. Residual errors are generated by first generating the $p = p_1 + p_2$ diagonal entries of the residual covariance matrix $\boldsymbol{\Sigma}$ from a uniform distribution on $(0.5, 1.5)$, and then generating each column of the error matrix from $\mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

In *Sim2* we include both sparse and dense latent factors. We extend *Sim1* to

Table 2.3: Configurations of sparse (S) and dense (D) factors in *Sim1* and *Sim2* with two views

| | *Sim1* | | | | | | *Sim2* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Factors | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $\boldsymbol{Y}^{(1)}$ | S | S | S | S | - | - | S | D | S | S | D | - | - | - |
| $\boldsymbol{Y}^{(2)}$ | S | S | - | - | S | S | S | D | - | - | - | S | S | D |

Table 2.4: Configurations of sparse (S) and dense (D) factors in *Sim3* and *Sim4* with four views

| | *Sim3* | | | | | | *Sim4* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Factors | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $\boldsymbol{Y}^{(1)}$ | S | - | - | S | - | - | S | - | - | - | D | - | - | - |
| $\boldsymbol{Y}^{(2)}$ | - | S | - | S | S | S | - | S | - | S | - | D | - | - |
| $\boldsymbol{Y}^{(3)}$ | - | - | S | - | S | S | - | - | S | S | - | - | D | - |
| $\boldsymbol{Y}^{(4)}$ | - | - | - | - | - | S | - | - | S | - | - | - | - | D |

$k = 8$ latent factors, where one of the shared sparse factors is now dense, and two dense factors, each specific to one view, are added. For all dense factors, each loading is generated according to a $\mathrm{N}(0, 4)$ Gaussian distribution (Table 2.3).

*Four views* We perform two additional simulations having four views with $p_1 = 70$, $p_2 = 60$, $p_3 = 50$ and $p_4 = 40$. The number of samples is set to $n = \{20, 30, 40, 50\}$. In *Sim3*, we let $k = 6$ and only simulate sparse factors. The first three factors are specific to $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$ and $\boldsymbol{y}^{(3)}$ respectively, and last three correspond to different subsets of the views (Table 2.4). In *Sim4* we let $k = 8$ and include both sparse and dense factors (Table 2.4). Samples in these two simulations are generated following the same method as in *Sim1* and *Sim2*.

*Ten views* To further evaluate BASS on multiple views, we perform two additional simulations on ten couple data sets with $p_v = 50$ for $v = 1, \ldots, 10$. The number of samples is also set to $n = \{20, 30, 40, 50\}$. In *Sim5*, we let $k = 8$ and only simulate sparse factors (Table 2.5). In *Sim6* we let $k = 10$ and simulate both sparse and dense factors (Table 2.5). Samples in these two simulations are generated following

Table 2.5: Configurations of sparse (S) and dense (D) factors in *Sim5* and *Sim6* with ten views

| | *Sim5* | | | | | | | | *Sim6* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Factors | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $\boldsymbol{Y}^{(1)}$ | S | - | - | - | - | - | - | - | S | - | - | - | - | - | D | - | - | - |
| $\boldsymbol{Y}^{(2)}$ | S | - | - | S | - | - | - | - | S | - | - | S | - | - | D | - | - | - |
| $\boldsymbol{Y}^{(3)}$ | S | - | - | S | S | - | - | - | - | - | - | S | - | - | D | D | - | - |
| $\boldsymbol{Y}^{(4)}$ | S | S | - | S | S | - | S | - | - | S | - | S | - | - | D | D | - | - |
| $\boldsymbol{Y}^{(5)}$ | - | S | - | S | S | - | S | - | - | S | - | S | S | - | - | D | D | - |
| $\boldsymbol{Y}^{(6)}$ | - | S | - | - | - | - | S | S | - | S | - | - | S | - | - | D | D | - |
| $\boldsymbol{Y}^{(7)}$ | - | - | S | - | - | - | S | S | - | S | S | - | S | - | - | - | D | D |
| $\boldsymbol{Y}^{(8)}$ | - | - | S | - | - | - | S | S | - | - | S | - | S | - | - | - | D | D |
| $\boldsymbol{Y}^{(9)}$ | - | - | S | - | - | - | - | S | - | - | S | - | - | - | - | - | - | D |
| $\boldsymbol{Y}^{(10)}$ | - | - | S | - | - | S | - | - | - | - | S | - | - | S | - | - | - | D |

the same method as before.

### *2.5.2 Models for comparison*

We compare BASS with five available linear models accepting multiple views: the Bayesian group factor analysis model with an ARD prior (GFA) (Klami et al., 2013), an extension of GFA by allowing element-wise sparsity with independent ARD priors (sGFA) (Khan et al., 2014; Suvitaival et al., 2014), a regularized version of CCA (RCCA) (González et al., 2008), sparse CCA (SCCA) (Witten and Tibshirani, 2009) and the Bayesian joint factor analysis model studied by Ray et al. (2014) (JFA). We further include a flexible non-linear model, manifold relevance determination (MRD) model (Damianou et al., 2012), in our comparisons. To further evaluate sensitivity of BASS on starting values we study three different initialization methods: random starting points, a small number of MCMC runs (50 iterations) and a small number of PX-EM runs (20 iterations).

The GFA model studied by Klami et al. (2013) puts an ARD prior on each column of the loading matrices, encouraging column-wise shrinkage of the loading matrix but not sparsity within these loadings. The computation complexity of GFA model with variational update requires $O(k^3m + k^2p + kpn)$ computation in updating loading

matrices and their covariances. Updating factors is in the same order as BASS. Updating ARD parameters is in lower order. Therefore GFA has $O(k^3m + k^2p + k^2n + kpn)$ per iteration. In our simulations, we run the GFA model with the factor number set to the correct values.

The sGFA model proposed by Khan et al. (2014) allows element-wise sparsity using independent ARD priors on loading elements. Loading columns are modeled by a spike and slab type mixture to allow column-wise sparsity. Inference is performed with a Gibbs sampler without using block update. Its complexity is in $O(k^3 + k^2pn)$ per iteration. We run the sGFA model with correct factor numbers in our six simulations.

We run the regularized version of classical CCA (RCCA) for comparison in *Sim1* and *Sim2* (González et al., 2008). Classical CCA aims to find $k$ canonical projection directions $\boldsymbol{u}_h$ and $\boldsymbol{v}_h$ $(h = 1, \ldots, k)$ for $\boldsymbol{Y}^{(1)}$ and $\boldsymbol{Y}^{(2)}$ respectively such that i) the correlation between $\boldsymbol{u}_h^\top \boldsymbol{Y}^{(1)}$ and $\boldsymbol{v}_h^\top \boldsymbol{Y}^{(2)}$ is maximized for $h = 1, \ldots, k$; and ii) $\boldsymbol{u}_{h'}^\top \boldsymbol{Y}^{(1)}$ is uncorrelated to $\boldsymbol{u}_h^\top \boldsymbol{Y}^{(1)}$ with $h' \neq h$, and similarly for $\boldsymbol{v}_h$ and $\boldsymbol{Y}^{(2)}$. Let these two projection matrices be denoted $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) \in \mathbb{R}^{p_1 \times k}$ and $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k) \in \mathbb{R}^{p_2 \times k}$. These matrices are the maximum likelihood estimates of the shared loading matrices in a probabilistic CCA model up to linear transformations (Bach and Jordan, 2005). However, classical CCA requires the observation covariance matrices to be non-singular and thus is not applicable in the current simulations. Therefore, we use a regularized version of CCA (RCCA) (González et al., 2008) by adding $\lambda_1 \boldsymbol{I}_{p_1}$ and $\lambda_2 \boldsymbol{I}_{p_2}$ to the two sample covariance matrices. The two regularization parameters $\lambda_1$ and $\lambda_2$ are chosen according to leave-one-out cross-validation with the search space defined on a $11 \times 11$ grid from 0.0001 to 0.01. The projection directions $\boldsymbol{U}$ and $\boldsymbol{V}$ are estimated using the best regularization parameters. We let $\boldsymbol{\Lambda}' = (\boldsymbol{U}; \boldsymbol{V})$; this matrix is comparable to the simulated loading matrix up to orthogonal transformations. We calculate the matrix $\boldsymbol{P}$ such that the Frobenius norm

between $\boldsymbol{\Lambda}'\boldsymbol{P}^\top$ and simulated $\boldsymbol{\Lambda}$ is minimized, with the constraint that $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}$. This is done by the constraint preserving updates of the objective function (Wen and Yin, 2013). After finding the optimal orthogonal transformation matrix, we recover $\boldsymbol{\Lambda}'\boldsymbol{P}^\top$ as the estimated loading matrix. We choose 6 and 8 regularized projections for comparison in *Sim1* and *Sim2* respectively, representing the true number of latent linear factors. RCCA does not apply to multiple coupled views, therefore is not included in other simulations.

The sparse CCA (SCCA) method (Witten and Tibshirani, 2009) maximizes correlation between two views after projecting the original space with $\ell_1$ penalties on the projection directions, producing sparse matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. This method is encoded in the R package `PMA` (Witten et al., 2013). As with RCCA, we find an optimal orthogonal transformation matrix $\boldsymbol{P}$ such that the Frobenius norm between $\boldsymbol{\Lambda}'\boldsymbol{P}^\top$ and simulated $\boldsymbol{\Lambda}$ was minimized, where $\boldsymbol{\Lambda}'$ is the vertical concatenation of the recovered sparse $\boldsymbol{U}$ and $\boldsymbol{V}$. We choose 6 and 8 sparse projections in *Sim1* and *Sim2* for comparison respectively. An extension of SCCA allows for multiple views (Witten and Tibshirani, 2009). For *Sim3* and *Sim4*, we recover four sparse projection matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}, \boldsymbol{U}^{(4)}$ and for *Sim5* and *Sim6*, we recover ten projection matrices. $\boldsymbol{\Lambda}'$ is calculated with the concatenation of those projection matrices. Then the orthogonal transformation matrix $\boldsymbol{P}$ is calculated similarly by minimizing the Frobenius norm between $\boldsymbol{\Lambda}'\boldsymbol{P}^\top$ and the true loading matrix $\boldsymbol{\Lambda}$. The number of canonical projections is set to 6 in *Sim3*, 8 in *Sim4* and *Sim5* and 10 in *Sim6*.

The Bayesian joint factor analysis model (JFA) studied by Ray et al. (2014) puts Indian buffet process prior (Griffiths and Ghahramani, 2011) on the factor indicators and inverse gamma prior on both loadings and factor values. Therefore the sparsity structure is assigned on factors instead of loading matrices. In addition, JFA only partitions latent factors to view specific ones and the one shared by all views. Its complexity is in $O(k^3 + k^2pn)$ per iteration with Gibbs sampler. We run JFA model

on our simulations with factor numbers set to the correct ones.

The non-linear manifold relevance determination (MRD) model (Damianou et al., 2012) extends Gaussian process latent variable (GPLVM) model (Lawrence, 2005) to include multiple views. A GPLVM puts a Gaussian process prior on latent variable space. It has a dual probabilistic PCA interpretation with loading columns marginalized out using a Gaussian prior. MRD extends GPLVM by putting multiple weight vectors on latent variables through a Gaussian process kernel. Each of the weight vectors corresponds to an view, therefore they determine a soft partition of latent variable space. Its complexity is in cubic in number of samples. This complexity is further reduced to quadratic using a sparse Gaussian process prior. Posterior inference and prediction using the MRD model is performed with Matlab package `vargplvm` (Damianou et al., 2012). We use the linear kernel with feature selection (i.e., `Linard2` kernel). We run the MRD model on our simulated data with the correct number of factors.

### *2.5.3 Methods of comparison*

We compare the loading matrices estimated by BASS with those generated from alternative methods. We use the two stability indices proposed by Gao et al. (2013) to make the comparison. The sparse stability index (SSI) measures the similarity between sparse loadings. SSI is invariant to column scale and factor switching, but it penalizes factor splitting and matrix rotation. Larger values of the SSI indicate better recovery. Let $\boldsymbol{C} \in \mathbb{R}^{k_1 \times k_2}$ be the absolute correlation matrix of columns of two sparse loading matrices. Then SSI can be calculated by (2.22). The dense stability index (DSI) quantifies the difference between dense loadings. It is invariant to orthogonal matrix rotation, factor switching, and scale changes. Let $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$

be the dense loading matrices. The DSI can be calculated using (2.23).

$$\text{SSI} = \frac{1}{2k_1}\sum_{h_1=1}^{k_1}\left(\max(\boldsymbol{c}_{h_1,\cdot}) - \frac{\sum_{h_2=1}^{k_2} I(c_{h_1,h_2} > \overline{\boldsymbol{c}_{h_1,\cdot}})c_{h_1,h_2}}{k_2-1}\right)$$

$$+ \frac{1}{2k_2}\sum_{h_2=1}^{k_2}\left(\max(\boldsymbol{c}_{\cdot,h_2}) - \frac{\sum_{h_1=1}^{k_1} I(c_{h_1,h_2} > \overline{\boldsymbol{c}_{\cdot,h_2}})c_{h_1,h_2}}{k_1-1}\right), \tag{2.22}$$

$$\text{DSI} = \frac{1}{p^2}\text{tr}(\boldsymbol{M}_1\boldsymbol{M}_1^\top - \boldsymbol{M}_2\boldsymbol{M}_2^\top). \tag{2.23}$$

In *Sim1*, *Sim3* and *Sim5*, all factors are regarded as sparse, and SSI's are calculated between true *combined* loading matrices and combined recovered loading matrices. In *Sim2*, *Sim4* and *Sim6*, because none of the compared methods explicitly distinguishes sparse and dense factors, we categorize them as follows. We first select a global sparsity threshold on the elements of the combined loading matrix. Here we set that value to 0.15. Elements below this threshold are set to zero in the loading matrix. Then we choose the first $q$ loading columns with the fewest non-zero elements as the sparse loadings, where $q$ equals to the number of sparse loadings in the true loading matrices. The remaining loading columns are considered dense loadings. We find that varying the sparsity threshold does not affect the separation of sparse and dense loadings significantly for those compared models. SSI's are then calculated for the true combined sparse loading matrix and the combined recovered sparse loadings. To calculate DSI, we treat the loading matrices $\boldsymbol{\Lambda}^{(v)}$ for each view separately, and calculate the DSI for the recovered dense components of each view. The final DSI for each method is the sum of the $m$ separate DSI's. Due to the fact that MRD does not provide estimations of loading matrix, we exclude MRD model in this comparison.

We further evaluate the prediction performance of BASS and other methods. According to (2.8), the joint distribution of any $\boldsymbol{y}_i^{(v)}$ and the rests $\boldsymbol{y}_i^{(-v)}$ can be

written as

$$\begin{pmatrix} \boldsymbol{y}_i^{(v)} \\ \boldsymbol{y}_i^{(-v)} \end{pmatrix} \sim \mathrm{N}\left( \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Lambda}^{(v)}(\boldsymbol{\Lambda}^{(v)})^\top + \boldsymbol{\Sigma}^{(v)} & \boldsymbol{\Lambda}^{(v)}(\boldsymbol{\Lambda}^{(-v)})^\top \\ \boldsymbol{\Lambda}^{(-v)}(\boldsymbol{\Lambda}^{(v)})^\top & \boldsymbol{\Lambda}^{(-v)}(\boldsymbol{\Lambda}^{(-v)})^\top + \boldsymbol{\Sigma}^{(-v)} \end{pmatrix} \right),$$

where $\boldsymbol{\Lambda}^{(-v)}$ and $\boldsymbol{\Sigma}^{(-v)}$ are the loading matrix and error covariance excluding the $v$th view. Therefore the conditional distribution of $\boldsymbol{y}_i^{(v)}$ is a multivariate response in a multiple linear regression model treating $\boldsymbol{y}_i^{(-v)}$ as predictors, with mean of

$$\begin{aligned} \mathbb{E}(\boldsymbol{y}_i^{(v)}|\boldsymbol{y}_i^{(-v)}) &= \boldsymbol{\Lambda}^{(v)}(\boldsymbol{\Lambda}^{(-v)})^\top \left(\boldsymbol{\Lambda}^{(-v)}(\boldsymbol{\Lambda}^{(-v)})^\top + \boldsymbol{\Sigma}^{(-v)}\right)^{-1} \boldsymbol{y}_i^{(-v)} \\ &= \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}^{(v)} (\boldsymbol{\lambda}_{\cdot h}^{(-v)})^\top \left(\boldsymbol{\Lambda}^{(-v)}(\boldsymbol{\Lambda}^{(-v)})^\top + \boldsymbol{\Sigma}^{(-v)}\right)^{-1} \boldsymbol{y}_i^{(-v)}. \end{aligned} \tag{2.24}$$

We use this property to predict certain views given others. For the six simulations, we generate $n = \{10, 30, 50, 100, 200\}$ as training data. In addition we generate test data using true model parameters. The number of test samples is set to 200. For each simulation study, we choose one view in the test data as response and use other views and model parameters estimated by training data to perform prediction. Mean squared error (MSE) is used to evaluate the prediction performance. For *Sim1* and *Sim2*, $\boldsymbol{y}_i^{(2)}$ is used as response; for *Sim3* and *Sim4*, $\boldsymbol{y}_i^{(3)}$ is used as response; and for *Sim5* and *Sim6*, $\boldsymbol{y}_i^{(8)}$, $\boldsymbol{y}_i^{(9)}$ and $\boldsymbol{y}_i^{(10)}$ are used as responses. The JFA model uses sparsity inducing prior instead of an independent Gaussian on latent factors, therefore we exclude JFA model in prediction.

### *2.5.4 Simulation results*

We first evaluate the performance of BASS in terms of recovering the correct number of sparse and dense factors in the six simulations. We perform 20 repeats for each initialization of BASS: random initialization (EM), 50 MCMC runs (MCMC-EM) and 20 parameter expanded EM runs (PX-EM). In *Sim1* and *Sim3*, we set the starting number of factors to 10. In *Sim2*, *Sim4*, *Sim5* and *Sim6*, we set the starting

Table 2.6: Percentage of latent factors correctly estimated across 20 runs with $n = 40$.

| | EM | MCMC-EM | PX-EM |
|---|---|---|---|
| *Sim1* | 79.17% | 99.17% | 91.67% |
| *Sim2* | 61.25% | 93.75% | 85.62% |
| *Sim3* | 50.00% | 78.57% | 73.57% |
| *Sim4* | 62.78% | 86.11% | 82.78% |
| *Sim5* | 17.22% | 86.67% | 66.67% |
| *Sim6* | 13.64% | 60.45% | 62.73% |

factor number to 15. We calculate the percentage of correctly identified factors across the 20 runs in the simulations with $n = 40$ (Table 2.6). MCMC-EM has the most accurate results, followed by PX-EM and then EM. With the increase of the number of views, the accuracy of all methods starts to deteriorate.

We run the other methods on the six simulations and compare the estimated loading matrices. BASS recovers the closest matches to the simulated loading matrices across the compared methods from a visual inspection (Figures 2.3, 2.4 and 2.5). The correctly estimated loading matrices by three different initializations of BASS produce similar results. We only plot matrices from one method.

*Two views* We then quantitatively compare the results. With two views (*Sim1* and *Sim2*), our model produce the best SSI's and DSI's among the compared models across different sample sizes (Figure 2.6). The column-wise sparsity induced by spike-slab type prior in sGFA produces nice loading selection with zero columns (Figure 2.3). However, its performance is limited in sparse loadings because the ARD prior does not produce sufficient element-wise sparsity. Therefore it produces relative low SSI's (Figure 2.6). As a consequence of not matching sparse loadings well, sGFA has difficulty recovering dense loadings, especially with small sample sizes (Figure 2.6). The GFA model suffers from recovering sparse loadings due to the ARD prior assigned on the entire column (Figure 2.3, Figure 2.6). Its dense loadings are

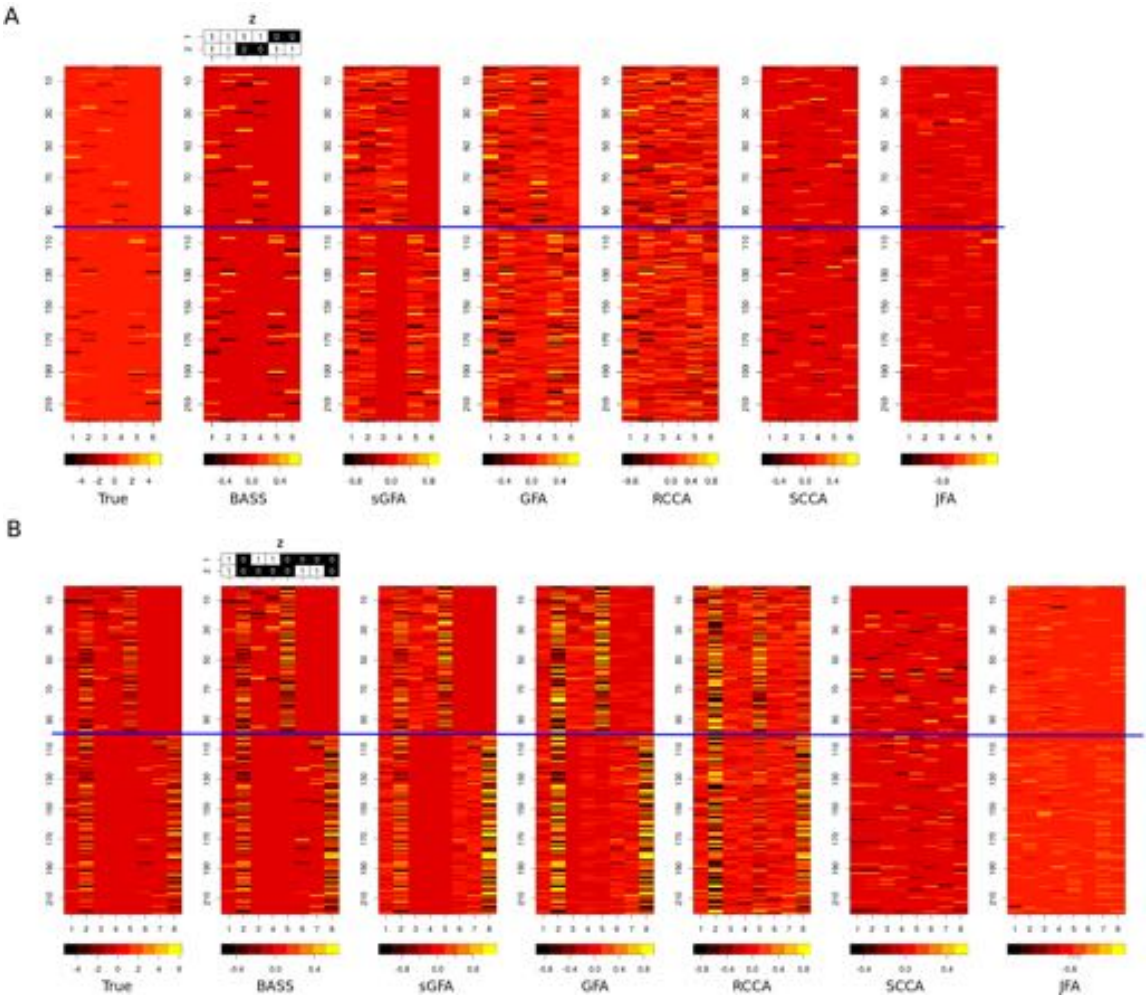


FIGURE 2.3: Estimated loading matrices for two paired views with $n = 40$ for different methods. The columns of estimated loadings are reordered and flipped sign when necessary for visual convenience. Horizontal lines separate two views. Panel A: Results in *Sim1*. Panel B: Results in *Sim2*.

also influenced with small sample sizes (Figure 2.6). RCCA also suffers in the two simulations because the recovered loadings are not sufficiently sparse (Figure 2.3). SCCA recovers shared sparse loadings well in *Sim1* (Figure 2.3). However SCCA does not model local covariance structure, and therefore is unable to recover the sparse loadings specific to either of views in *Sim1* (Figures 2.3A), resulting again in poor SSI's (Figure 2.6). Adding dense loadings makes it worse (Figure 2.3B, 2.6). JFA model does not recover the true loading well due to the sparsity is assigned on

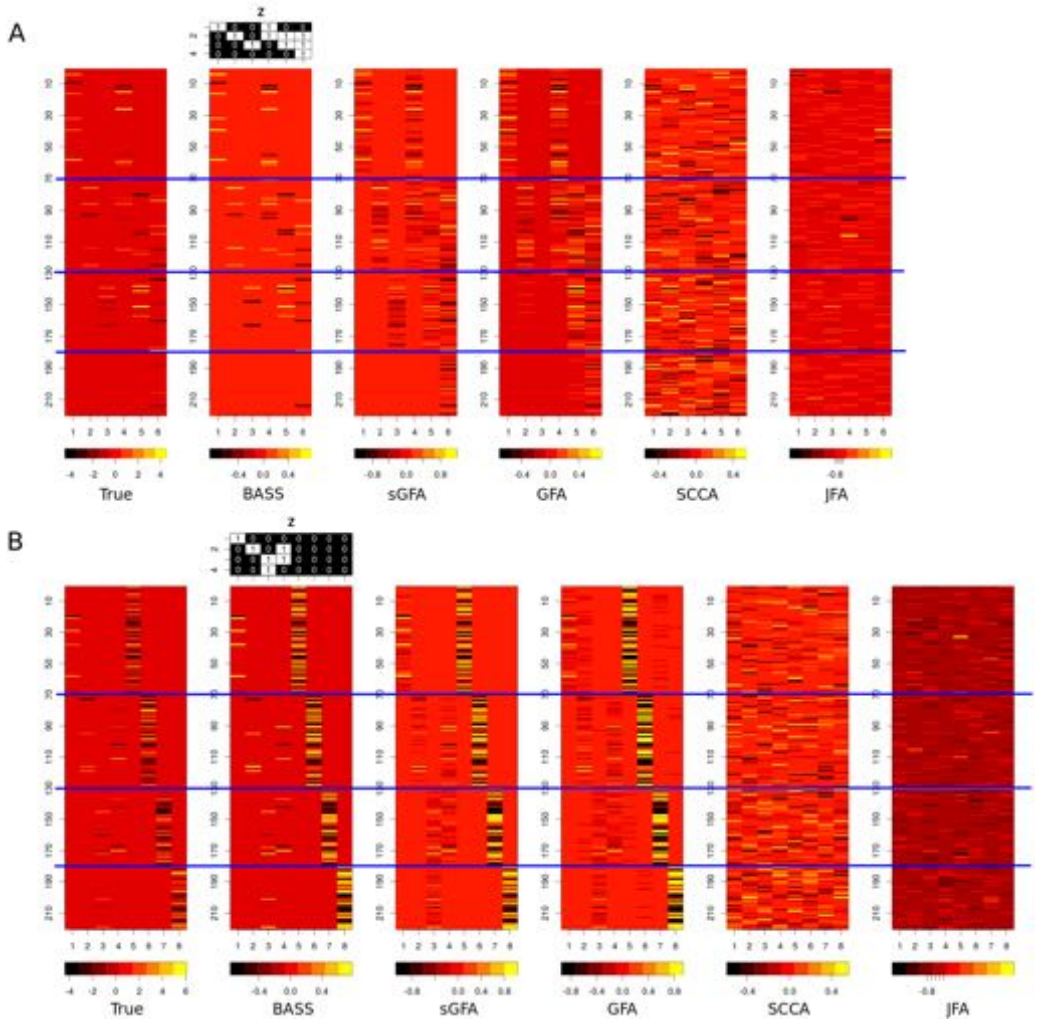


FIGURE 2.4: Estimated loading matrices for four coupled views with $n = 40$ for different methods. The columns are re-arranged in the similar manner as in Figure 2.3. Panel A: Results in *Sim3*. Panel B: Results in *Sim4*.

factors instead of loadings (Figure 2.3). Its SSI's and DSI's also greatly deteriorate (Figure 2.6).

We next evaluate their prediction performance with two views. In *Sim1*, SCCA achieves the best prediction accuracy in three training sample sizes (Table 2.7). This can be attributed to the nice performance of SCCA in identifying shared sparse loadings (Figure 2.3), and the prediction accuracy comes only through shared loadings. Note from (2.24) that zero columns in either $\mathbf{\Lambda}^{(v)}$ or $\mathbf{\Lambda}^{(-v)}$ decouple the contribution

of the factors to the dependency between $\boldsymbol{y}_i^{(v)}$ and $\boldsymbol{y}_i^{(-v)}$. In *Sim2*, both shared sparse and dense factors contribute to the prediction performance. In this setting BASS achieves the best prediction accuracy (Table 2.7).

*Four views* For simulations with four views (*Sim3* and *Sim4*), BASS still can correctly identify sparse and dense property of factors and their active views (Figure 2.4). sGFA still achieves column-wise sparsity well as in two views, however its sparsity level within factors is not as good as BASS. GFA suffers from column shrinkage: columns with zero values are not effectively shrunk to zero (Figure 2.4B). Its element-wise shrinkage is also not as effective as BASS or sGFA (Figure 2.4). The results of SCCA and JFA do not match true loading matrices (Figure 2.4). The results of stability indices show that BASS still produce the best SSI's and DSI's among the compared models in almost all different sample sizes (Figure 2.7). sGFA achieves similar SSI values in *Sim3* with $n = 40$ compared to BASS with random initialization (EM), but still inferior compared to MCMC-EM and PX-EM. The advantage of BASS in other cases is very clear (Figure 2.7). BASS also achieves the best prediction performance with $\boldsymbol{y}_i^{(3)}$ as response and the rest views as predictors (Table 2.8).

*Ten views* When we increase the view number to ten (*Sim5* and *Sim6*), BASS still can correctly identify the sparse and dense properties of factors and their active views (Figure 2.5). The performance of sGFA in column selection remains effective, well with an inferior element-wise shrinkage compared to BASS (Figure 2.5). GFA suffers greatly from both column-wise and element-wise sparsity (Figure 2.5). SCCA and JFA do not produce results that match true loading matrices (Figure 2.5). For stability indices, BASS with MCMC-EM and PX-EM produce the best SSI's in *Sim5* among the compared models in almost all different sample sizes (Figure 2.7). sGFA

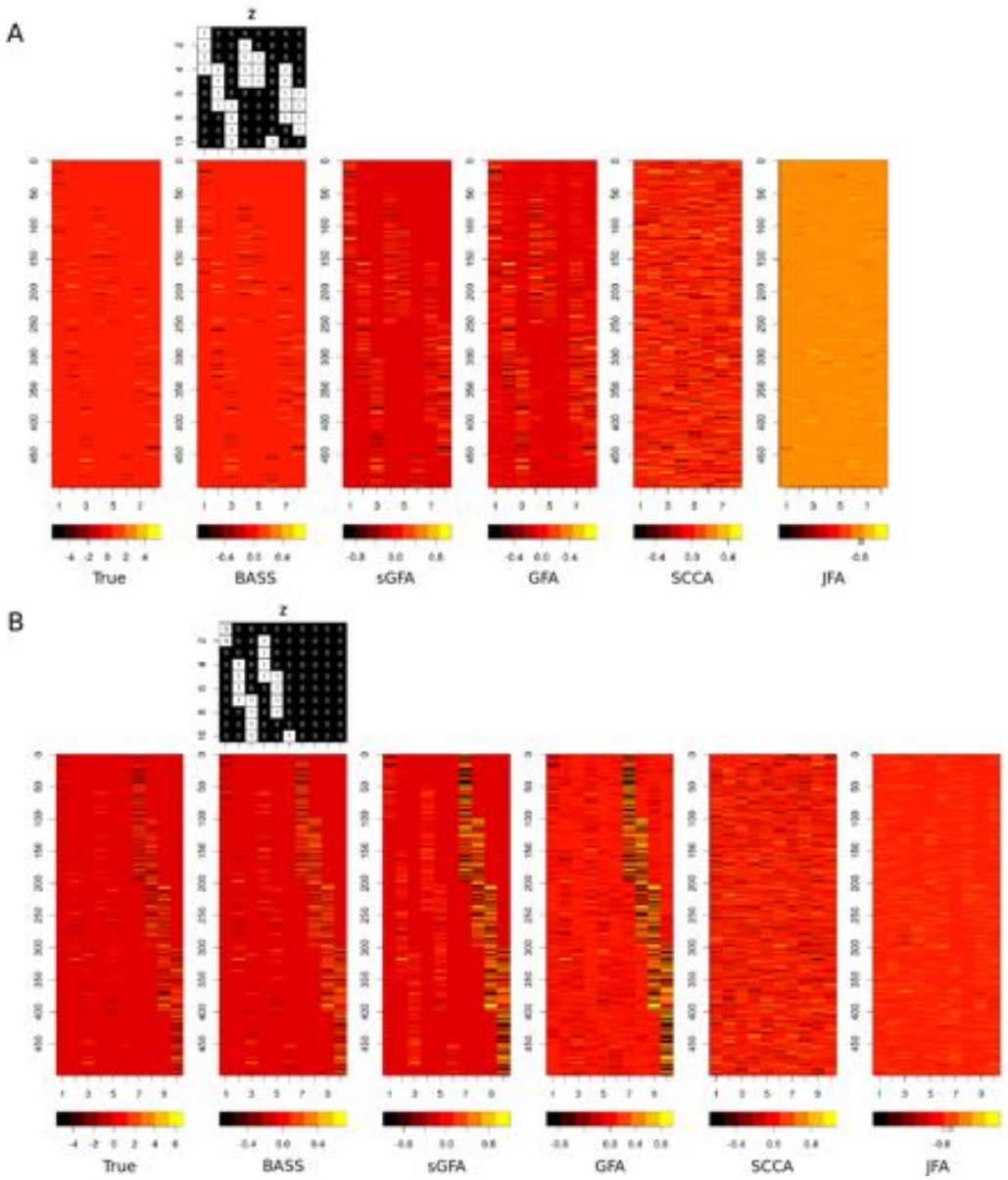


FIGURE 2.5: Estimated loading matrices for ten coupled views with $n = 40$ for different methods. The columns are re-arranged in the similar manner as in Figure 2.3. Panel A: Results in *Sim5*. Panel B: Results in *Sim6*.

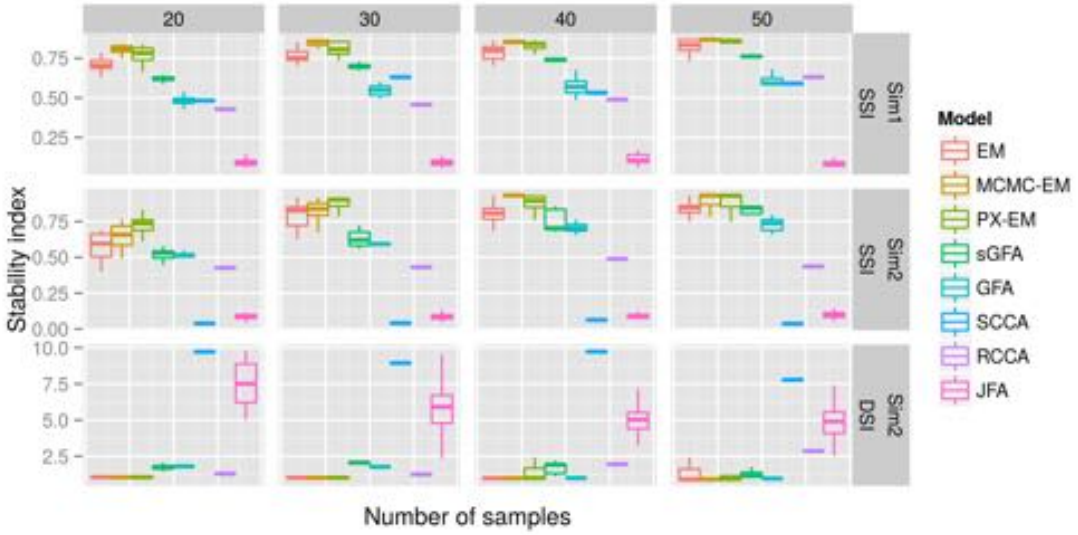


FIGURE 2.6: Comparison of stability indices on estimated loading matrices with two views. For SSI, a larger value indicates the estimated sparse loadings are closer to true sparse loadings. For DSI, a smaller value indicates estimated dense loadings are closer to the true dense loadings. The boundaries of the box are the first and third quartiles. The line extends to the highest/lowest value that is within 1.5 times the distance between the first and third quartiles of the box boundaries.

achieves better SSI's than BASS with random initialization (EM). GFA has a better SSI's than EM only with $n = 40$. The advantage of BASS over other models is clear (Figures 2.7). In *Sim6*, BASS has SSI's and DSI's at least as good as the best of other compared models (Figure 2.8). BASS also achieves the best prediction performance in *Sim5*. However GFA has lowest MSE's in *Sim6* with $n = 20$ and $n = 40$, although its loading matrices do not produce sufficient column-wise and element-wise sparsity (Figure 2.5).

## 2.6 Applications

In this section we consider three different applications of BASS. In the first application we evaluate the prediction performance with multivariate correlated response variables in the Mulan library (Tsoumakas et al., 2011). In the second application

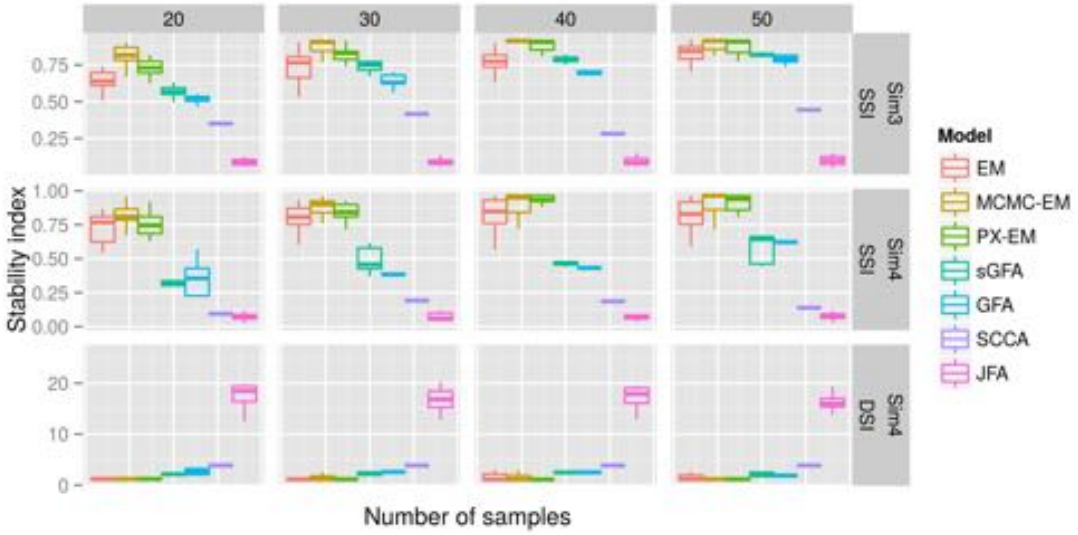


FIGURE 2.7: Comparison of stability indices on estimated loading matrices with four views. For SSI, a larger value indicates the estimated sparse loadings are closer to true sparse loadings. For DSI, a smaller value indicates estimated dense loadings are closer to the true dense loadings. Boxes have the same meaning as in Figure 2.6.

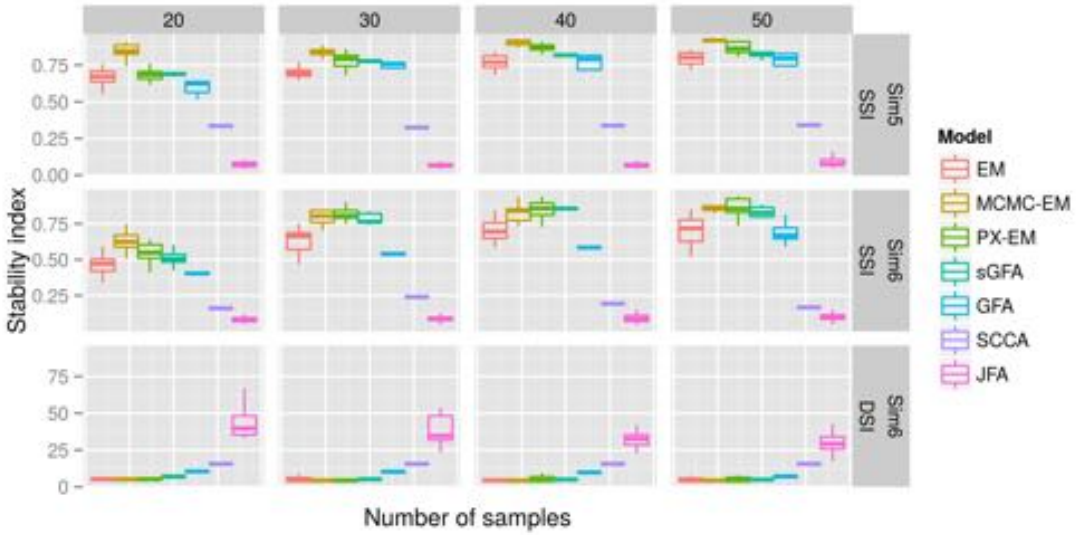


FIGURE 2.8: Comparison of stability indices on estimated loading matrices with ten views. For SSI, a larger value indicates the estimated sparse loadings are closer to true sparse loadings. For DSI, a smaller value indicates estimated dense loadings are closer to the true dense loadings. Boxes have the same meaning as in Figure 2.6.

Table 2.7: Prediction accuracy with two views on $n_s = 200$ test samples. $\boldsymbol{y}_i^{(2)}$ in test samples is treated as response and $\boldsymbol{y}_i^{(1)}$ is used to predict the response using parameters learned from training sets. Prediction accuracy is measured by mean squared error (MSE) between simulated $\boldsymbol{y}_i^{(1)}$ and $\mathbb{E}(\boldsymbol{y}_i^{(1)}|\boldsymbol{y}_i^{(2)})$. Values presented are the mean MSE with standard deviation calculated from 20 repeats of different models. Model with smallest MSE is bolded. When multiple models have the smallest MSE the one with least standard deviation is bolded.

| | | BASS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $n_t$ | EM | MCMC-EM | PX-EM | sGFA | GFA | SCCA | RCCA | MRD-lin |
| *Sim1* | 10 | 1.00(0.024) | 1.03(0.024) | 1.02(0.028) | 1.00(<1e-3) | 0.98(0.002) | **0.88** | 1.01 | 1.08(0.024) |
| | 30 | 0.90(0.022) | 0.88(0.001) | 0.88(0.003) | 0.92(0.005) | 0.93(0.002) | **0.88** | 0.97 | 1.00(0.016) |
| | 50 | 0.88(0.011) | **0.87(0.003)** | 0.88(0.014) | 0.90(0.004) | 0.92(0.002) | 0.88 | 0.92 | 0.98(0.028) |
| | 100 | 0.88(0.010) | 0.87(0.001) | 0.87(0.005) | 0.89(0.003) | 0.89(<1e-3) | **0.87** | 0.91 | 0.97(0.016) |
| | 200 | 0.88(0.007) | 0.87(0.004) | 0.87(0.005) | 0.88(0.001) | 0.88(<1e-3) | **0.87** | 0.95 | 1.16(0.202) |
| *Sim2* | 10 | 0.80(0.161) | 0.82(0.162) | **0.68(0.003)** | 0.74(0.043) | 0.89(0.023) | 0.86 | 0.72 | 1.14(0.002) |
| | 30 | 0.72(0.092) | 0.72(0.097) | 0.67(0.016) | 0.67(0.014) | **0.66(0.006)** | 0.86 | 0.70 | 1.15(0.034) |
| | 50 | 0.71(0.155) | 0.70(0.155) | 0.65(0.105) | **0.63(0.009)** | 0.67(<1e-3) | 0.85 | 0.72 | 1.17(0.009) |
| | 100 | 0.63(0.066) | 0.61(0.013) | 0.62(0.013) | 0.62(0.005) | **0.61(0.001)** | 0.85 | 0.75 | 1.13(0.013) |
| | 200 | 0.65(0.099) | 0.61(0.012) | 0.63(0.020) | 0.62(0.007) | **0.61(0.002)** | 0.85 | 0.81 | 1.55(0.591) |

we apply BASS on gene expression data from the Cholesterol and Pharmacogenomic (CAP) study. The data consist of expression level measurements for about ten thousands genes in multiple lymphoblastoid cell lines (LCLs) under two conditions (Mangravite et al., 2013; Brown et al., 2013). BASS is used to detect the sparse covariance structures specific to each condition, and then to construct two

Table 2.8: Prediction accuracy with four views on $n_s = 200$ test samples. $\boldsymbol{y}_i^{(3)}$ in test samples is treated as response and $\boldsymbol{y}_i^{(1)}$, $\boldsymbol{y}_i^{(2)}$ and $\boldsymbol{y}_i^{(4)}$ are used to predict the response using parameters learned from training sets. Means of MSE and standard deviations are calculated and shown in a similar manner to the results shown in Table 2.7.

| | | BASS | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $n_t$ | EM | MCMC-EM | PX-EM | sGFA | GFA | SCCA | MRD-lin |
| *Sim3* | 10 | 1.03(0.044) | 1.02(0.019) | 1.01(0.010) | 1.00(<1e-3) | **0.97(0.001)** | 1.00 | 1.00(<1e-3) |
| | 30 | 0.91(0.049) | **0.87(0.016)** | 0.88(0.007) | 0.90(0.007) | 0.93(0.003) | 1.00 | 0.99(0.021) |
| | 50 | 0.85(0.019) | **0.85(<1e-3)** | 0.87(0.038) | 0.87(0.005) | 0.88(0.002) | 1.01 | 1.04(0.095) |
| | 100 | 0.85(0.019) | **0.84(0.002)** | 0.84(0.003) | 0.86(0.004) | 0.87(0.001) | 1.11 | 0.92(0.014) |
| | 200 | 0.84(0.001) | 0.84(<1e-3) | 0.84(0.004) | 0.84(0.001) | **0.83(0.001)** | 1.13 | 1.16(0.140) |
| *Sim4* | 10 | 1.05(0.095) | 1.03(0.094) | 1.10(0.138) | **1.00(<1e-3)** | 1.32(0.029) | 1.35 | 1.98(0.067) |
| | 30 | 0.97(0.020) | **0.95(0.015)** | 0.96(0.013) | 0.97(0.007) | 1.03(0.003) | 1.40 | 1.50(0.090) |
| | 50 | 0.94(0.013) | **0.93(0.005)** | 0.94(0.012) | 0.95(0.005) | 1.02(0.017) | 1.40 | 1.50(0.084) |
| | 100 | 0.93(0.015) | **0.93(0.007)** | 0.93(0.010) | 0.94(0.003) | 0.96(<1e-3) | 1.51 | 1.47(0.088) |
| | 200 | 0.91(0.029) | 0.92(0.022) | 0.89(0.047) | 0.93(0.001) | **0.89(0.001)** | 1.77 | 1.58(0.132) |

Table 2.9: Prediction mean squared error with ten views on $n_s = 200$ test samples. $\boldsymbol{y}_i^{(8)}, \boldsymbol{y}_i^{(9)}$ and $\boldsymbol{y}_i^{(10)}$ in test samples are treated as responses and the rests are used to predict the response using parameters learned from training sets. Means of MSE and standard deviations are calculated and shown in a similar manner to the results shown in Table 2.7.

| | | BASS | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $n_t$ | EM | MCMC-EM | PX-EM | sGFA | GFA | SCCA | MRD-lin |
| *Sim5* | 10 | 1.01(0.020) | 1.00(0.011) | 1.00(0.007) | 0.99(0.008) | 1.00(0.002) | **0.99** | 1.49(0.001) |
| | 30 | 0.88(0.031) | **0.86(0.018)** | 0.87(0.028) | 0.89(0.005) | 0.90(0.002) | 0.99 | 1.01(0.035) |
| | 50 | 0.86(0.023) | **0.85(<1e-3)** | 0.86(0.022) | 0.87(0.003) | 0.88(0.001) | 0.99 | 0.97(0.020) |
| | 100 | 0.85(0.007) | **0.85(<1e-3)** | 0.85(0.002) | 0.86(0.003) | 0.87(0.001) | 1.01 | 0.92(0.039) |
| | 200 | 0.85(0.006) | 0.84(<1e-3) | 0.84(<1e-3) | 0.84(0.001) | **0.83(0.001)** | 0.96 | 1.06(0.105) |
| *Sim6* | 10 | 0.61(0.164) | 0.57(0.116) | **0.51(0.031)** | 0.58(0.012) | 0.75(0.011) | 0.97 | 1.00(<1e-3) |
| | 30 | 0.49(0.160) | 0.40(0.093) | **0.38(0.007)** | 0.43(0.006) | 0.40(0.005) | 0.98 | 0.46(0.006) |
| | 50 | 0.44(0.099) | 0.39(0.011) | **0.39(0.004)** | 0.41(0.002) | 0.40(0.001) | 1.01 | 0.42(0.009) |
| | 100 | 0.39(0.033) | 0.39(0.004) | 0.39(0.011) | 0.39(0.002) | **0.39(0.001)** | 0.97 | 0.52(0.249) |
| | 200 | 0.38(0.003) | **0.38(0.001)** | 0.38(0.001) | 0.39(0.001) | 0.39(0.001) | 1.01 | 0.40(0.020) |

condition-specific co-expression networks. In the third application, we apply BASS on document data with approximately $20,000$ newsgroup documents divided into 20 newsgroups (Joachims, 1997).

### *2.6.1 Multivariate response prediction*

The Mulan library consists of multiple data sets with the aim of studying multi-label predictions (Tsoumakas et al., 2011). This library is used to test the Bayesian CCA model in multi-label prediction context with 0/1 labels (Klami et al., 2013). There are two views ($m = 2$), the labels are treated as one view ($\boldsymbol{Y}^{(1)}$) and the features are treated as another ($\boldsymbol{Y}^{(2)}$). Recently Mulan adds multi-target regression data sets with continuous target variables. We choose ten benchmark data sets from Mulan library. Four of them have 0/1 labels as responses, which are also studied in (Klami et al., 2013). Another six data sets consist of continuous responses (Table 2.10).

We run BASS, sGFA, GFA and MRD on these ten data sets. Prediction accuracy is used to compare the models. For 0/1 labels, we use the Hamming loss between the predicted labels and true labels to calculate the prediction error. The predicted labels on test samples are calculated using the same thresholding rules in (Klami

Table 2.10: Multivariate response prediction from Mulan library. First View is used as predictors and the second view is treated as response. $n_t$: the number of training samples. $n_s$: the number of test samples. The first view in the first four data sets are 0/1 responses, and the rest six are continuous responses. For 0/1 response, prediction accuracy is evaluated using Hamming loss between predicted labels and test labels in test samples. For continuous response, mean squared error (MSE) is used to evaluate prediction accuracy. Values presented are the minimum Hamming loss/MSE across 20 repeats of different models. Model with smallest MSE is bolded. When multiple models have the smallest MSE the one with least standard deviation is bolded.

| Data set | $p_1$ | $p_2$ | $n_t$ | $n_s$ | BASS | sGFA | GFA | MRD-lin |
|---|---|---|---|---|---|---|---|---|
| bibtex | 1836 | 159 | 4880 | 2515 | 0.014(0.001) | 0.014(0.001) | **0.014(<1e-3)** | 0.014(0.001) |
| delicious | 983 | 500 | 12920 | 3185 | 0.016(0.001) | **0.016(<1e-3)** | 0.017(<1e-3) | 0.020(<1e-3) |
| mediamill | 120 | 101 | 30993 | 12914 | **0.032(0.001)** | 0.032(0.005) | 0.034(<1e-3) | 0.043(<1e-3) |
| scene | 294 | 6 | 1211 | 1196 | 0.131(0.016) | **0.123(0.029)** | 0.130(0.002) | 0.138(0.026) |
| rf1 | 64 | 8 | 4108 | 5017 | **0.292(0.050)** | 0.390(0.008) | 0.309(<1e-3) | 0.370(0.146) |
| rf2 | 576 | 8 | 4108 | 5017 | **0.271(0.027)** | 0.478(0.004) | 0.427(0.001) | 0.438(0.160) |
| scm1d | 280 | 16 | 8145 | 1658 | **0.211(0.005)** | 0.225(0.028) | 0.213(<1e-3) | 0.212(0.163) |
| scm20d | 61 | 16 | 7463 | 1503 | 0.650(0.015) | **0.538(0.006)** | 0.720(0.002) | 0.608(0.033) |
| atp1d | 370 | 6 | 237 | 100 | **0.176(0.032)** | 0.208(0.006) | 0.201(0.001) | 0.219(0.113) |
| atp7d | 370 | 6 | 196 | 100 | 0.597(0.063) | 0.537(0.015) | **0.537(0.003)** | 0.545(0.049) |

et al., 2013). The value of the threshold is chosen so that the Hamming loss between estimated labels and true labels in training set is maximized. We use the R package `PresenceAbsence` and Matlab function `perfcurve` to find the best thresholds for those models. For continuous target variables, mean squared error (MSE) is used to evaluate prediction accuracy. We initialize BASS with 500 factors and 50 PX-EM initial iterations. The other models are set to the default parameters with factor numbers set to $\min(p_1, p_2, 50)$. The linear kernel with feature selection (Linard2 kernel) is used in MRD. All the models are repeated 20 times, and minimum errors are reported. The results are summarized in Table 2.10. BASS achieves the best prediction accuracy in five data sets. The averaged factors identified by BASS are shown in Table 2.11.

### *2.6.2 Gene expression data analysis*

We apply our model to gene expression data from the Cholesterol and Pharmacogenomic (CAP) study, consisting of expression level measurements for $10,195$ genes in

Table 2.11: Averaged estimated latent factors from the ten data sets in Mulan library. S represents a sparse vector; D represents a dense vector.

| $\boldsymbol{Y}^{(1)}$<br>$\boldsymbol{Y}^{(2)}$ | S<br>S | S<br>- | -<br>S | D<br>D | D<br>- | -<br>D | S<br>D | D<br>S | Total |
|---|---|---|---|---|---|---|---|---|---|
| bibtex | 355 | 55 | 2 | 0 | 0 | 0 | 0 | 0 | 413 |
| delicious | 489 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 499 |
| mediamill | 56 | 19 | 36 | 0 | 0 | 0 | 0 | 30 | 141 |
| scene | 27 | 20 | 0 | 32 | 4 | 0 | 35 | 15 | 133 |
| rf1 | 5 | 1 | 0 | 8 | 0 | 0 | 6 | 4 | 25 |
| rf2 | 43 | 93 | 1 | 39 | 5 | 0 | 57 | 25 | 263 |
| scm1d | 29 | 13 | 2 | 22 | 0 | 1 | 43 | 6 | 115 |
| scm20d | 11 | 4 | 2 | 13 | 0 | 0 | 20 | 1 | 53 |
| atp1d | 0 | 50 | 0 | 3 | 3 | 1 | 2 | 0 | 59 |
| atp7d | 0 | 46 | 0 | 1 | 1 | 1 | 2 | 0 | 51 |

480 lymphoblastoid cell lines (LCLs) after 24-hour exposure to either a control buffer ($\boldsymbol{Y}^{(1)}$) or 2 $\mu M$ simvastatin acid ($\boldsymbol{Y}^{(2)}$) (Mangravite et al., 2013; Brown et al., 2013). In this example, the number of views $m = 2$, representing gene expression levels on the same samples and genes after the two different exposures. The expression levels are preprocessed to adjust for experimental traits (batch effects and cell growth rate) and clinical traits of donors (age, BMI, smoking status and gender). We have projected the adjusted expression levels to the quantiles of a standard normal within gene. Then we apply BASS with the initial number of factors set to $k = 2,000$. We perform parameter estimation 100 times on these data with PX-EM initialization of 100 iterations. Across these 100 runs, the estimated number of recovered factors is approximately 870. Across the 100 runs, we only discover very few dense factors (Table 2.12). This potentially is due to the systematic variations have been adjusted by the preprocessing step. In addition the idiosyncratic errors explain the majority of total variance (85.27%).

We compute the proportion of variance explained (PVE) by those sparse factors (Figure 2.9A). The PVE for the $h$th factor is calculated as the variance explained

Table 2.12: Estimated latent factors in the CAP gene expression data with two views. S represents a sparse vector; D represents a dense vector. PVE: proportion of variance explained.

| | $\boldsymbol{Y}^{(1)}$<br>$\boldsymbol{Y}^{(2)}$ | S<br>S | S<br>- | -<br>S | D<br>D | D<br>- | -<br>D | rest | Total |
|---|---|---|---|---|---|---|---|---|---|
| PX-EM | #Factor | 731 | 63 | 62 | 1 | 0 | 0 | 12 | 870 |
| | PVE(%) | 11.18 | 0.88 | 0.82 | 0.17 | 0.20 | 0.10 | 1.98 | 15.31 |
| EM | #Factor | 23 | 175 | 200 | 5 | 26 | 28 | 16 | 473 |
| | PVE(%) | 0.35 | 0.42 | 0.61 | 3.63 | 31.97 | 38.91 | 14.80 | 90.67 |

by the $h$th factor divided by the total variance: $\text{tr}(\boldsymbol{\lambda}_{\cdot h}\boldsymbol{\lambda}_{\cdot h}^\top)/\text{tr}(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Sigma})$. Shared sparse factors explain more variance than specific sparse factors, which indicates the expression variations are mostly maintained between the two treatments. We also find that 87.48% of the specific sparse factors contain fewer than 100 genes, and 0.71% of those factors have greater than 500 genes. The shared sparse factors have more genes than those specific ones. 71.95% shared sparse factors have fewer than 100 genes, and 4.54% such factors have greater than 500 genes. (Figure 2.9B).

The sparse factors specific to each view characterize the local sparse covariance estimates. We use view specific sparse factors to a construct a gene co-expression network that is uniquely found in that condition. We call such a network the condition-specific network. The problem of constructions of condition specific co-expression networks have been both studied by machine learning and computational biology approaches (Li, 2002; Ma et al., 2011). Our BASS model provides an natural alternative way to solve this problem. We denote $\boldsymbol{B}_s^{(v)}$ as the sparse loadings in $\boldsymbol{B}^{(v)}$ ($v \in \{1, 2\}$). Then $\boldsymbol{\Omega}_s^{(v)} = \boldsymbol{B}_s^{(v)}(\boldsymbol{B}_s^{(v)})^\top + \boldsymbol{\Sigma}^{(v)}$ represents the regularized estimate of the covariance matrix specific to each view after controlling for the contributions of the dense factors. We invert this positive definite covariance matrix to get a precision matrix $\boldsymbol{R}^{(v)} = (\boldsymbol{\Omega}_s^{(v)})^{-1}$. The partial correlation between gene $j_1$ and $j_2$ is then calculated by normalizing the precision matrix (Edwards, 2000; Schäfer and Strimmer,

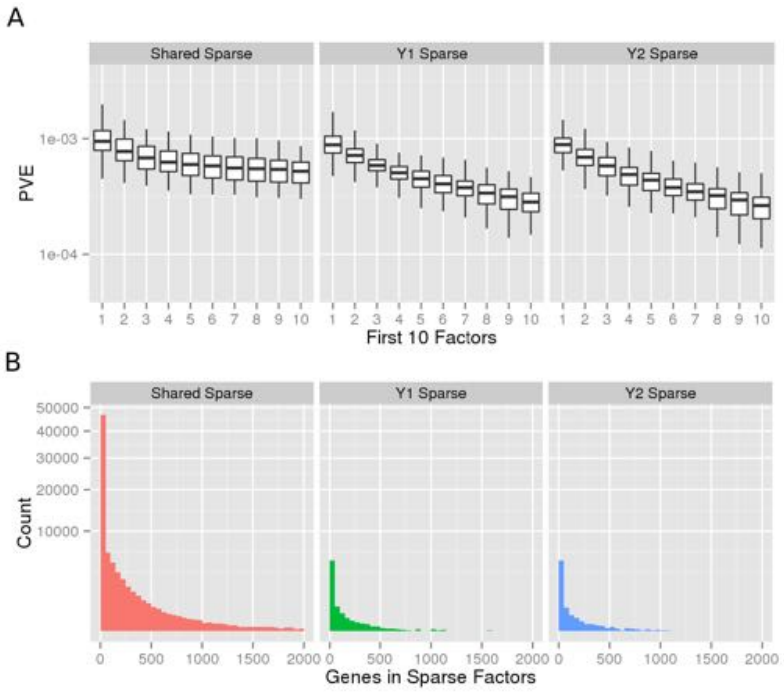


FIGURE 2.9: Results of applying BASS to the CAP gene expression data. $\boldsymbol{Y}^{(1)}$: the view from buffer-treated samples. $\boldsymbol{Y}^{(2)}$: the view from statin-treated samples. Panel A: the proportion of variance explained (PVE) by different factors. Factors are ordered by their PVE's and first 10 factors are displayed. PVE is on the $\log_{10}$ scale. Panel B: Histogram of the number of genes in different sparse factors. The count is displayed in square root scale.

2005):

$$\rho^{(v)}_{j_1 j_2} = -\frac{r^{(v)}_{j_1 j_2}}{\sqrt{r^{(v)}_{j_1 j_1} r^{(v)}_{j_2 j_2}}}.$$

A partial correlation that is zero for two genes suggests that they are conditionally independent (conditional on the remaining genes in the network). Connecting genes with non-zero partial correlation results a undirected network known as a Gaussian graphical model (Edwards, 2000; Koller and Friedman, 2009).

We use following method to combine the results of 100 runs to construct a single condition-specific gene co-expression network for each view. For each run, we first construct a network by connecting genes with partial correlation greater than a

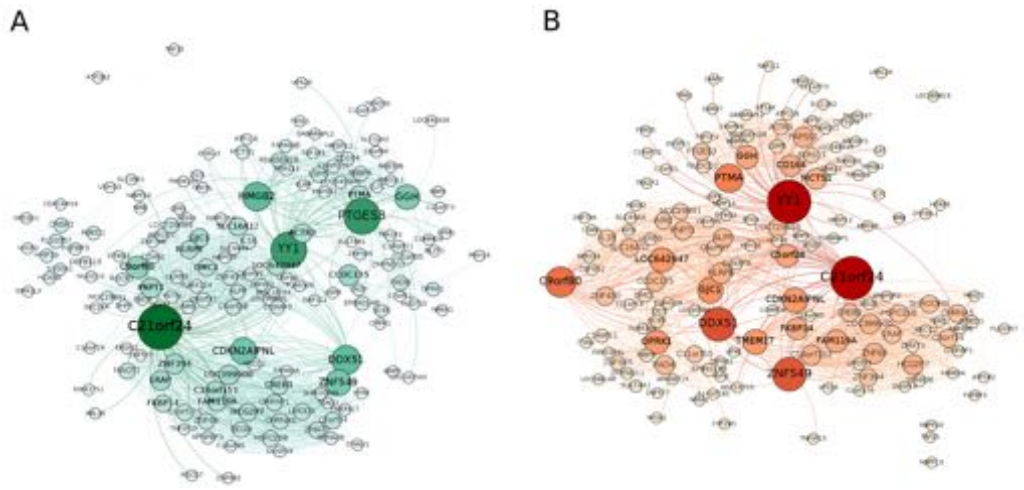


FIGURE 2.10: Estimated condition-specific gene co-expression networks from CAP data. Two networks are constructed to represent the condition-specific co-expression between buffer-treated samples (Panel A) and statin-treated samples (Panel B). The node and label size is scaled according to the number of shortest paths from all vertices to all others that pass through that node (betweenness centrality).

threshold (0.01). Then we combine the 100 networks to construct a single network by keeping the edges that appear in more than 50 (50%) networks. The final two condition-specific gene co-expression networks contains 160 genes and $1,244$ edges (buffer treated view, Figure 2.10A) 154 genes and $1,030$ edges (statin-treated view, Figure 2.10B) respectively.

### *2.6.3 Document data analysis*

In this application we consider the 20 newsgroup document data (Joachims, 1997). The documents have processed so that duplicates and headers are removed, resulting $18,846$ documents. The data are downloaded using `scikit-learn` Python package (Pedregosa et al., 2011). We convert the raw data into TF-IDF vectors and select 319 words using SVM feature selection in `scikit-learn`. One document has a zero vector across all the words therefore is deleted. We further select ten documents from each newsgroup as test data.

We apply BASS on the transposed data with 20 newsgroups as 20 views. We set $2,000$ initial factors and perform 100 parameter estimations, with 100 initial PX-EM iterations. There are approximately 825 factors recovered. We analyze the group specific words in following way. For each estimated loading, we calculate its Pearson correlations with group indicator vectors consisting of ones for a specific group and zeros for other groups. Then the loadings with the ten largest absolute correlation coefficients are considered and the words with the largest absolute factor scores corresponding to the ten loadings are listed. The results of one run are shown in Table A.1. For example, the `alt.atheism` newsgroup has 'islam', 'keith' and 'okcforum' as the top words, and the `rec.sport.baseball` newsgroup has 'baseball', 'braves' and 'runs' as top words. We further partition the newsgroups into six classes according to subject matter to analyze the shared words across different newsgroups (Table 2.13). Similarly we calculate the Pearson correlations with cross group 0/1 indicator vectors and analyze the top words in the ten factors with largest absolute correlation coefficients (Table 2.13). For example, the newsgroups of `talk.religion.misc`, `alt.atheism` and `soc.religion.christian` share 'god', 'bible' and 'christian' as top words. One of the selected shared loading for this newsgroup class is shown in Figure 2.11A.

Then we use the estimated loading and factors from training set to predict the document groups in the test set. The estimated loading matrix and factors give a regularized matrix approximation of training data matrix. To estimate the loadings in the test set, we left multiplied the test data matrix by the Moore-Penrose pseudoinverse of factors estimated from training data. This give a rough estimate of the loading matrix for test data. Then test labels are predicted using ten nearest neighbors based on the loading rows. For the 200 test documents, we generate an approximately 58.17% accuracy using Hamming loss. One predicted and true newsgroup labels are shown in Figure 2.11B. Due to some of the newsgroups are very

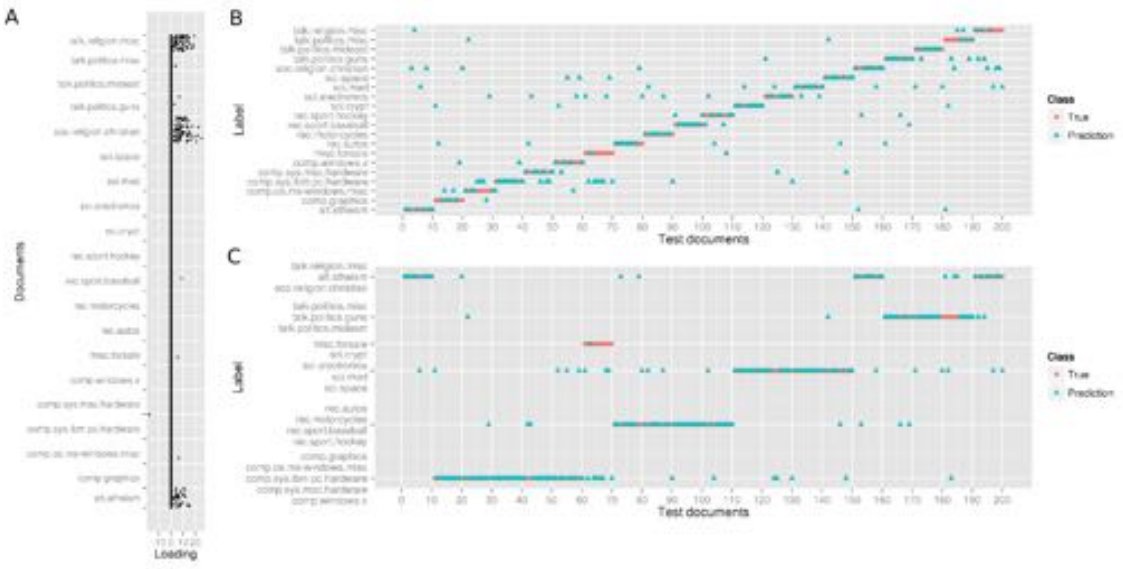


FIGURE 2.11: Newsgroup prediction on 200 test documents. Panel A: One factor loading selected as shared by three newsgroups ( `talk.religion.misc`, `alt.atheism` and `soc.religion.christian`). Panel B: 20 newsgroups prediction on 100 test documents using ten nearest neighbors based on estimated loadings. Panel C: Document group prediction based on high level classes with similar subject matter using ten nearest neighbors based on estimated loadings.

closely related to each other, while others are highly unrelated, we further partition them into six classes according to subject matter. Then ten nearest neighbors are used to predict this high level classes of the test data. We obtain an approximately 75.05% accuracy using Hamming loss (Figure 2.11C).

## 2.7 Discussion and conclusion

In this chapter we have developed a Bayesian group factor analysis model with a hierarchical prior that induces both column-wise and element-wise sparsity. There exists a reach literature studying paired or multiple views jointly (e.g. Parkhomenko et al. (2009); Witten and Tibshirani (2009); Zhao and Li (2012) among others). The line of interpreting as linear factor analysis models includes the original inter-battery/multi-battery model (Browne, 1979, 1980), the probabilistic CCA model (Bach and Jordan,

Table 2.13: First ten words of the group shared factors for six different newsgroup classes.

| Newsgroup classes | First ten shared words | | Newsgroup classes | First ten shared words | |
|---|---|---|---|---|---|
| comp.graphics | windows | dos | | sale | shipping |
| comp.os.ms-windows.misc | thanks | mac | | sell | ca |
| comp.sys.ibm.pc.hardware | graphics | go | misc.forsale | condition | wanted |
| comp.sys.mac.hardware | file | scsi | | offer | thanks |
| comp.windows.x | window | server | | forsale | edu |
| | dod | baseball | | government | it |
| rec.autos | car | ride | talk.politics.misc | israeli | israel |
| rec.motorcycles | bike | cars | talk.politics.guns | jews | gun |
| rec.sport.baseball | motorcycle | bmw | talk.politics.mideast | atf | guns |
| rec.sport.hockey | game | team | | firearms | batf |
| | clipper | henry | | god | bible |
| sci.crypt | encryption | orbit | talk.religion.misc | bible | heaven |
| sci.electronics | space | people | alt.atheism | christian | sandvik |
| sci.med | chip | circuit | soc.religion.christian | clh | faith |
| sci.space | digex | voltage | | jesus | church |

2005), the sparse probabilistic projection (Archambeau and Bach, 2009) and most recently the BCCA (Klami et al., 2013) and GFA models (Klami et al., 2014a). It is until recently that the column-wise sparsity (or group-wise sparsity) has been appreciated in this problem, mostly because its effects of decoupling latent variables from views and adaptively selecting factor numbers. This is mostly due to the work of the Bayesian version of CCA (Virtanen et al., 2011). Recently sGFA model is developed to combine column-wise and element-wise sparsity using a combination of independent ARD priors and a spike-slab prior for column selection. The model developed in this chapter pushes one step further using a more effective shrinkage prior and allowing sparse and dense mixture on the factor loadings. Modeling sparse and dense loadings is very closely related to the problem of low rank and sparse decomposition of covariance matrices (Chandrasekaran et al., 2009; Candès et al., 2011; Zhou et al., 2011). With the assumption of full column rank of dense loadings and one single view, our model provides a Bayesian solution to the low rank/sparse decomposition problem.

The column-wise shrinkage in BASS is achieved through the top two layers of the TPB distribution. With current parameter settings, it is equivalent to the standard

horseshoe prior put on the entire column. The horseshoe prior has been shown to induce better shrinkage effects compared to the Student-t prior (ARD), the double-exponential prior (Bayesian lasso) and other similar shrinkage priors, at the same time maintaining a good computational tractability (Carvalho et al., 2010). In addition, our local shrinkage induces element-wise sparsity. Combined with the two-component mixture this allows the dense and sparse factors in any combinations of views. Shared dense factors could be viewed as supervised low rank decomposition if we treat one view as labels. Shared sparse factors capture interpretable associations among variables in different views. To our knowledge our model BASS is the first such model that allows different dense/sparse factor combinations among multiple views.

We develop EM algorithms to find MAP solutions of our model. The random initialized EM algorithm is easily stuck in bad initial loading orientations. We further develop a fast and robust PX-EM algorithm by introducing expanded rotation matrix. Our method utilizes the rotation invariance property of likelihood for our joint factor model (Ročková and George, 2015). Introducing this additional parameter greatly facilitates the EM algorithm to escape from bad initializations. The additional complexity we paid is a single Cholesky decomposition and a matrix multiplication. We compare original EM, PX-EM and EM with a few MCMC runs as initialization (MCMC-EM) in simulations. Results show after a few PX-EM runs our model can find a good orientation that matches our prior favorable structure. In addition, our PX-EM is faster than the GFA model and sGFA model.

In this study we focus on the interpretation of those models from a factor analysis point of view. By concatenating all the view matrices, we get a joint factor model. It has been long appreciated that the problem of limited sample sizes in factor models (Carvalho et al., 2008). Concatenation of multiple views makes this problem worse due to that it only increases variables not observations. The structured regularization

of the joint loading is one necessary approach to provide meaningful solutions. BASS achieves this through a structured shrinkage prior with fast and robust parameter estimations.

The extensions of multi-view linear factors models to non-linear or non-Gaussian models have been studies recently (Salomatin et al., 2009; Damianou et al., 2012; Klami et al., 2014b; Klami, 2014). The idea of inducing structured sparsity in the loadings can be analogized in both of the settings. For example, we could consider more sophisticated Gaussian process kernels in the non-linear models, and formulate in a structured way. We anticipate such multi-view models would be more popular in the further.

# 3

# MELD - a fast moment estimation approach for generalized Dirichlet latent variable models

Many modern statistical applications require the analysis of large scale, heterogeneous data types including continuous, categorical, and count variables. For example, in social science, survey data often consist of collections of different data types (e.g., height, gender, and age). In population genetics, researchers are interested in analyzing genotype (integer-valued) and heterogeneous traits of varying data types. Often data take the form of an $n \times p$ matrix $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^\top$, with $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip})^\top$ a $p$ dimensional vector of measurements of varying types for $i$th subject with $i = 1, \ldots, n$.

In this chapter we contribute to the existing literature in two important aspects. First, we develop a new model for mixed data types. The new model is called *generalized latent Dirichlet model*. Such a model assumes that each subject *partially* belongs to $k$ different components, with mixture proportions of the components following a Dirichlet distribution. The model has been studied following different trajectories, e.g. in population genetics (Pritchard et al., 2000b,a), documents modeling (Blei

et al., 2003) and contingency table modeling (Bhattacharya and Dunson, 2012). Those models are also known as *mixed membership models*, with the name indicating the partial membership of each subject. A recent book by Airoldi et al. (2014) gives a comprehensive introduction to this class of models. The generalized latent Dirichlet model proposed in this chapter extends previous models to allow mixed data types in a unified modeling framework.

The second contribution of this chapter is that we propose a generalized method of moments (GMM) approach to estimate parameters for the proposed new model. In contrast to previous estimation methods such as EM or MCMC algorithms, the GMM approach developed here does not require initiation of latent variables. This is achieved by extending the moment tensor methods proposed recently (Anandkumar et al., 2014b). Our GMM approach distinguishes itself from those moment tensor methods in multiple ways. First previous methods rely on matrix decomposition techniques such as singular value decomposition (SVD) or eigenvalue decomposition. This limitation prevents those methods to estimate an over complete component parameters, meaning the number of latent components greater than the dimension of observed variables. Our GMM approach circumvents this limitation and could be used to the case where the number of components is greater than the dimension of variables. Second our moment functions are defined as low order (second or third) *heterogeneous* polynomials instead of homogeneous polynomials. This allows us to develop a fast coordinate descent algorithm, which could not be achieved if homogeneous polynomials are used. We name our approach MELD standing for Moment Estimation for generalized Latent Dirichlet variable model.

The rest of this chapter is organized as follows. We introduce our generalized latent Dirichlet variable model in Section 3.1. Some well known existing models are reviewed and their connections with our new model are discussed in this section. In Section 3.2 we start with generalized method of moments and introduce the moment

functions used to perform parameter estimations in our new model. A two stage estimation procedure is proposed and a fast coordinate descent algorithm is developed. In Section 3.3 we demonstrate our approach by multiple simulations. In Section 3.4 we apply our approach to three public available data sets. We conclude this chapter by a discussion in Section 3.5.

## 3.1 Generalized latent Dirichlet variable models

In this section we introduce a new generalized latent Dirichlet variable model for modeling mixed data types.

### *3.1.1 Modeling mixed data types*

We first give a brief review of existing approaches for modeling mixed data types. The history of modeling mixed data types has been mainly following two paths. One approach assumes there are latent Gaussian variables behind observations and the observed variables with mixed data types are manifestations (indicators) of the latent variables (Muthén, 1983, 1984). Those latent variables are also called latent traits (Arminger and Küsters, 1988) or liability scores in population genetics literature (Luo et al., 2001). Those models are routinely used in the social science literature, focusing almost entirely on the case in which data are categorical or continuous. The categorical observed variables are resulted by thresholding the latent Gaussian variables. The well known probit model belongs to this class. Often structural equation modeling approaches or factor analysis models are used to model the dependence structure among latent variables (Muthén, 1984; Shi and Lee, 2000; Quinn, 2004). Most recently Hoff (2007) develops an extended rank likelihood approach for mixed data types using a semiparametric copula model. Latent Gaussian variables are assumed separately from the marginal distributions of observed variables through an inverse cumulative distribution function (CDF) technique. The approach augments

the copula model with latent variables that satisfy rank constraints and parameter estimations are performed using MCMC algorithms. The idea soon gains popularity and applications and extensions of the original method have been developed (Gruhl et al., 2013; Murray et al., 2013).

A second approach for modeling mixed data types defines an exponential family distribution for each of the observed variables. This approach uses theories from generalized linear models with independent variables (predictors) partially replaced by latent variables (Sammel et al., 1997; Moustaki and Knott, 2000; Dunson, 2000, 2003). Dependence among observed variables is achieved by assuming certain latent variables are shared across the generalized linear models. Those models are potentially flexible due to each variable is modeled by its own distribution. However one limitation is that the latent variables determine both the shape of the marginal distribution and the dependence structure among observed variables.

### *3.1.2 Latent Dirichlet variable model with mixed data types*

In this subsection we introduce a new model for mixed data types. We call the model generalized latent Dirichlet variable model. Let $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip})^\top$ be a $p$ dimensional observation for subject $i$. The latent Dirichlet variable model assumes each variable of $\boldsymbol{y}_i$ is drawn from a mixture of distributions specified for that particular variable, and the mixture weights of different variables for subject $i$ are the same. Let's assume there are $k$ latent components. We denote the mixture weight vector assigned on the $k$ components for subject $i$ as $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ik})^\top \in \Delta^{k-1}$. Here $\Delta^{k-1}$ denotes the $k-1$ probability simplex, indicating the sum over each coordinate of $\boldsymbol{x}_i$ being one. Conditional on $\boldsymbol{x}_i$, the distribution of $j$th variable of $\boldsymbol{y}_i$ is assumed to be

$$y_{ij}|\boldsymbol{x}_i \sim \sum_{h=1}^{k} x_{ih} g_j(\boldsymbol{\phi}_{jh}), \tag{3.1}$$

where $g_j(\boldsymbol{\phi}_{jh})$ is the density of the $j$th variable in component $h$. In this setting, a pure subject has a weight vector with all zeros except for a single one. We view the mixture weight $\boldsymbol{x}_i$ as latent variable. A full likelihood specification is completed by choosing a population distribution for the latent variable $\boldsymbol{x}_i \sim P$, with $P$ a distribution on the simplex $\Delta^{k-1}$. In this work, we put a Dirichlet distribution on this latent variable vector $\boldsymbol{x}_i \sim \text{Dir}(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^\top$. The resulting model is called a generalized latent Dirichlet variable model.

The corresponding density $g_j(\boldsymbol{\phi}_{jh})$ is absolutely continuous with respect to a dominating measure $(\Omega, \mathcal{H}, \mu)$, and it is indexed by parameter $\boldsymbol{\phi}_{jh}$. Currently we do not specify its parametric form. It could be chosen to belong to the exponential family. We further let $\boldsymbol{m}_i = (m_{i1}, \ldots m_{ip})^\top$ denote a membership vector for subject $i$, where $m_{ij} \in \{1, \ldots, k\}$ indicates the component that $y_{ij}$ is generated from. Model (3.1) can be written in a generative form

$$\begin{aligned} y_{ij} \mid m_{ij} = h &\sim g_j(\boldsymbol{\phi}_{jh}), \\ m_{ij} \mid \boldsymbol{x}_i &\sim \text{Multi}(\boldsymbol{x}_i), \\ \boldsymbol{x}_i &\sim \text{Dir}(\boldsymbol{\alpha}). \end{aligned} \tag{3.2}$$

With specifications of the density function $g_j(\boldsymbol{\phi}_{jh})$, our model reduces to several well known models, as reviewed below.

*Latent Dirichlet allocation*

Modeling documents using probabilistic models has a long history. The bag-of-words model studied by Hofmann (1999) assumes the word in a document follows a mixture of multinomial distributions. Latent Dirichlet allocation (LDA) (Blei et al., 2003) extends the model to allow each document has its own mixture weights. One way of presenting LDA model is

$$y_{ij} | m_{ij} = h \sim \text{Multi}(\boldsymbol{\phi}_h),$$

$$m_{ij}|\boldsymbol{x}_i \sim \text{Multi}(\boldsymbol{x}_i),$$
$$\boldsymbol{x}_i \sim \text{Dir}(\boldsymbol{\alpha}). \tag{3.3}$$

Here $y_{ij}$ is the $j$th position in document $i$ and $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip_i})^\top$ is a document with $p_i$ positions. We distinguish *position* in a document from a *word* in the sense that a word is reserved for indicating a unit in a vocabulary. Two words are of different types. Instead a position is a realization of a word in a document and two positions could take the same word. $\boldsymbol{\phi}_h$ is a probability vector over the whole vocabulary specified for $h$th component. $\boldsymbol{x}_i$ is the mixture weights over $k$ components for $i$th document. To complete the model we assign a Poisson distribution over the total number of positions $p_i$ in document $i$. The number of categories $y_{ij}$ could take, say $d$, equals to the number of words in the vocabulary. Since $\boldsymbol{\phi}_h$ defines a multinomial distribution over the vocabulary, it is interpreted as a topic (Blei et al., 2003). Another way of interpreting the LDA model is to use Poisson distributions. This is because for $d$ independent Poisson variables $n_1, \ldots, n_d$ with rate $\lambda_c$ for $c = 1, \ldots, d$, conditional on the total, their distribution is equivalent to a multinomial distribution Multi$(n_0, \boldsymbol{\pi})$ with $n_0 = \sum_c n_c$ and $\boldsymbol{\pi} = (\lambda_1, \ldots, \lambda_d)^\top / (\sum_c \lambda_c)$. Therefore the LDA model could also be written as

$$y_{ij}|m_{ij} = h \sim \text{Poisson}(\phi_{jh}),$$
$$m_{ij}|\boldsymbol{x}_i \sim \text{Multi}(\boldsymbol{x}_i),$$
$$\boldsymbol{x}_i \sim \text{Dir}(\boldsymbol{\alpha}). \tag{3.4}$$

Here $y_{ij}$ is the number of counts of $j$th word in document $i$ and $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{id})^\top$ is a summary of word counts. $\phi_{jh}$ is a scalar indicating the Poisson rate for $j$th word in topic $h$. When we are not interested in modeling the total positions in a document, above two models are equivalent.

*Admixture model in population genetics*

The multilocus admixture model proposed in population genetics literature has the similar idea as the LDA model (Pritchard et al., 2000b). In the original paper, the genotype data of $n$ dipoid individuals at $p$ loci are modeled. Let $y_{ij}^{(1)}$ and $y_{ij}^{(2)}$ denote the genotype of $i$th individual at locus $j$. Because there are two chromosomes in a dipoid individual, each of which is originated from one parent, therefore the ordered pair $(y_{ij}^{(1)}, y_{ij}^{(2)})$ is used to denote the genotype. When such pairs are not observed, we could use phasing methods to estimate the haplotype (Browning and Browning, 2011) and then obtain such pairs. Assuming $(y_{ij}^{(1)}, y_{ij}^{(2)})$ is known, the admixture model proposed by Pritchard et al. (2000b) assumes

$$\begin{aligned} y_{ij}^{(\cdot)} | m_{ij}^{(\cdot)} = h &\sim \text{Multi}(\boldsymbol{\phi}_{jh}), \\ m_{ij}^{(\cdot)} | \boldsymbol{x}_i &\sim \text{Multi}(\boldsymbol{x}_i), \\ \boldsymbol{x}_i &\sim \text{Dir}(\boldsymbol{\alpha}). \end{aligned} \tag{3.5}$$

This model assumes the two copies of the genotype at $j$th locus for individual $i$ have their own membership variables. $y_{ij}^{(\cdot)}$ at both copies could take $d_j$ different values, where $d_j$ equals to the number of possible alleles at locus $j$. $\boldsymbol{\phi}_{jh}$ is the genotype distribution for $j$th locus in population $h$. Its dimension equals to $d_j$. For example, when single nucleotide polymorphism (SNP) data are studied, $d_j = 2$ and $y_{ij}^{(\cdot)}$ could take $(0, 1)$ two values, meaning missing or existing of a particular reference allele. If we further assume Hardy-Weinberg equilibrium, meaning that the two copies of the allele are independently inherited from the two parents with a common population frequency, we could re-write the model by letting $y_{ij} = y_{ij}^{(1)} + y_{ij}^{(2)}$ as

$$\begin{aligned} y_{ij} | m_{ij} = h &\sim \text{Binomial}(2, \phi_{jh}), \\ m_{ij} | \boldsymbol{x}_i &\sim \text{Multi}(\boldsymbol{x}_i), \\ \boldsymbol{x}_i &\sim \text{Dir}(\boldsymbol{\alpha}). \end{aligned} \tag{3.6}$$

Here $\phi_{jh}$ indicates the reference allele frequency in population $h$ at locus $j$.

*Simplex factor model for contingency tables*

The simplex factor model proposed by Bhattacharya and Dunson (2012) for contingency table modeling also could be written in a similar manner. We first introduce some basic concepts in modeling of contingency tables. For $n$ observations of a $p$ dimensional categorical vector $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip})^\top$ with $y_{ij} \in \{1, \ldots, d_j\}$ for $i = 1, \ldots, n$, the data could be formulated into a $p$ way contingency table of dimension $d_1 \times d_2 \cdots \times d_p$ (Dunson and Xing, 2009). Let $\boldsymbol{c} = (c_1, \ldots, c_p)^\top$ with $c_j \in \{1, \ldots, d_j\}$, researchers are interested in modeling the probability of $\pi_{\boldsymbol{c}} = \Pr(\boldsymbol{y}_i = \boldsymbol{c}) = \Pr(y_{i1} = c_1, \ldots, y_{ip} = c_p)$, which is the probability of observing a particular cell in the $p$ way table. We have $\sum_{\boldsymbol{c}} \pi_{\boldsymbol{c}} = 1$ where the summation is taken over all cells in the table. Dunson and Xing (2009) call $\boldsymbol{\pi} = \{\pi_{\boldsymbol{c}}\}$ a probability tensor. Modeling contingency table becomes finding a parsimonious way to represent the probability tensor.

The simplex factor model proposed by Bhattacharya and Dunson (2012) assumes

$$\Pr(y_{ij} = c_j | \boldsymbol{x}_i, \boldsymbol{\Phi}_j) = \sum_{h=1}^{k} x_{ih} \phi_{jhc_j},$$

where $\boldsymbol{\Phi}_j = (\boldsymbol{\phi}_{j1}, \ldots, \boldsymbol{\phi}_{jk})$ is the collection of probability vectors for $j$th variable in the $k$ latent components. $\boldsymbol{x}_i$ is a Dirichlet latent variable in $k-1$ simplex $\Delta^{k-1}$. The authors show that this model can be viewed as a Tucker decomposition of the probability tensor $\boldsymbol{\pi}$

$$\pi_{\boldsymbol{c}} = \int \prod_{j=1}^{p} \Pr(y_{ij} = c_j | \boldsymbol{x}_i, \boldsymbol{\Phi}_j) \mathrm{d}P(\boldsymbol{x}_i) = \sum_{h_1=1}^{k} \cdots \sum_{h_p=1}^{k} g_{h_1,\ldots,h_p} \prod_{j=1}^{p} \phi_{jh_jc_j}.$$

The arms in the decomposition correspond to $\boldsymbol{\phi}_{jh}$'s for the component distributions. The core tensor depends on the distribution assumption on the latent Dirichlet variable $P(\boldsymbol{x}_i)$. By augmenting the simplex factor model with membership variable $m_{ij}$,

we get an equivalent representation

$$\begin{aligned} y_{ij}|m_{ij} = h &\sim \text{Multi}(\boldsymbol{\phi}_{jh}), \\ m_{ij}|\boldsymbol{x}_i &\sim \text{Multi}(\boldsymbol{x}_i), \\ \boldsymbol{x}_i &\sim \text{Dir}(\boldsymbol{\alpha}). \end{aligned} \tag{3.7}$$

To sum up, the models reviewed above in (3.3), (3.4), (3.5), (3.6) and (3.7) can be thought of as special forms of our model in (3.2). Model (3.2) generalizes the traditional mixed membership models to allow mixed data types. The parameters $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{jh}\}_{1\leqslant j\leqslant p, 1\leqslant h\leqslant k}$ are mixture component parameters shared by all subjects.

## 3.2 Generalized method of moments for parameter estimation

With $n$ independent samples $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ from model (3.2), we can write following likelihood after marginalizing over the latent Dirichlet variables

$$p(\boldsymbol{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}) = \prod_{i=1}^{n} \Bigg[ \int \prod_{j=1}^{p} \Bigg( \sum_{h=1}^{k} x_{ih} g_j(y_{ij} \mid \boldsymbol{\phi}_{jh}) \Bigg) \mathrm{d}P(\boldsymbol{x}_i) \Bigg].$$

One can choose a specific form of the component distribution $g_j(y_{ij} \mid \boldsymbol{\phi}_{jh})$ for each of the $j$th variable. Then a complete likelihood can be obtained by augmenting the model with membership variables $\boldsymbol{M} = \{m_{ij}\}_{1\leqslant i\leqslant n, 1\leqslant j\leqslant p}$. Parameter estimation can be done using EM or MCMC algorithms. Those algorithms alternate between updating latent variables, membership variables and population parameters, which inevitably leads to slow convergence, inefficiency and instability.

In this chapter we are going to use method of moments, particularly generalized method of moments (GMM), to perform parameter estimation in model (3.2). Our GMM does not require initiation of latent variables. In addition in using GMM we do not need to specify a distributional form for $g_j(y_{ij} \mid \boldsymbol{\phi}_{jh})$. Instead, only certain moments are required for parameter estimation. Our GMM is related to recent moment tensor methods developed for latent variable models including mixture of

Gaussians, hidden Markov models, mixed membership models, and stochastic block models (Arora et al., 2012; Anandkumar et al., 2012a,b; Hsu and Kakade, 2013; Anandkumar et al., 2014a,b). However our GMM is different from previous methods in that heterogeneous low order polynomials are used instead of homogeneous polynomials.

In this section we first review moment methods in parameter estimation. Then we introduce the moment functions and the quadratic objectives for our generalized latent Dirichlet model (3.2).

### *3.2.1 A brief summary of generalized method of moments*

Using the method of moments (MM) to perform parameter estimation has a long history, dating back to Pearson's method to estimate a mixture of two Gaussian distributions (Pearson, 1894). The idea behind MM is to derive a list of moment functions that have expectation of zero at the true parameter values. For a set of observations $\boldsymbol{y}_i$ with $i = 1, \ldots, n$, MM specifies $\ell$ moment functions to form a moment vector $\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta}) = (f_1(\boldsymbol{y}_i, \boldsymbol{\theta}), \ldots, f_\ell(\boldsymbol{y}_i, \boldsymbol{\theta}))^\top$ satisfying $\mathbb{E}[\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta})] = \mathbf{0}$ at the true parameter $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p$. A parameter estimate can be found by solving the $\ell$ sample equations with $p$ unknowns

$$\widehat{\boldsymbol{\theta}} \text{ such that } \boldsymbol{f}_n(\boldsymbol{\theta}) \equiv \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta}) = \mathbf{0} \text{ at } \widehat{\boldsymbol{\theta}}. \tag{3.8}$$

When $\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta})$ is linear in $\boldsymbol{\theta}$ (e.g., linear regression with instrumental variables) with independent moment conditions and $\ell = p$, then $\boldsymbol{\theta}$ can be uniquely determined. When $\ell > p$, the system in (3.8) might be over-determined, in which case standard MM cannot be applied.

Generalized method of moments (GMM) addresses this problem by minimizing the following quadratic equation

$$\widehat{\boldsymbol{\theta}} \equiv \arg\min_{\boldsymbol{\theta}} \big[ Q_n(\boldsymbol{\theta}; \boldsymbol{A}_n) = \boldsymbol{f}_n(\boldsymbol{\theta})^\top \boldsymbol{A}_n \, \boldsymbol{f}_n(\boldsymbol{\theta}) \big], \tag{3.9}$$

where $\boldsymbol{A}_n$ is a positive semidefinite weight matrix. Hansen establishes the asymptotic theory of GMM estimators (Hansen, 1982). The efficiencies of GMM estimators are shown to depend on the weight matrix. The asymptotically optimal weight matrix can be chosen such that

$$\boldsymbol{A}_n^{-1} = \boldsymbol{S}_n = \mathrm{Var}[n^{1/2}\boldsymbol{f}_n(\boldsymbol{\theta})].$$

Hansen proposes a two stage estimation procedure in which an initial estimate of $\widehat{\boldsymbol{\theta}}$ is found using a suboptimal weight matrix, such as the identity. This initial parameter estimate is then used to calculate the weight matrix $\boldsymbol{A}_n$. This updated matrix is then used in (3.9) to obtain the final parameter estimate (Hall, 2005). Consistency and asymptotic normality of GMM has been studied by Hansen (1982).

Methods of applications of GMM to latent variable models with structural assumptions also have a long history. Those methods match certain sample moment statistics to their population counterparts. Minimizing their weighted distance generates a generalized least square estimator (Browne, 1973; Bentler, 1983; Anderson and Gerbing, 1988). Although those methods are called generalized least square methods, they share the same idea with the GMM estimator. More recently, Gallant et al. (2013) apply GMM to a specific class of latent variable models by defining moment conditions based on the complete data, including the latent variables. In contrast, Bollen et al. (2014) rely on a model-implied instrumental variable to find a GMM estimator. Generally, current GMM approaches focus on latent variable models that satisfy restrictive assumptions or require the instantiation of latent variables in a computationally intensive estimation algorithm.

### *3.2.2 Moment functions in MELD*

In this subsection we describe the GMM developed for your generalized latent Dirichlet variable models. Applying GMM to all of the parameters in model (3.2) is not feasible due to the massive dimensionality of the parameter space. Higher-order

moment functions are complex and involve large numbers of unknowns. They are also unstable often with large variances. We will build on the moment tensor methods established recently and define moment functions that depend on lower order moments.

A series of recent work about using moment tensors to estimate parameters in latent variable models have been proposed. Hsu et al. (2012); Anandkumar et al. (2012a,b) are believed to be first papers using methods of moments with third order moment tensor to perform parameter estimation. The authors show that second order moment matrix and third order moment tensor have a decomposable form for certain latent variable models including hidden Markov model, independent component analysis and latent Dirichlet allocation model. Parameter estimation is conducted by first projecting the third order moment tensor to a matrix and then using singular value decompositions (SVD's) to the projected matrix and the original second order moment matrix. Model parameters can be recovered from the singular values/vectors. Arora et al. (2012) extend the SVD approach to nonnegative matrix factorization (NMF). Hsu and Kakade (2013) develop a method of moments for mixture of spherical Gaussians. Anandkumar et al. (2014a) study the stochastic block model using the similar idea. Anandkumar et al. (2014b) further establish tensor power methods for parameter estimations using moment tensors. The moment tensor approach offers substantial computational advantages over other methods, as illustrated in various applications (Tung and Smola, 2014; Anandkumar et al., 2014a; Colombo and Vlassis, 2015).

We now state the GMM developed in this chapter. Recall that $y_{ij}$ is the $j$th variable for subject $i$, which may include various data types, including continuous, categorical, or count data. We encode $y_{ij}$ as follows. When $y_{ij}$ is a categorical variable, it is represented as a binary vector $\boldsymbol{b}_{ij}$, where the $c_j$th coordinate is set to one and all others are zero, where $c_j$ is the category of observation $y_{ij}$. If $y_{ij}$ is a

non-categorical variable, then $\boldsymbol{b}_{ij} \equiv y_{ij}$ is a scalar value. The variable $\boldsymbol{b}_{ij}$ represents $y_{ij}$ with mixed data types in the moment functions. In the following, we assume that there are $k$ latent components and that the latent Dirichlet variable $\boldsymbol{x}_i$ follows a Dirichlet distribution with parameter $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_k)^\top$. We let $\alpha_0 = \sum_{h=1}^{k} \alpha_h$. We let $\boldsymbol{\phi}_{jh}$ denote the mean parameter of $\boldsymbol{b}_{ij}$ in component $h$

$$\boldsymbol{\phi}_{jh} = \mathbb{E}(\boldsymbol{b}_{ij} | m_{ij} = h).$$

When $y_{ij}$ is categorical, $\boldsymbol{\phi}_{jh}$ is a vector. When $y_{ij}$ is a scalar, $\boldsymbol{\phi}_{jh}$ is also a scalar. We define the following two types of moment functions

$$\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) = \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it} - \frac{\alpha_0}{\alpha_0 + 1} \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_t - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^\top, \tag{3.10}$$

$$1 \leqslant j, t \leqslant p,\ j \neq t$$

$$\begin{aligned}\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) &= \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it} \\ &- \frac{\alpha_0}{\alpha_0 + 2} \Big( \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{\mu}_t + \boldsymbol{\mu}_j \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it} + \boldsymbol{b}_{ij} \circ \boldsymbol{\mu}_s \circ \boldsymbol{b}_{it} \Big) \\ &+ \frac{2\alpha_0^2}{(\alpha_0 + 1)(\alpha_0 + 2)} \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_s \circ \boldsymbol{\mu}_t - \boldsymbol{\Lambda}^{(3)} \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t, \end{aligned} \tag{3.11}$$

$$1 \leqslant j, s, t \leqslant p,\ j \neq s \neq t.$$

Here $\boldsymbol{\mu}_j = \mathbb{E}(\boldsymbol{b}_{ij}) = \boldsymbol{\Phi}_j \boldsymbol{\alpha} / \alpha_0$ for $j = 1, \ldots, p$. $\boldsymbol{\Lambda}^{(2)} = \operatorname{diag}\big(\boldsymbol{\alpha} / [\alpha_0(\alpha_0 + 1)]\big)$ and $\boldsymbol{\Lambda}^{(3)}$ is a three-way diagonal tensor with $\lambda_h^{(3)} = 2\alpha_h / [\alpha_0(\alpha_0 + 1)(\alpha_0 + 2)]$ on the diagonal for $h = 1, \ldots, k$. We use $\circ$ to denote an outer product, and use $\times_s$ to indicate multiplication of a tensor with a matrix for mode $s$. The second moment function $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ is a $d_j \times d_t$ matrix, and the third moment function $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ is a $d_j \times d_s \times d_t$ tensor. We set $d_j = 1$ when the $j$th variable is non-categorical.

The following theorem states that, at the true parameter value, the expectations of the moment functions are zero. The proof can be found in Appendix B.1.

**Theorem 3.1** (Moment conditions in MELD)**.** *The expectations of the second moment matrix* $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ *and third moment tensor* $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ *defined in* (3.10) *and* (3.11) *are zero at true model parameter values* $\boldsymbol{\Phi}_0$

$$\mathbb{E}[\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}_0)] = \mathbf{0}, \quad \mathbb{E}[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi}_0)] = \mathbf{0}.$$

*3.2.3 Two stage optimal estimation*

We will use the two types of moment functions in (3.10) and (3.11) in Hansen's two stage optimal GMM estimation procedure. We re-state Hansen's two stage GMM estimation procedure as follows

(1) Estimate $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \big[Q_n(\boldsymbol{\theta}; \boldsymbol{I}) = \boldsymbol{f}_n(\boldsymbol{\theta})^\top \boldsymbol{f}_n(\boldsymbol{\theta})\big]$.

(2) Given $\widehat{\boldsymbol{\theta}}$ calculate $\boldsymbol{S}_n$ and set $\boldsymbol{A}_n = \boldsymbol{S}_n^{-1}$.

(3) Compute $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \big[Q_n(\boldsymbol{\theta}; \boldsymbol{A}_n) = \boldsymbol{f}_n(\boldsymbol{\theta})^\top \boldsymbol{A}_n \boldsymbol{f}_n(\boldsymbol{\theta})\big]$ as final estimator.

Based on $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ and $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$, we now define two versions of moment vectors by stacking the second moment matrices and third moment tensors

$$\begin{aligned}\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) = \Big(&\mathrm{vec}[\boldsymbol{F}_{12}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{1p}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \\ &\mathrm{vec}[\boldsymbol{F}_{23}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{2p}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{p-1,p}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top\Big)^\top. \qquad (3.12)\end{aligned}$$

$$\begin{aligned}\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) = \Big(&\mathrm{vec}[\boldsymbol{F}_{12}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{p-1,p}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \\ &\mathrm{vec}[\boldsymbol{F}_{123}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{12p}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \\ &\mathrm{vec}[\boldsymbol{F}_{134}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top, \ldots, \mathrm{vec}[\boldsymbol{F}_{13p}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top \ldots, \mathrm{vec}[\boldsymbol{F}_{p-2,p-1,p}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top\Big)^\top .. \qquad (3.13)\end{aligned}$$

The first version of moment vector $\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ depends on second moment matrices and the second version of moment vector $\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ depends on both second moment

matrices and third moment tensors. The vectorized moment matrices and tensors are ordered in a way that the subscript index on the right side runs faster than the subscript index on the left. The second moment matrix $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ is symmetric, so we only need to consider the matrices with $j < t$ in the moment vectors. Assuming $d$ levels for $\boldsymbol{y}_i$ results in a moment vector of dimension $p(p-1)d^2/2$. The third moment tensor $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ is also symmetric with respect to its indices, so we only include moment tensors with $j < s < t$ when $\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ is formed. The dimension of the second version of moment vector is $p(p-1)d^2/2 + [p^3 - 3p(p-1) - p]\, d^3/6$.

We now state the quadratic functions we use for parameter estimation as follows

$$Q_n^{(2)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(2)}) = \boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})^\top \boldsymbol{A}_n^{(2)} \boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi}), \tag{3.14}$$

$$Q_n^{(3)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(3)}) = \boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})^\top \boldsymbol{A}_n^{(3)} \boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi}), \tag{3.15}$$

where $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$ and $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$ are sample estimates of the expectations of the moment vectors

$$\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}), \quad \boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi}),$$

and $\boldsymbol{A}_n^{(2)}$ and $\boldsymbol{A}_n^{(3)}$ are two positive semidefinite matrices. When we calculate $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$, $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_t$ in $\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ are replaced by their sample estimates

$$\widehat{\boldsymbol{\mu}}_j = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{b}_{ij}, \qquad \widehat{\boldsymbol{\mu}}_t = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{b}_{it},$$

instead of their parametric counterparts $\boldsymbol{\Phi}_j\boldsymbol{\alpha}/\alpha_0$ and $\boldsymbol{\Phi}_t\boldsymbol{\alpha}/\alpha_0$ respectively. Similarly $\widehat{\boldsymbol{\mu}}_j$, $\widehat{\boldsymbol{\mu}}_s$ and $\widehat{\boldsymbol{\mu}}_t$ are used in $\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ for calculating $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$. Avoiding using their parametric forms allows us to develop a fast coordinate descent algorithm for GMM estimation.

In the first stage, we set $\boldsymbol{A}_n^{(\cdot)}$ to an identity matrix. Then the quadratic functions in (3.14) and (3.15) can be re-written as follows

$$Q_n^{(2)}(\boldsymbol{\Phi}, \boldsymbol{I}) = \sum_{j=1}^{p-1}\sum_{t=j+1}^{p} ||\boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi})||_F^2,$$

$$Q_n^{(3)}(\boldsymbol{\Phi}, \boldsymbol{I}) = \sum_{j=1}^{p-1} \sum_{t=j+1}^{p} ||\boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi})||_F^2 + \sum_{j=1}^{p-2} \sum_{s=j+1}^{p-1} \sum_{t=s+1}^{p} ||\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})||_F^2,$$

where $\boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi})$ and $\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})$ are sample estimates of the expectations of the second moment matrix and third moment tensor with $\boldsymbol{\mu}_j$, $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ replaced by $\widehat{\boldsymbol{\mu}}_j$, $\widehat{\boldsymbol{\mu}}_s$ and $\widehat{\boldsymbol{\mu}}_t$ , and $||\cdot||_F^2$ indicates the Frobenius norm, the element-wise sum of squares.

We obtain a first stage estimator of $\boldsymbol{\Phi}$ by minimizing the quadratic forms using Newton-Raphson. Note that only the last term of $\boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi})$ and $\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})$ involves unknown parameter $\boldsymbol{\Phi}$. For simplicity we denote

$$\boldsymbol{E}_{n,jt}^{(2)} = \boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi}) + \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^\top, \tag{3.16}$$

$$\boldsymbol{E}_{n,jst}^{(3)} = \boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi}) + \boldsymbol{\Lambda}^{(3)} \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t. \tag{3.17}$$

Note that $\boldsymbol{E}_{n,jt}^{(2)}$ and $\boldsymbol{E}_{n,jst}^{(3)}$ can be computed directly from the samples. We optimize $\boldsymbol{\phi}_{jh}$ with other mean parameters fixed. After calculating the gradient and Hessian of $Q^{(2)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I})$, the update rule simply becomes

$$\boldsymbol{\phi}_{jh}^s = \frac{\sum_{t=1, t\neq j}^{p} (\overline{\boldsymbol{E}}_{n,jt}^{(2)} \boldsymbol{\phi}_{th})^\top}{(\lambda_h^{(2)}) \sum_{t=1, t\neq j}^{p} \boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th}}, \tag{3.18}$$

where $\overline{\boldsymbol{E}}_{n,jt}^{(2)} = \boldsymbol{E}_{n,jt}^{(2)} - \sum_{h'\neq h} \lambda_{h'}^{(2)} \boldsymbol{\phi}_{jh'} \circ \boldsymbol{\phi}_{th'}$ and $\lambda_h^{(2)}$ is the $h$th diagonal entry of $\boldsymbol{\Lambda}^{(2)}$. The update rule for $\boldsymbol{\phi}_{jh}$ with $Q^{(3)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I})$ can be calculated as

$$\boldsymbol{\phi}_{jh}^s = \frac{\lambda_h^{(2)} \sum_{t=1, t\neq j}^{p} (\overline{\boldsymbol{E}}_{n,jt}^{(2)} \boldsymbol{\phi}_{th})^\top + \lambda_h^{(3)} \sum_{s=1, s\neq j}^{p} \left[ \sum_{t=1, t\neq s, t\neq j}^{p} (\overline{\boldsymbol{E}}_{n,jst}^{(3)} \times_2 \boldsymbol{\phi}_{sh} \times_3 \boldsymbol{\phi}_{th})^\top \right]}{(\lambda_h^{(2)})^2 \sum_{t=1, t\neq j}^{p} \boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th} + (\lambda_h^{(3)})^2 \sum_{s=1, s\neq j}^{p} \left[ \sum_{t=1, t\neq s, t\neq j}^{p} (\boldsymbol{\phi}_{sh}^\top \boldsymbol{\phi}_{sh})(\boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th}) \right]}, \tag{3.19}$$

where $\overline{\boldsymbol{E}}_{n,jst}^{(3)} = \boldsymbol{E}_{n,jst}^{(3)} - \sum_{h'\neq h} \lambda_{h'}^{(3)} \boldsymbol{\phi}_{jh'} \circ \boldsymbol{\phi}_{sh'} \circ \boldsymbol{\phi}_{th'}$. $\lambda_h^{(3)}$ is the $h$th diagonal entry of $\boldsymbol{\Lambda}^{(3)}$. The derivations can be found in Appendix B.5. After updating $\boldsymbol{\phi}_{jh}$ using the above

equations, we retract $\boldsymbol{\phi}_{jh}$ to its probability simplex when $y_{ij}$ is a categorical variable. We use the difference of the objective function between two iterations divided by the dimension of the moment vector to determine convergence. In particular, we stop iterations when this value is smaller than $1 \times 10^{-5}$.

After an initial consistent estimate of $\boldsymbol{\Phi}$ is found, we calculate the asymptotic covariance matrix of moment functions $\boldsymbol{S}_n^{(\cdot)}$ and define a new weight matrix $\boldsymbol{A}_n^{(\cdot)} = (\boldsymbol{S}_n^{(\cdot)})^{-1}$ for a second stage GMM estimation. The form of $\boldsymbol{S}_n^{(\cdot)}$ can be derived analytically, and we provide the results in the Appendix B.6. In our implementation, the calculation of $\boldsymbol{A}_n^{(\cdot)}$ requires the inversion of a full-rank dense matrix $\boldsymbol{S}_n^{(\cdot)}$ with dimension scaling as $O(p^2d^2)$ for $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$ and $O(p^3d^3)$ for $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$. In addition, when including the off-diagonal entries in the weight matrix, the updating rules become intrinsically complicated. In practice, we only extract the diagonal elements of $\boldsymbol{S}_n^{(\cdot)}$ and let $\boldsymbol{A}_n^{(\cdot)} = 1/\text{diag}[(\boldsymbol{S}_n^{(\cdot)})]$ in the second stage estimation. This approximation has been used in previous GMM implementations (Jöreskog and Sörbom, 1987). The gradient descent update equations can be found by slight modification of (3.18) and (3.19) with weights included.

Note that the moment functions do not solve the identifiability problems with respect to $\boldsymbol{\Phi}$. When $\boldsymbol{\alpha}$ is a constant vector, any permutation $\tau$ of $1, \ldots, k$ with $\boldsymbol{\Phi}_j(\tau) = (\boldsymbol{\phi}_{j\tau(1)}, \ldots, \boldsymbol{\phi}_{j\tau(k)})$ for all $j = 1, \ldots, p$ satisfies the moment condition. This problem is inherited from the label switching problem in mixture models. A similar situation occurs when there are ties in $\boldsymbol{\alpha}$ and the permutation is restricted to each tie. However, in real world applications, a minimizer of the quadratic function is generally sufficient to find a parameter estimate that is close to a unique configuration of the true parameter.

*Properties of parameter estimates*

We use GMM asymptotic theory to show that parameter estimate in MELD is consistent. We assume the following regularity conditions on $\boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ and the parameter space $\boldsymbol{\Theta}$.

**Assumption 3.1** (Regularity conditions (Assumption 3.2, 3.9 and 3.10. (Hall, 2005))). *1)* $\boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ *is continuous on* $\boldsymbol{\Theta}$ *for all* $\boldsymbol{y}_i \in \mathcal{Y}$*; 2)* $\mathbb{E}[\boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})] < \infty$ *and continuous for* $\boldsymbol{\Phi} \in \boldsymbol{\Theta}$*; 3)* $\boldsymbol{\Theta}$ *is compact and* $\mathbb{E}[sup_{\boldsymbol{\Phi}\in\boldsymbol{\Theta}}||\boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})||] < \infty$.

*Remark* 3.1. Conditions 1) and 2) are satisfied in MELD. Condition 3) is also satisfied, noting that $\boldsymbol{\phi}_{jh} \in \Delta^{d_j-1} \subset \mathbb{R}^{d_j}$ is compact for categorical variables. For a non-categorical variable, we could restrict our parameter space to a large compact domain such as closed intervals across the real line without sacrificing practical performance.

With these conditions, we further assume that the weight matrix $\boldsymbol{A}_n^{(\cdot)}$ converges to a positive definite matrix $\boldsymbol{A}^{(\cdot)}$ in probability. We define the population analogs of the quadratic functions as

$$Q_0^{(2)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(2)}) = \mathbb{E}[\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top \boldsymbol{A}^{(2)} \mathbb{E}[\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})], \tag{3.20}$$

$$Q_0^{(3)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(3)}) = \mathbb{E}[\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^\top \boldsymbol{A}^{(3)} \mathbb{E}[\boldsymbol{f}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]. \tag{3.21}$$

We have the following lemma showing the uniform convergence of $Q_n^{(\cdot)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(\cdot)})$.

**Lemma 3.1** (Uniform convergence (Lemma 3.1 (Hall, 2005))). *Under regularity conditions in Assumption 3.1,*

$$\sup_{\boldsymbol{\Phi}\in\boldsymbol{\Theta}} |Q_n^{(2)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(2)}) - Q_0^{(2)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(2)})| \xrightarrow{p} 0 \qquad \sup_{\boldsymbol{\Phi}\in\boldsymbol{\Theta}} |Q_n^{(3)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(3)}) - Q_0^{(3)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(3)})| \xrightarrow{p} 0,$$

**Theorem 3.2** (Consistency). *Under the same conditions in Lemma 3.1, the estimator* $\widehat{\boldsymbol{\Phi}}^{(2)}$ *that minimizes* $Q_n^{(2)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(2)})$ *converges to the true parameter* $\boldsymbol{\Phi}_0$ *in probability. A similar result holds for* $\widehat{\boldsymbol{\Phi}}^{(3)}$ *that minimizes* $Q_n^{(3)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(3)})$.

The proof can be found in Appendix B.2. Briefly, following Hall (2005), we first show that $\widehat{\boldsymbol{\Phi}}^{(\cdot)}$ minimizes $Q_0^{(\cdot)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(\cdot)})$ with probability one as $n \to \infty$. Then the theorem can be proved.

The asymptotic normality of $\widehat{\boldsymbol{\Phi}}^{(2)}$ and $\widehat{\boldsymbol{\Phi}}^{(3)}$ can also be established by assuming the following conditions on $\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}$.

**Assumption 3.2** (Conditions on $\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}$ (Assumptions 3.5, 3.12 and 3.13. (Hall, 2005)))**.** *1) $\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}$ exists and is continuous on $\boldsymbol{\Theta}$ for all $\boldsymbol{y}_i \in \mathcal{Y}$; 2) $\boldsymbol{\Phi}_0$ is an interior point of $\boldsymbol{\Theta}$; 3) $\mathbb{E}[\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}] = \boldsymbol{G}_0^{(\cdot)}(\boldsymbol{\Phi}) < \infty$; 4) $\boldsymbol{G}_0^{(\cdot)}(\boldsymbol{\Phi})$ is continuous on some neighborhood $N_\epsilon$ of $\boldsymbol{\Phi}_0$; 5) the sample estimate $\boldsymbol{G}_n^{(\cdot)}(\boldsymbol{\Phi})$ uniformly converges to $\boldsymbol{G}_0^{(\cdot)}(\boldsymbol{\Phi})$.*

*Remark* 3.2. We derive $\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}$ in Appendix B.4. Conditions 1, 3, and 4 are satisfied in MELD. Condition 5 can be shown with continuousness of the derivative and the compactness of $\boldsymbol{\Theta}$.

**Theorem 3.3** (Asymptotic normality)**.** *With Assumptions 3.1 and 3.2, we have*

$$n^{1/2}\left( vec(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) - vec(\boldsymbol{\Phi}_0) \right) \xrightarrow{p} N\left( \boldsymbol{0}, \boldsymbol{M}^{(\cdot)} \boldsymbol{S}^{(\cdot)} (\boldsymbol{M}^{(\cdot)})^\top \right)$$

*with*

$$\boldsymbol{M}^{(\cdot)} = [(\boldsymbol{G}_0^{(\cdot)})^\top \boldsymbol{A}^{(\cdot)} \boldsymbol{G}_0^{(\cdot)}]^{-1} (\boldsymbol{G}_0^{(\cdot)})^\top \boldsymbol{A}^{(\cdot)},$$

where $\boldsymbol{G}_0^{(\cdot)} = \mathbb{E}[\partial \boldsymbol{f}^{(\cdot)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial \boldsymbol{\Phi}]|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_0}$ and $\boldsymbol{S}^{(\cdot)} = \lim_{n\to\infty} \mathrm{Var}[n^{1/2} \widehat{\boldsymbol{f}}_n^{(\cdot)}(\boldsymbol{\Phi}_0)]$. The proof can be found in Appendix B.3. The optimal estimator can be obtained so that the weight matrix $\boldsymbol{A}_n^{(\cdot)} \to \boldsymbol{A}^{(\cdot)} = (\boldsymbol{S}^{(\cdot)})^{-1}$ (Hansen, 1982).

*Recovering membership variables*

Given an estimate of the parameter $\widehat{\boldsymbol{\Phi}}$, we can calculate estimate of membership allocation $m_{ij}$ for each $y_{ij}$. For categorical variables, we find the $\{m_{ij}\}$ that maximize

$$\{\widehat{m}_{ij}\} = \underset{\{m_{ij}\}}{\mathrm{argmax}} \left( \prod_{j=1}^{p} \prod_{i=1}^{n} \prod_{h=1}^{k} \widehat{\phi}_{jhy_{ij}}^{\mathbf{1}(m_{ij}=h)} \right). \tag{3.22}$$

Here **1** is the indicator function. The solution can be calculated using

$$\widehat{m}_{ij} = \underset{h}{\text{argmax}}(\widehat{\phi}_{jhy_{ij}}) \text{ for } h = 1, \ldots, k.$$

For non-categorical variables, we find the $\{m_{ij}\}$ that minimizes the distance metric

$$\widehat{m}_{ij} = \underset{h}{\text{argmin}}\, \mathrm{D}(\widehat{\phi}_{jh}, y_{ij}) \text{ for } h = 1, \ldots, k. \tag{3.23}$$

The negative log density is a natural metric. Once the $\{m_{ij}\}$ are allocated, subject mixture proportion $\boldsymbol{x}_i$ can be estimated from those membership variables.

### *3.2.4 Model selection using goodness of fit tests*

We discuss how to choose the value of $k$ in this subsection using goodness of fit tests. With the optimal weight matrix $\boldsymbol{A}_n^{(\cdot)} \to (\boldsymbol{S}^{(\cdot)})^{-1}$, the values of the objective function can be used to construct tests similar to the classical trio tests in the maximum likelihood (ML) context: the Wald, Lagrange multiplier (score) and likelihood ratio tests (Newey and West, 1987). Under the null those test statistics asymptotically follow a chi-squared distribution. We could construct a sequence of test statistics under different values of $k$ to access the goodness of fit in MELD. However this approach requires the calculations of optimal weight matrices and needs large matrix inversions. We avoid this approach in following analysis. Alternatively we assess the goodness of fit using two methods, one is an information criterion based on complete likelihood, the integrated complete likelihood (ICL) (Biernacki et al., 2000), and one is a fitness index proposed by Bentler (1983). Both of the methods do not require calculation of optimal weight matrices.

The ICL method is similar to the classical Bayesian information criterion (BIC) which approximates the integrated likelihood (Schwarz, 1978). However ICL does not require integration of latent variables to get a marginal likelihood. Instead it is based on complete likelihood. Write the integrated likelihood of complete data

$(\boldsymbol{Y}, \boldsymbol{M})$ (Biernacki et al., 2000) in MELD

$$p(\boldsymbol{Y}, \boldsymbol{M}|k, \boldsymbol{\alpha}) = \int_{\boldsymbol{\Phi}} \left( \prod_{i=1}^{n} \int_{\boldsymbol{x}_i} p(\boldsymbol{y}_i, \boldsymbol{m}_i|\boldsymbol{\Phi}, \boldsymbol{x}_i) p(\boldsymbol{\Phi}, \boldsymbol{x}_i|k, \boldsymbol{\alpha}) \mathrm{d}\boldsymbol{x}_i \right) \mathrm{d}\boldsymbol{\Phi}$$

$$= \int_{\boldsymbol{\Phi}} p(\boldsymbol{Y}|\boldsymbol{M}, \boldsymbol{\Phi}) p(\boldsymbol{\Phi}|k) \mathrm{d}\boldsymbol{\Phi} \times \prod_{i=1}^{n} \int_{\boldsymbol{x}_i} p(\boldsymbol{m}_i|\boldsymbol{x}_i) p(\boldsymbol{x}_i|k, \boldsymbol{\alpha}) \mathrm{d}\boldsymbol{x}_i, \quad (3.24)$$

where $\boldsymbol{m}_i = (m_{1i}, \ldots, m_{pi})^\top$. The first term in the right hand side of (3.24) can be computed by assigning $\boldsymbol{\Phi}$ a prior distribution $p(\boldsymbol{\Phi}|k)$. When the data are all categorical, this term can be calculated in a closed form with a prior $\boldsymbol{\phi}_{jh} \sim \mathrm{Dir}(\boldsymbol{\beta}_j)$

$$\int_{\boldsymbol{\Phi}} p(\boldsymbol{Y}|\boldsymbol{M}, \boldsymbol{\Phi}) p(\boldsymbol{\Phi}|k) \mathrm{d}\boldsymbol{\Phi} = \prod_{j=1}^{p} \prod_{h=1}^{k} \frac{\prod_{c_j=1}^{d_j} \Gamma(o_{jhc_j} + \beta_{c_j})}{\Gamma(\sum_{c_j} o_{jhc_j} + \sum_{c_j} \beta_{c_j})} \frac{\Gamma(\sum_{c_j} \beta_{c_j})}{\prod_{c_j=1}^{d_j} \Gamma(\beta_{c_j})},$$

where $o_{jhc_j}$ is the number of subjects with $j$th variable taking $c_j$th category and belonging to $h$th component, and $\Gamma(\cdot)$ is the gamma function. When the closed form integration does not exist, we use BIC to approximate the logarithm of the term, generating

$$\log p(\boldsymbol{Y}|\boldsymbol{M}, k) \approx \max_{\boldsymbol{\Phi}} \log p(\boldsymbol{Y}|\boldsymbol{M}, \boldsymbol{\Phi}, k) - \frac{\nu}{2} \log(n).$$

Here $\nu$ is the number of free parameters in $\boldsymbol{\Phi}$. The second term in (3.24) has a closed form solution in MELD

$$\prod_{i=1}^{n} \int_{\boldsymbol{x}_i} p(\boldsymbol{m}_i|\boldsymbol{x}_i) p(\boldsymbol{x}_i|k, \boldsymbol{\alpha}) \mathrm{d}\boldsymbol{x}_i = \prod_{i=1}^{n} \left( \frac{\Gamma(\alpha_0)}{\prod_{h=1}^{k} \Gamma(\alpha_h)} \frac{\prod_{h=1}^{k} \Gamma(n_{ih} + \alpha_h)}{\Gamma(p + \alpha_0)} \right),$$

where $n_{ih}$ is the number of variables belonging to component $h$ in subject $i$. In our GMM framework we plugin our GMM estimator $\widehat{\boldsymbol{\Phi}}$ and the recovered membership variable $\widehat{\boldsymbol{M}}$ to get following ICL criterion

$$\mathrm{ICL} = -2 \log(\boldsymbol{Y}|\widehat{\boldsymbol{M}}, \widehat{\boldsymbol{\Phi}}, k) + \nu \log(n) - 2K(\widehat{\boldsymbol{M}}), \quad (3.25)$$

where

$$K(\widehat{\boldsymbol{M}}) = n \log \Gamma(\alpha_0) + \sum_{i=1}^{n} \sum_{h=1}^{k} \log \Gamma(\widehat{n}_{ih} + \alpha_h) - n \sum_{h=1}^{k} \log \Gamma(\alpha_h) - n \log \Gamma(p + \alpha_0).$$

The ICL can be viewed as follows. The first two terms in the right hand side of (3.25) is a penalized likelihood assuming the membership variables are known. The last term introduces an additional penalty to penalize the fitness of the membership variables. It has been shown to generate good model selection behavior in mixture of Gaussians (Steele and Raftery, 2010). Small values of ICL suggest good fit.

As a second method, we use the fitness index (FI) proposed by Bentler (1983). The FI is based on the value of the objective function evaluated at parameter estimate and it is defined as

$$\text{FI} = 1 - \frac{Q_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}, \boldsymbol{A}_n^{(\cdot)})}{(\boldsymbol{e}_n^{(\cdot)})^\top \boldsymbol{A}_n^{(\cdot)} \boldsymbol{e}_n^{(\cdot)}}, \tag{3.26}$$

where $\boldsymbol{e}_n^{(\cdot)}$ is the vectorization of $\boldsymbol{E}_{n,jt}^{(2)}$ for $j < t$ or $\boldsymbol{E}_{n,jst}^{(3)}$ for $j < s < t$. This fitness index is for any weight matrix $\boldsymbol{A}_n^{(\cdot)}$. It can be viewed as a normalized objective function and its value is smaller than one. Large values of FI suggest good fit.

In our simulations, which will be shown in detail in Section 3.3, we find the ICL criterion favors small model greatly. Therefore only the results of using FI are shown.

### *3.2.5 Computational complexity*

The calculation of the moment statistics and the weight matrices are performed once. The complexity of calculating $\boldsymbol{E}_{n,jt}^{(2)}$ for all $j, t$ requires accessing all the data, and has complexity $O(p^2 n)$. Similarly, calculating $\boldsymbol{E}_{n,jst}^{(3)}$ for all $j, s, t$ requires $O(p^3 n)$ complexity. The calculation of the optimal weight matrix for $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$ requires $O(p^2 k)$ and for $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$ it requires $O(p^3 k)$. We now discuss the computational complexity of MELD per iteration. For simplicity, we assume that the number of levels of $y_{ij}$ are fixed across dimensions. For the first version of our GMM with objective function $Q_n^{(2)}(\boldsymbol{\Phi}, \boldsymbol{I})$, each Newton-Raphson update has complexity $O(p)$. Thus, the total complexity for each iteration is $O(p^2 k)$. For our second version of GMM with

objective function $Q_n^{(3)}(\boldsymbol{\Phi}, \boldsymbol{I})$, each Newton-Raphson update takes $O(p^2)$, thus the overall complexity is $O(p^3 k)$.

We now analyze the number of moment functions and number of free parameters in MELD. Assuming the number of levels $d$ of $y_{ij}$ is the same across dimensions, the number of free parameters in $\boldsymbol{\Phi}$ is $pk(d-1)$. The number of moment functions using $Q_n^{(2)}(\boldsymbol{\Phi}, \boldsymbol{I})$ is $p(p-1)d^2/2$. When $p(p-1)d^2/2 > pk(d-1)$, the moment vector $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$ provide sufficient restrictions satisfying its expected first derivative $G_0^{(2)}(\boldsymbol{\Phi})$ of full column rank, and the GMM estimator is consistent. For example, when $d = 5$ and $k = 4$, we need $p > 2$ to provide consistent estimators. When we include third moment tensors in the moment vector $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$, the number of moment functions becomes $[p^3 - 3p(p-1) - p]d^3/6 + p(p-1)d^2/2$, which scales with $O(p^3)$. Our second version of GMM includes additional moment restrictions, and hence is more efficient compared to our first version of GMM estimators, as shown in our simulation studies. However, the tradeoff is increased computational complexity and decreased robustness to violations of model assumptions.

One notable feature of our GMM algorithms is that, after passing through all of the samples and calculating certain moment statistics, parameter estimations only depend on the two quantities $\boldsymbol{E}_{n,jt}^{(2)}$ and $\boldsymbol{E}_{n,jst}^{(3)}$. This feature allows our method to perform fast parameter estimations when applied to modern data sets with large sample sizes because we only need to scan all of the samples once.

## 3.3 Simulations

In this section, we evaluate the accuracy and run time of MELD in simulations with both categorical and mixed data types. We use two stage estimations described in previous sections. In the first stage an identity weight matrix is used. In second stage we set $\boldsymbol{A}_n^{(\cdot)} = 1/\text{diag}[(\boldsymbol{S}_n^{(\cdot)})]$. For notation convenience we suppress the weight

matrix $\boldsymbol{A}^{(\cdot)}$ and subscript $n$ in the objective functions $Q_n^{(\cdot)}(\boldsymbol{\Phi}, \boldsymbol{A}^{(\cdot)})$.

### *3.3.1 Categorical data*

For simulations with categorical data, we consider two settings: 1) a low dimensional setting ($p = 20$) so that both second and third order moment functions may be efficiently calculated; 2) a moderate dimensional setting ($p = 100$) to estimate population structure of genomic data under Hardy-Weinberg equilibrium (HWE) and non-HWE.

*Low dimensional simulations* We simulate $p = 20$ categorical variables in this setting, each with $d = 4$ levels. We set the number of components to $k = 3$ and generate $\boldsymbol{\phi}_{jh}$ from $\text{Dir}(0.5, 0.5, 0.5, 0.5)$ with $h = 1, \ldots, k$. $\boldsymbol{\alpha}$ is set to $(0.1, 0.1, 0.1)^\top$. We draw $n = \{50, 100, 200, 500, 1,000\}$ samples from the generative model in equation (3.2) with $g_j(\boldsymbol{\phi}_{jh})$ a multinomial distribution. For each value of $n$, we generate ten independent data sets. We run MELD for different values of $k = \{1, \ldots, 5\}$. The FI criterion consistently chooses the correct number of latent components $k$ in first stage estimation with both $Q^{(2)}(\boldsymbol{\Phi})$ and $Q^{(3)}(\boldsymbol{\Phi})$ (Table 3.1). For second stage, FI does not perform well. It overestimates the number of components with $Q^{(2)}(\boldsymbol{\Phi})$ and simulations under $n = 50, 100$ with $Q^{(3)}(\boldsymbol{\Phi})$. For larger sample sizes, FI underestimates the number of comoponents with $Q^{(3)}(\boldsymbol{\Phi})$. The trajectories of the objective functions under different values of $k$ are shown in Figure 3.1. The convergence of parameter estimations on the ten simulated data sets under $n = 1,000$ and $k = 3$ is shown in Figure 3.2. MELD converged in about 25 iterations with $Q^{(2)}(\boldsymbol{\Phi})$ and in about 10 iterations with $Q^{(3)}(\boldsymbol{\Phi})$.

We use mean squared error (MSE) between the estimated mean parameters of $y_{ij}$'s and their true mean parameters to evaluate the precision of estimation. The estimated mean parameters of $y_{ij}$'s are calculated by recovering their membership

Table 3.1: Goodness of fit tests using the fitness index (FI) in low dimensional categorical simulation. Larger values of FI indicate better fit, with the maximum at one. Results shown are based on ten simulated data sets for each value of $n$. Standard deviations of FI are provided in parentheses.

| n | $k$ | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage | $Q^{(3)}(\mathbf{\Phi})$ 1st stage | $Q^{(3)}(\mathbf{\Phi})$ 2nd stage |
|---|---|---|---|---|---|
| | 1 | 0.824(0.020) | -1.865(4.950) | 0.585(0.031) | -27.987(40.128) |
| | 2 | 0.908(0.011) | -1.285(2.630) | 0.745(0.032) | -27.240(54.415) |
| 50 | **3** | **0.930(0.004)** | 0.570(0.044) | **0.775(0.022)** | -27.713(42.607) |
| | 4 | 0.901(0.010) | 0.588(0.032) | 0.735(0.023) | 0.254(0.090) |
| | 5 | 0.795(0.033) | **0.686(0.021)** | 0.653(0.024) | **0.305(0.091)** |
| | 1 | 0.860(0.012) | -0.521(2.142) | 0.651(0.021) | -0.031(0.703) |
| | 2 | 0.930(0.011) | 0.282(0.644) | 0.795(0.030) | -4.457(8.924) |
| 100 | **3** | **0.960(0.005)** | 0.677(0.044) | **0.851(0.009)** | -4.868(11.889) |
| | 4 | 0.942(0.012) | 0.691(0.042) | 0.822(0.010) | 0.225(0.019) |
| | 5 | 0.863(0.046) | **0.782(0.022)** | 0.768(0.044) | **0.232(0.080)** |
| | 1 | 0.869(0.012) | 0.679(0.060) | 0.682(0.021) | 0.298(0.070) |
| | 2 | 0.940(0.007) | 0.699(0.047) | 0.838(0.014) | **0.306(0.054)** |
| 200 | **3** | **0.980(0.001)** | 0.761(0.061) | **0.919(0.004)** | 0.278(0.049) |
| | 4 | 0.967(0.006) | 0.780(0.019) | 0.891(0.008) | 0.287(0.050) |
| | 5 | 0.911(0.017) | **0.824(0.008)** | 0.864(0.015) | 0.286(0.042) |
| | 1 | 0.882(0.007) | 0.783(0.022) | 0.713(0.012) | 0.414(0.080) |
| | 2 | 0.948(0.006) | 0.799(0.019) | 0.870(0.013) | **0.427(0.080)** |
| 500 | **3** | **0.992(<0.001)** | 0.884(0.024) | **0.966(0.001)** | 0.388(0.073) |
| | 4 | 0.983(0.004) | 0.874(0.026) | 0.938(0.005) | 0.365(0.065) |
| | 5 | 0.937(0.014) | **0.894(0.006)** | 0.921(0.007) | 0.353(0.042) |
| | 1 | 0.888(0.003) | 0.828(0.006) | 0.729(0.008) | 0.571(0.017) |
| | 2 | 0.951(0.003) | 0.855(0.009) | 0.881(0.008) | **0.615(0.030)** |
| 1,000 | **3** | **0.996(<0.001)** | **0.950(0.005)** | **0.982(0.001)** | 0.609(0.031) |
| | 4 | 0.989(0.002) | 0.938(0.004) | 0.961(0.003) | 0.579(0.030) |
| | 5 | 0.953(0.010) | 0.932(0.006) | 0.951(0.006) | 0.550(0.034) |

variables using (3.22). We compare our method with the simplex factor model (SFM) (Bhattacharya and Dunson, 2012) and latent Dirichlet allocation (LDA) (Blei et al., 2003) on the ten simulated data sets. For the SFM, we run $10,000$ steps of MCMC with fixed $k$ and a burn-in of $5,000$ iterations. Posterior thinned samples are collected by keeping one posterior draw after every 50 steps. From the posterior samples we calculate posterior mean as our estimate. For the LDA model, we use the `lda` package in R (Chang, 2012) with collapsed Gibbs sampling. We use the same number of MCMC iterations and burn-in iterations as with SFM. The Dirichlet parameter for mixture proportions is set to $\alpha = 0.1$ and the Dirichlet parameter for topic distributions is set to $\beta = 0.5$.

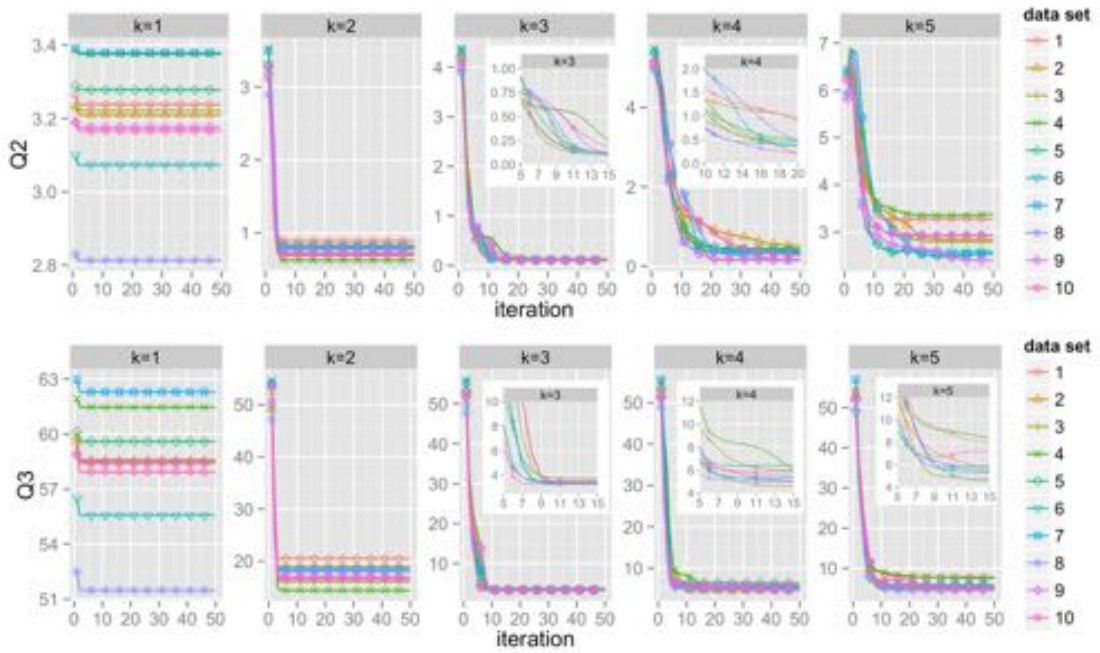


FIGURE 3.1: Parameter estimation with MELD in the low dimensional categorical simulations. Parameters are estimated with $Q^{(2)}(\mathbf{\Phi})$ and $Q^{(3)}(\mathbf{\Phi})$ on 10 simulated data sets with $n = 1,000$ and $k = 3$. Results shown are for first stage estimation.

MSE's with different values of $k$ are shown in Figure 3.3 (values are shown in Table B.1) and the running times of different methods are reported in Table 3.2. MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation has the most accurate parameter estimation and fastest running speed in most cases. The second stage of MELD $Q^{(3)}(\mathbf{\Phi})$ does not perform well with small values of $n$ (i.e., $n = 50$), but estimation accuracy is better when $n$ is larger. SFM has comparable MSE's when $n$ is not large. However when $n = 500$ and $1,000$, MELD outperforms SFM.

We further evaluate performance in the presence of contamination. For each simulated data set, we randomly replace a proportion of observations (4% and 10%) with draws from a discrete uniform distribution. The MSE's under different values of $k$ are shown in Figure 3.3. With 4% contamination, MELD has the most accurate parameter estimation in almost all cases. MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation performs the best, followed by MELD $Q^{(3)}(\mathbf{\Phi})$ with first stage estimation. The

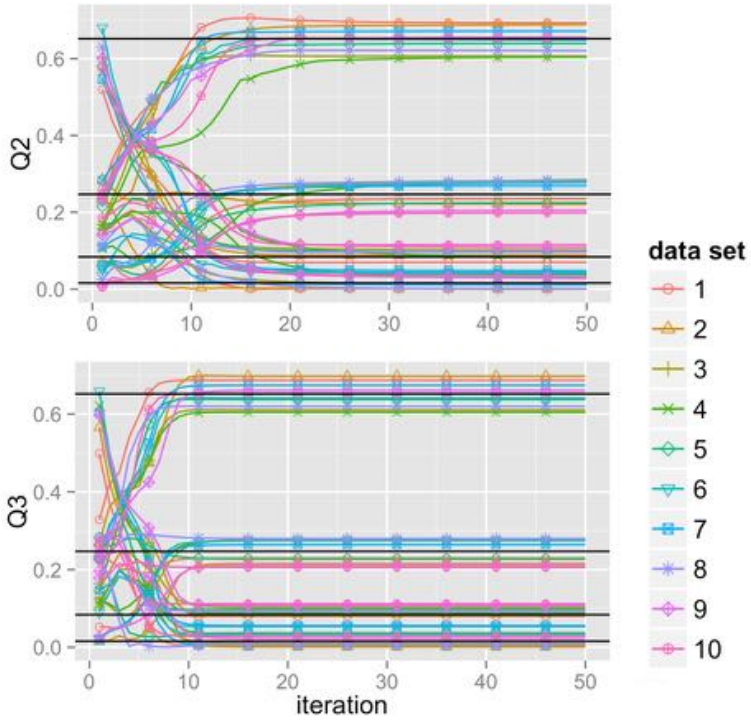


FIGURE 3.2: Convergence of parameter estimation with MELD in the low dimensional categorical simulation. Results plotted are for ten simulated data sets with $n = 1,000$ and $k = 3$ in the first stage estimation. True parameter values are shown as dark lines.

MSE's of SFM increase. When we increase contamination to 10%, MELD has the most accurate MSE in all cases. MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation performs best, followed by MELD $Q^{(2)}(\mathbf{\Phi})$ with second stage estimation. MELD $Q^{(2)}(\mathbf{\Phi})$ consistently performs better than $Q^{(3)}(\mathbf{\Phi})$, suggesting the robustness of using lower order moments in parameter estimation.

*Inference of population structure* In this setting, we simulate genotype data from an admixed population of a diploid species. Each observation $y_{ij}$ is a categorical variable with three levels $\{0, 1, 2\}$ representing the genotype at locus $j$ in subject $i$. The number represents the number of copes of the reference alleles on the two copies of the chromosome. We first assume that the genotype distribution is in

Table 3.2: Comparison of total running time in seconds between MELD, SFM, and LDA in categorical simulation. Methods are run on a Intel(R) Core i7-3770 CPU at 3.40GHz machine. MELD represents the averaged running time for the first stage estimation on the ten simulated data sets for each value of $n$. Average number of iterations to convergence are in parentheses. The second stage estimation requires one or two additional iterations starting from the parameters estimated in the first stage.

| $n$ | $k$ | MELD $Q^{(2)}(\mathbf{\Phi})$ | MELD $Q^{(3)}(\mathbf{\Phi})$ | SFM | LDA |
|---|---|---|---|---|---|
| 50 | 1 | **0.026(2)** | 0.489(2) | 9.199 | 0.104 |
| | 2 | 0.221(9) | 2.083(5) | 27.058 | **0.156** |
| | 3 | 0.410(11) | 4.645(7) | 44.758 | **0.207** |
| | 4 | 0.681(13) | 6.765(7) | 54.122 | **0.254** |
| | 5 | 0.921(14) | 9.495(8) | 66.367 | **0.301** |
| 100 | 1 | **0.022(2)** | 0.418(2) | 7.203 | 0.208 |
| | 2 | **0.192(7)** | 2.333(5) | 25.980 | 0.309 |
| | 3 | **0.364(9)** | 3.942(6) | 46.337 | 0.407 |
| | 4 | 0.588(11) | 5.577(6) | 64.178 | **0.502** |
| | 5 | 0.934(15) | 7.795(7) | 82.907 | **0.593** |
| 200 | 1 | **0.028(2)** | 0.470(2) | 9.463 | 0.416 |
| | 2 | **0.227(9)** | 2.185(5) | 31.547 | 0.617 |
| | 3 | **0.400(10)** | 3.914(6) | 53.503 | 0.811 |
| | 4 | **0.658(13)** | 5.869(6) | 63.947 | 1.001 |
| | 5 | **0.983(15)** | 8.498(8) | 75.846 | 1.183 |
| 500 | 1 | **0.017(1)** | 0.409(2) | 15.611 | 1.060 |
| | 2 | **0.276(11)** | 3.222(7) | 41.810 | 1.566 |
| | 3 | **0.397(10)** | 4.107(6) | 55.475 | 2.048 |
| | 4 | **0.578(11)** | 5.083(6) | 87.932 | 2.512 |
| | 5 | **0.954(15)** | 7.070(6) | 93.092 | 2.967 |
| 1,000 | 1 | **0.016(1)** | 0.413(2) | 25.923 | 2.148 |
| | 2 | **0.289(11)** | 3.287(7) | 57.390 | 3.151 |
| | 3 | **0.371(10)** | 3.901(6) | 89.434 | 4.108 |
| | 4 | **0.614(12)** | 5.065(6) | 111.845 | 5.043 |
| | 5 | **1.000(16)** | 7.047(6) | 146.234 | 5.946 |

Hardy-Weinberg equilibrium (HWE), which means that the two copies of the allele are independently inherited from the two parents with a common reference allele frequency. Let the reference frequency of an allele in the population be $\pi^{(a)}$. Then the probability of observing genotypes 0, 1, and 2 are $(1-\pi^{(a)})^2$, $2(1-\pi^{(a)})\pi^{(a)}$ and $(\pi^{(a)})^2$, respectively. We simulate genotype data for 100 loci for $n = \{50, 100, 200, 500, 1,000\}$ subjects. The reference allele frequencies for each population are drawn uniformly from $(0.05, 0.95)$. Then we relax the HWE assumption and assume the genotype distribution to follow a multinomial distribution with three outcomes, the non-HWE

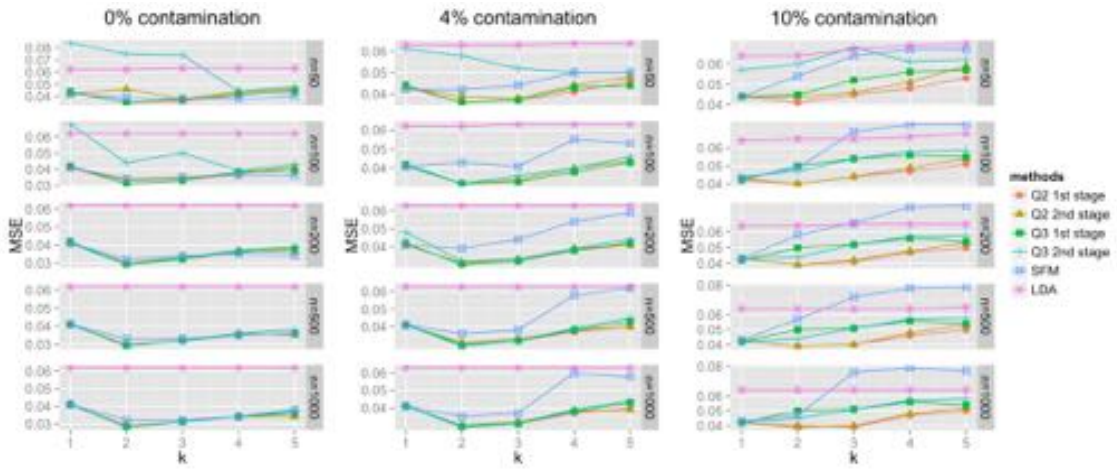


FIGURE 3.3: Comparison of mean squared error (MSE) of estimated parameters in categorical simulations. For SFM and LDA, posterior means of parameters are calculated using 100 posterior draws on each of the ten simulated data sets. The values of the MSE's and their standard deviations are in Table B.1, B.2 and B.3.

case. The non-HWE can be thought as the case where the reference allele has different frequencies on the two copies of the chromosome. Let $\pi_1^{(a)}$ and $\pi_2^{(a)}$ be their frequencies. Then the probability of observing genotypes 0, 1, and 2 are $(1-\pi_1^{(a)})(1-\pi_2^{(a)})$, $(1-\pi_1^{(a)})\pi_2^{(a)} + (1-\pi_2^{(a)})\pi_1^{(a)} = \pi_1^{(a)} + \pi_2^{(a)} - 2\pi_1^{(a)}\pi_2^{(a)}$ and $\pi_1^{(a)}\pi_2^{(a)}$, respectively. We generate the multinomial parameters of genotype distributions by drawing from $\text{Dir}(0.5, 0.5, 0.5)$, from which $\pi_1^{(a)}$ and $\pi_2^{(a)}$ could be determined. The number of populations is set to $k = 4$, with mixture proportions $\boldsymbol{x}_i$ drawn from $\text{Dir}(0.1, 0.1, 0.1, 0.1)$. For each value of $n$, ten data sets are generated. We apply MELD with $Q^{(2)}(\boldsymbol{\Phi})$ to these simulated data setting the number of latent populations to $k = \{1, \ldots, 5\}$.

Evaluating the goodness of fit across $k$'s, FI chooses the correct number of latent populations in most cases (Table 3.3). We compare MELD with SFM, LDA, and two state-of-the-art methods in population genetics: ADMIXTURE (ADM) (Alexander et al., 2009) and Logistic factor analysis (LFA) (Hao et al., 2013). ADM uses a fast likelihood-based approach to estimate the allele frequencies in each population (a $p \times k$

Table 3.3: Goodness of fit tests using the fitness index (FI) in simulation of inference of population structure. Larger values of FI indicate better fit, with the maximum at one. Results shown are based on ten simulated data sets for each value of $n$. Standard deviations of FI are provided in parentheses.

| | | HWE | | non-HWE | |
|---|---|---|---|---|---|
| n | $k$ | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage |
| 50 | 1 | 0.921(0.004) | 0.816(0.013) | 0.916(0.008) | 0.614(0.116) |
| | 2 | 0.929(0.003) | 0.820(0.014) | 0.934(0.007) | 0.662(0.062) |
| | 3 | **0.932(0.002)** | 0.827(0.009) | 0.949(0.005) | 0.734(0.017) |
| | **4** | 0.931(0.004) | **0.834(0.009)** | **0.953(0.004)** | 0.776(0.004) |
| | 5 | 0.919(0.003) | 0.831(0.006) | 0.925(0.009) | **0.795(0.007)** |
| 100 | 1 | 0.948(0.002) | 0.886(0.007) | 0.934(0.004) | 0.801(0.011) |
| | 2 | 0.955(0.002) | 0.890(0.008) | 0.951(0.004) | 0.810(0.016) |
| | 3 | 0.960(0.001) | 0.898(0.007) | 0.965(0.004) | 0.833(0.016) |
| | **4** | **0.961(0.002)** | **0.912(0.003)** | **0.975(0.002)** | 0.872(0.011) |
| | 5 | 0.951(0.003) | 0.910(0.002) | 0.960(0.006) | **0.877(0.004)** |
| 200 | 1 | 0.963(0.002) | 0.926(0.002) | 0.944(0.002) | 0.856(0.004) |
| | 2 | 0.971(0.001) | 0.935(0.002) | 0.959(0.002) | 0.870(0.004) |
| | 3 | 0.976(0.001) | 0.944(0.002) | 0.974(0.002) | 0.895(0.005) |
| | **4** | **0.978(0.001)** | **0.956(0.001)** | **0.988(0.001)** | **0.933(0.005)** |
| | 5 | 0.971(0.001) | 0.953(0.001) | 0.974(0.006) | 0.930(0.004) |
| 500 | 1 | 0.973(0.001) | 0.950(0.002) | 0.950(0.001) | 0.890(0.002) |
| | 2 | 0.980(0.001) | 0.960(0.001) | 0.962(0.002) | 0.905(0.005) |
| | 3 | 0.986(0.001) | 0.971(0.001) | 0.977(0.002) | 0.927(0.010) |
| | **4** | **0.990(0.002)** | **0.981(0.001)** | **0.995(<0.001)** | **0.973(0.001)** |
| | 5 | 0.983(0.001) | 0.979(0.001) | 0.986(0.003) | 0.966(0.002) |
| 1,000 | 1 | 0.976(0.001) | 0.957(0.001) | 0.953(0.001) | 0.901(0.002) |
| | 2 | 0.983(0.001) | 0.967(0.001) | 0.965(0.001) | 0.920(0.004) |
| | 3 | 0.989(0.001) | 0.979(0.001) | 0.979(0.002) | 0.939(0.009) |
| | **4** | **0.995(0.001)** | **0.990(<0.001)** | **0.998(<0.001)** | **0.986(0.001)** |
| | 5 | 0.989(0.001) | 0.989(<0.001) | 0.989(0.002) | 0.979(0.001) |

matrix), and mixture proportions for each subject (a $k \times n$ matrix) in an admixture model (Pritchard et al., 2000b). Then, the allele frequency at locus $j$ for subject $i$ can be calculated by multiplying the two matrices. The LFA model assumes the logit of allele frequencies factorize to a product of two matrices. Parameter estimation is performed using a modified PCA algorithm. Both models assume that the genotypes are in HWE. We calculate the MSE's between the mean parameters of the genotype distributions of $y_{ij}$'s and their true mean parameters. MELD $Q^{(2)}(\mathbf{\Phi})$ in the first stage estimation outperforms SFM, LDA and ADM in most cases. The LFA has the most accurate parameter estimation under the correct value of $k$. LDA has the

most stable performance across different values of $k$ (Table B.4 and B.5). In terms of speed, ADM has the fastest running time, followed by LDA and LFA. MELD is faster than SFM.

### *3.3.2 Mixed data types*

For simulations with mixed data we consider two settings: 1) a genetic association study where DNA sequence variations influence a quantitative trait; 2) a general setting with categorical, Gaussian, and Poisson variables.

*Genetic quantitative trait association study* We consider a simulation setting mimicking applications in which DNA sequence variations influence a quantitative trait. We generate a sequence of nucleotides $\{A, C, G, T\}$ at 50 genetic loci along with a continuous or integer-valued trait, leading to $p = 51$ variables. We set $k = 2$ latent components and simulate $n = 1,000$ individuals, with the first 500 from the first component and the last 500 from the second component. We choose eight loci $J = \{2, 4, 12, 14, 32, 34, 42, 44\}$ to be associated with the trait. Their multinomial parameters for each of the two components are randomly drawn from $\text{Dir}(0.5, 0.5, 0.5, 0.5)$. The distributions for nucleotides in other loci are set to $\text{Multi}(0.25, 0.25, 0.25, 0.25)$. Continuous traits are drawn from $\text{N}(-3, 1)$ and $\text{N}(3, 1)$, while count traits are drawn from Poisson(5) and Poisson(10), respectively for the two components. Ten data sets are simulated from the generative model of MELD in (3.2). To assess robustness, we add contamination (e.g., through genotyping errors) by randomly replacing 4%, 10% and 20% of the nucleotides with values uniformly generated from $\{A, C, G, T\}$.

We run MELD with first stage estimation of $Q^{(2)}(\mathbf{\Phi})$. We choose the number of components $k = \{1, \ldots, 5\}$. The fitness test indicates that FI chooses the correct value of $k$ on all ten data sets (Table 3.4). For each genomic locus, we calculate its marginal frequency according to the simulated data, and then we compute the aver-

Table 3.4: Goodness of fit test using the fitness index (FI) in a genetic association simulation. Values closer to one indicate a better fit. Values shown are the results of applying MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation to ten simulated data sets. Standard deviation of FI across the ten simulations are in parentheses.

| $k$ | 1 | **2** | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Gaussian trait | 0.982(<0.001) | **0.984(<0.001)** | 0.977(0.003) | 0.947(0.004) | 0.915(0.005) |
| Poisson trait | 0.997(<0.001) | **0.999(<0.001)** | 0.998(0.001) | 0.998(<0.001) | 0.989(0.003) |

aged KL distance between the estimated component distributions and the marginal frequency as follows

$$\text{aveKL}(y_{ij}) = \frac{1}{k}\sum_{h=1}^{k}\sum_{c_j=1}^{d_j}\Pr(y_{ij}=c_j|m_{ij}=h)\log\left(\frac{\Pr(y_{ij}=c_j|m_{ij}=h)}{\Pr(y_{ij}=c_j)}\right). \quad (3.27)$$

A smaller averaged KL distance suggests that the component distributions are closer to the marginal distribution, implying that the locus frequency is not differentiated across components. The set $J$ correspond exactly to the eight loci with largest averaged KL distance (Table 3.5).

We compare MELD with the Bayesian copula factor model (Murray et al., 2013), which estimates a correlation matrix $\boldsymbol{C}$ between variables. From this estimate, we compute partial correlations between the response variable (trait) and each genetic locus (Hoff, 2007). We run MCMC for the Bayesian copula factor model $10,000$ iterations with the correct value for $k$ and a burn-in of $5,000$ iterations. Posterior samples are collected every 50 iterations. We then select genomic locations for which their 95% credible intervals of the partial correlation does not include zero. The resulting loci are shown in Table 3.5. The Bayesian copula factor model selects nucleotides that are not in $J$ and misses locus 32 in most cases.

Table 3.5: Quantitative trait association simulation with 50 nucleotides and one response. Nucleotides not in $J = \{2, 4, 12, 14, 32, 34, 42, 44\}$ are labeled by an underline and missing nucleotides are crossed out. Results shown are for one of the ten simulated data sets. The complete results can be found in Table B.6 and B.7.

| Response | Contamination | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | Bayesian copula factor model |
|---|---|---|---|
| Gaussian | 0% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{18}, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 4% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{18}, \underline{27}, \cancel{32}, 34, 42, 44, \underline{45}\}$ |
| | 10% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \underline{27}, \cancel{32}, 34, 42, 44, \underline{49}, \underline{50}\}$ |
| | 20% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{9}, \underline{11}, 12, \cancel{14}, \underline{20}, \cancel{32}, 34, 42, 44\}$ |
| Poisson | 0% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 4% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 10% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \underline{16}, 32, 34, 42, 44\}$ |
| | 20% | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, \cancel{14}, 32, 34, \underline{35}, 42, 44\}$ |

*Mixed variables with categorical, Gaussian, and Poisson distributions* Next, we simulate data with $p = 100$ mixed variables. The first 95 variables are categorical, each with $d = 4$ levels. We simulate two additional Gaussian variables and three additional Poisson variables. We set the number of components to $k = 2$. The multinomial parameters for variables in $J_1 = \{1, 2, 3, 4, 5\}$ in the first component and variables in $J_2 = \{4, 5, 6, 7, 8\}$ in the second component are drawn from $\text{Dir}(0.5, 0.5, 0.5, 0.5)$. The distributions for the rest categorical variables in the two components are set to $\text{Multi}(0.25, 0.25, 0.25, 0.25)$. The Gaussian variables are drawn from $\text{N}(-3, 1)$ in component one and $\text{N}(3, 1)$ in component two. The Poisson variables are drawn from Poisson(5) and Poisson(10) in components one and two respectively. The mixture proportions Dirichlet parameter $\boldsymbol{\alpha}$ is set to $(0.1, 0.1)^\top$. Finally, $n = \{50, 100, 200, 500, 1,000\}$ samples are drawn from the generative model in (3.2). For each value of $n$, ten data sets are simulated. Applying MELD using $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation converges in fewer than 40 iterations with accurate estimates of the mean parameters for the categorical, Gaussian, and Poisson variables (Figures 3.4A, 3.4B; Table B.8). In this simulation, FI again is consistently maximized at the

correct value of the number of latent components (Table 3.6).

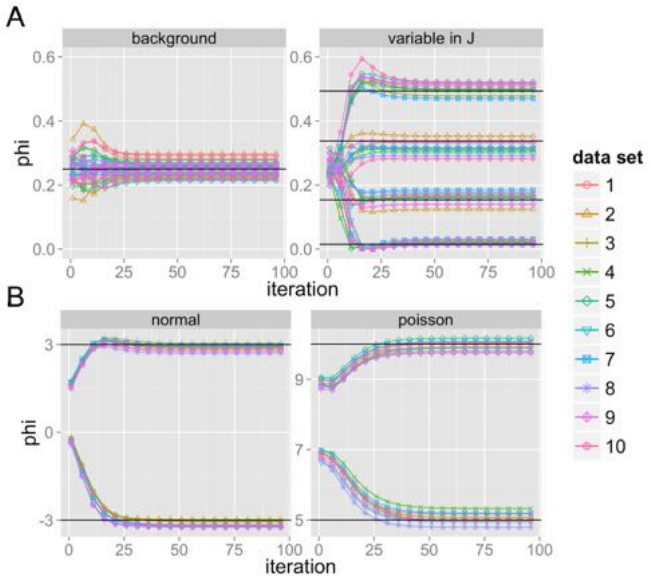


FIGURE 3.4: MELD applied to simulated categorical, Gaussian, and Poisson mixed data types. Parameters are estimated using MELD $Q^{(2)}(\mathbf{\Phi})$ on ten simulated data sets with $n = 1,000$ and $k = 2$. Results shown are for the first stage estimation. Panel A: Convergence of parameter estimates for categorical variables with true parameters drawn as dark lines. Panel B: Convergence of parameter estimates for Gaussian and Poisson variables with true parameters drawn as dark lines.

Table 3.6: Goodness of fit test using fitness index (FI) for categorical, Gaussian, and Poisson mixed data simulation. Values of FI closer to one indicate a better fit. Values shown are the results of applying MELD $Q^{(2)}(\mathbf{\Phi})$ on ten simulated data sets. Standard deviations of FI across the ten simulations are in parentheses.

| | $k$ | 1 | **2** | 3 | 4 |
|---|---|---|---|---|---|
| $n = 50$ | 1st stage | 0.985(0.005) | **0.993(0.002)** | 0.993(0.002) | 0.993(0.002) |
| | 2nd stage | 0.059(1.988) | **0.750(0.006)** | 0.732(0.006) | 0.711(0.007) |
| $n = 100$ | 1st stage | 0.989(0.003) | **0.997(0.001)** | 0.997(0.001) | 0.997(<0.001) |
| | 2nd stage | 0.854(0.047) | **0.872(0.002)** | 0.855(0.002) | 0.839(<0.002) |
| $n = 200$ | 1st stage | 0.991(0.001) | **0.998(<0.001)** | 0.998(<0.001) | 0.998(<0.001) |
| | 2nd stage | 0.920(0.018) | **0.935(<0.001)** | 0.925(0.001) | 0.916(0.001) |
| $n = 500$ | 1st stage | 0.992(0.001) | **0.999(<0.001)** | 0.999(<0.001) | 0.999(<0.001) |
| | 2nd stage | 0.964(0.004) | **0.974(<0.001)** | 0.970(<0.001) | 0.965(<0.001) |
| $n = 1,000$ | 1st stage | 0.993(<0.001) | **1.000(<0.001)** | 0.999(<0.001) | 0.999(<0.001) |
| | 2nd stage | 0.978(0.002) | **0.987(<0.001)** | 0.985(<0.001) | 0.982(<0.001) |

## 3.4 Applications

In this section we apply MELD to three public available data sets.

### *3.4.1 Promoter sequence analysis*

We apply MELD to promoter data available in the UCI machine learning repository (Lichman, 2013). The data include $n = 106$ nucleotide sequences $\{A, C, G, T\}$ of length 57. The first 53 sequences are located in promoter regions of the genome, and the last 53 sequences are located in non-promoter regions. The goal for these data is binary classification: Using nucleotide sequence we would like to predict whether or not the sequence is in a promoter or a non-promoter region. Here, we include the promoter or non-promoter status of the sequences as an additional binary variable, giving us $p = 57 + 1$ categorical variables. We apply MELD using $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation on the full data and also on the subset of the sequences in the promoter region and the subset of sequences in non-promoter regions separately (removing the promoter status variable). We set $k = \{1, \ldots, 8\}$. For $k = 2$, MELD converges in 2.13 seconds, compared with SFA, which takes 41.6 seconds to perform $10,000$ MCMC iterations for the same value of $k$. We evaluate different values of $k$ using the goodness of fit test. FI selects two components for the full data, two components for the promoter data, and one component for the non-promoter data (Table 3.7).

Table 3.7: Goodness of fit testing using the fitness index (FI) on the promoter data. Values shown are the result of applying MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation to the promoter data set.

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| full | 0.913 | **0.915** | 0.911 | 0.904 | 0.896 | 0.890 | 0.881 | 0.871 |
| promoter | 0.890 | **0.896** | 0.888 | 0.862 | 0.833 | 0.811 | 0.769 | -4.292 |
| non-promoter | **0.842** | 0.835 | 0.826 | 0.819 | 0.807 | 0.795 | 0.780 | 0.762 |

We choose $k = 2$ in following analysis. For each nucleotide position, we calculate the averaged KL distance between the estimated component distributions and its marginal distribution using equation (3.27). A biological interpretation of this metric is that it quantifies the stratification of each nucleotide distribution across components: A larger value of the averaged KL distance indicates greater stratification across components, which suggests that the nucleotide is important in differentiating the components. For the full data set, we observe approximately two peaks of the averaged KL distance, one around the 15th nucleotide and one around the 42nd nucleotide (Figure 3.5). The first peak corresponds to the start of the biologically conserved region for promoter sequences (Harley and Reynolds, 1987). For MELD applied only to promoter sequences, this peak is reduced, suggesting that, at approximately the 15th nucleotide, the components all include similarly well conserved distributions of this nucleotide. However, this peak is found in non-promoter sequences, meaning they have diverged component distributions. Together with the peak in full data set, we could reason that the peak in the full data set is caused by the stratification of promoter and non-promoter sequences; nucleotides around this peak have conserved distributions (important) for promoter sequences well they are relatively diverged in non-promoter regions (less important). The estimated component membership variables also show the importance of nucleotides around those nucleotides (Figure 3.6 and 3.7). For the peak around the 42nd nucleotide, this phenomenon is reversed. The increased averaged KL distance remains in promoter sequences but diminishes in non-promoter sequences. One possible explanation is that this region is important to non-promoter sequences well less important for promoter sequences.

We next use normalized mutual information to describe the dependence between nucleotide pairs (Bhattacharya and Dunson, 2012). Mutual information (MI) is a symmetric metric used to quantify the statistical information shared between two distributions (Cover and Thomas, 2006). Here, we use the normalized MI (nMI) by

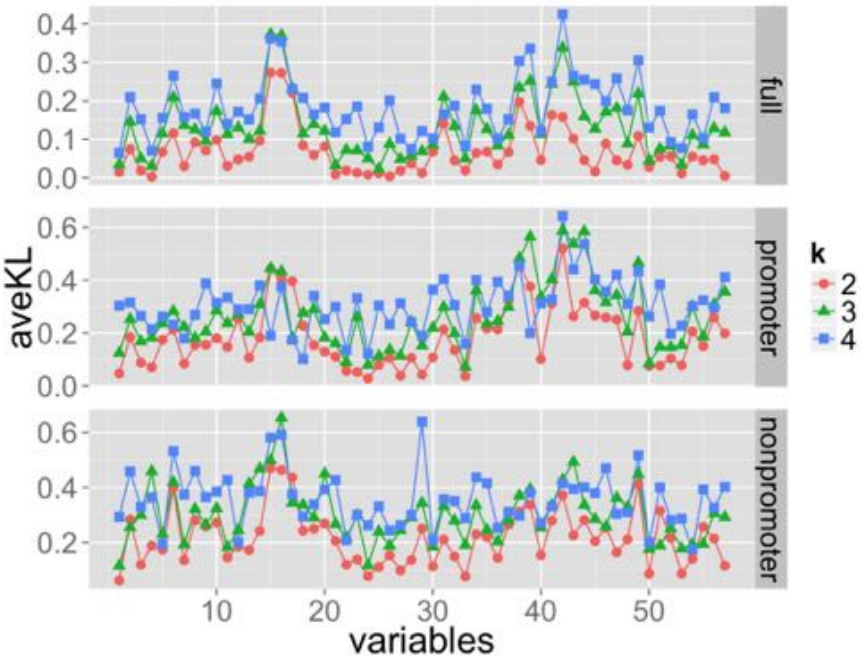


FIGURE 3.5: Averaged Kullback-Leibler distance of MELD applied to the promoter data. The x-axis is the nucleotide position. The y-axis is the averaged Kullback-Leibler (KL) distance between the estimated component distributions and the marginal frequency of each nucleotide. The three rows include the averaged KL distance across the full set of sequences (plus the binary classification vector, not shown; top), across the promoter sequences (middle), and across the non-promoter sequences (bottom).

dividing the MI by the geometric mean of entropies of the two distributions (Strehl and Ghosh, 2003). The nMI between the $j$th and $t$th categorical variable in MELD is calculated using

$$\text{nMI}(y_{ij}, y_{it}) = \frac{\text{MI}(y_{ij}, y_{it})}{\sqrt{\text{H}(y_{ij})\text{H}(y_{it})}},$$

where $\text{MI}(y_{ij}, y_{it})$ is the mutual information of $y_{ij}$ and $y_{it}$, and $\text{H}(y_{ij})$ is the entropy of $y_{ij}$

$$\text{MI}(y_{ij}, y_{it}) = \sum_{c_j} \sum_{c_t} \Pr(y_{ij} = c_j, y_{it} = c_t) \log \left( \frac{\Pr(y_{ij} = c_j, y_{it} = c_t)}{\Pr(y_{ij} = c_j)\Pr(y_{it} = c_t)} \right),$$

$$\text{H}(y_{ij}) = -\sum_{c_j} \Pr(y_{ij} = c_j) \log[\Pr(y_{ij} = c_j)].$$

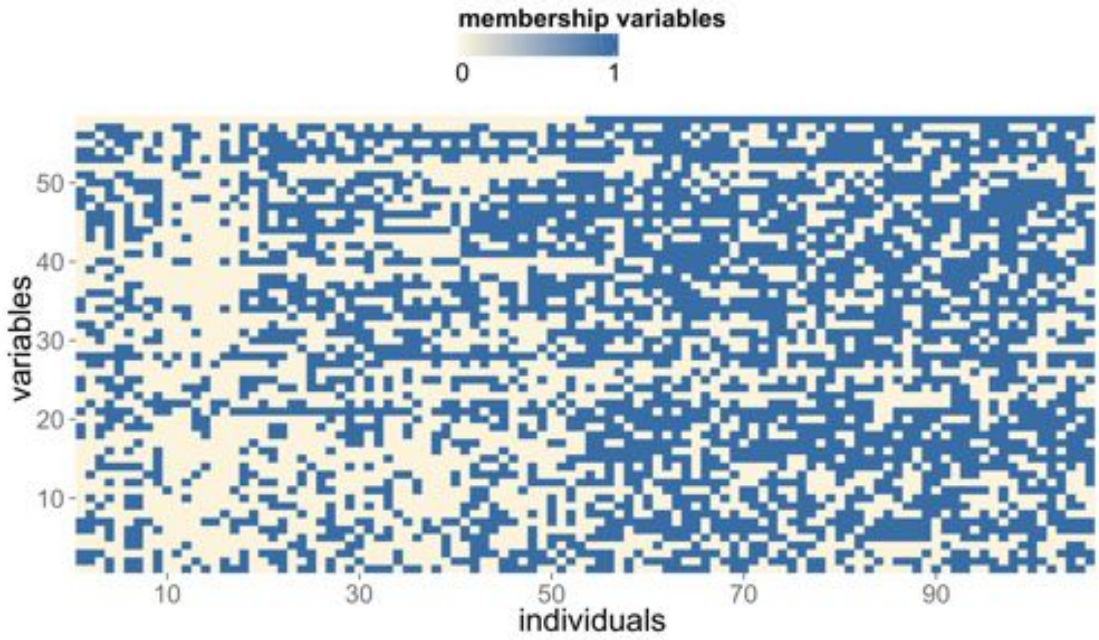


FIGURE 3.6: Recovered membership variables in application of promoter sequence analysis. The results shown are the membership variables for full sequence data with $k = 2$. Promoter and non-promoter sequences are correctly classified (top row).

The terms $\Pr(y_{ij} = c_j)$ and $\Pr(y_{ij} = c_j, y_{it} = c_t)$ are estimated using the first and second moment equations. When $y_{ij} \equiv y_{it}$, then $\text{nMI}(y_{ij}, y_{it}) = 1$.

We analyze the nMI of the results from applying MELD to the promoter data with $k = 2$. The regions around the 15th nucleotide and the 42nd nucleotide show relatively high nMI in the full sequence results (Figure 3.8A). However the dependence around the 15th nucleotide position is not observed in promoter sequences (Figure 3.8B). Neither of the two nMI peaks are observed in the results from the non-promoter sequences only (Figure 3.8C).

### *3.4.2 Political-economic risk data*

In a second application, we apply MELD to political-economic risk data (Quinn, 2004), which include five proxy variables of mixed types measured for 62 countries. The data set has been collected and analyzed to quantify a sense of political-economic

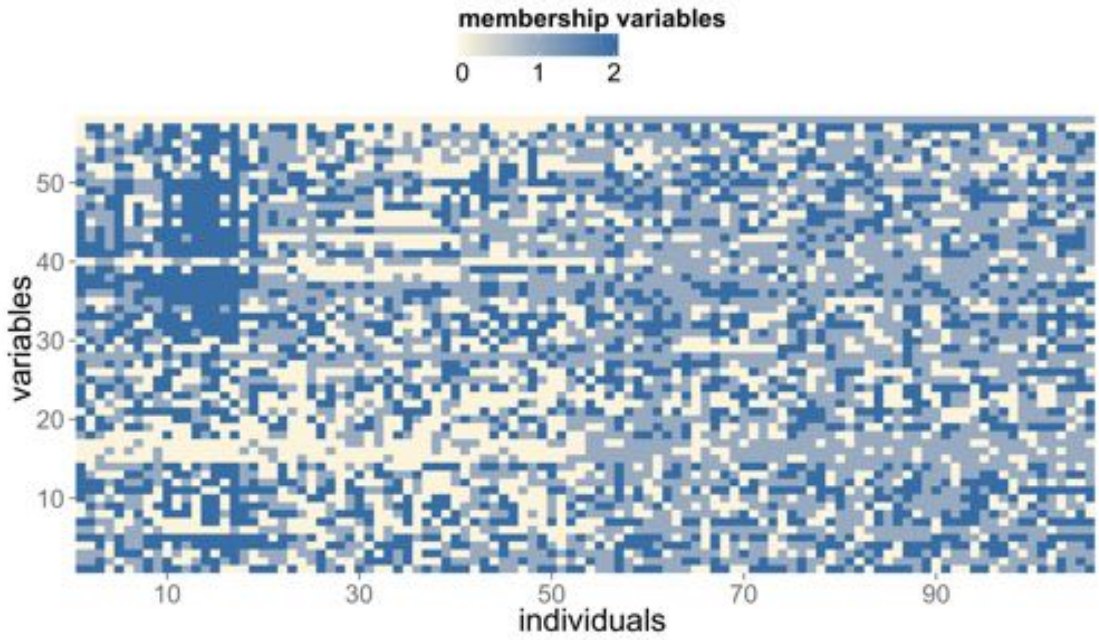


FIGURE 3.7: Recovered membership variables in application of promoter sequence analysis. The results shown are the membership variables for full sequence data with $k = 3$. Promoter and non-promoter sequences are correctly classified (top row).

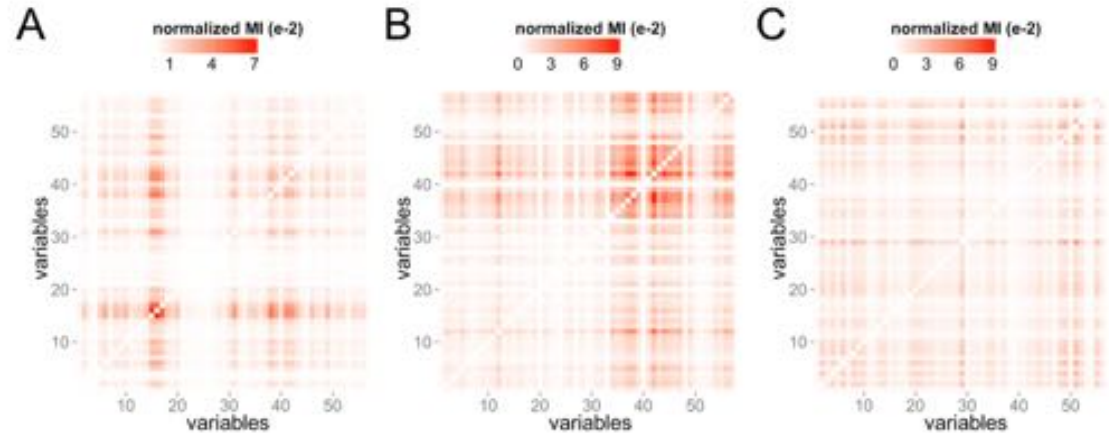


FIGURE 3.8: Normalized mutual information in the promoter data. The normalized mutual information (nMI) between every nucleotide pair is calculated using parameters estimated by MELD with $k = 2$. Panel A: Results for the full data. Panel B: Results for promoter sequences only. Panel C: Results for non-promoter sequences only.

risk, a latent quantity associated with each of the 62 countries, using a probit factor model (Quinn, 2004) and a Bayesian copula factor model (Murray et al., 2013). The data are available in the `MCMCpack` package. Treating the ordinal variables as categorical, there are three categorical variables and two real valued variables. The details of the five variables are shown in Table 3.8.

Table 3.8: Variables in the political-economic risk data

| Variable | Type | Explanation |
|---|---|---|
| `ind.jud` | binary | An indicator variable that measures the independence of the national judiciary. This variable is equal to one if the judiciary is judged to be independent and equal to zero otherwise. |
| `blk.mkt` | real | Black-market premium measurement. Original values are measured as the black-market exchange rate (local currency per dollar) divided by the official exchange rate minus one. Quinn (2004) transformed the original data to log scale. |
| `lack.exp.risk` | ordinal | Lack of appropriation risk measurement. Six levels with coding $0 < 1 < 2 < 3 < 4 < 5$. |
| `lack.corrup` | ordinal | Lack of corruption measurement. Six levels with coding $0 < 1 < 2 < 3 < 4 < 5$. |
| `gdp.worker` | real | Real gross domestic product (GDP) per worker in 1985 international prices. Recorded data are log transformed. |

We apply MELD with $k = \{1, \ldots, 5\}$ using both $Q^{(2)}(\mathbf{\Phi})$ and $Q^{(3)}(\mathbf{\Phi})$ with first stage estimation to the data set. For $Q^{(2)}(\mathbf{\Phi})$ with $k = 3$ MELD converged in 0.10s, and for $Q^{(3)}(\mathbf{\Phi})$ with $k = 3$ MELD converged in 0.45s. The Bayesian copula factor model takes 0.91s to complete $10,000$ MCMC iterations. The FI criterion for $Q^{(2)}(\mathbf{\Phi})$ selects $k = 4$ and, for $Q^{(3)}(\mathbf{\Phi})$, selects $k = 3$ (Table 3.9). We choose results from $Q^{(3)}(\mathbf{\Phi})$ with $k = 3$ for further analysis.

The estimated component parameters for the five variables show distinct interpretations of the three components (Figure 3.9). We might interpret the three components as low-risk, intermediate-risk, and high-risk political-economic status respectively. The first component has a high probability of independence of the national judiciary (`ind.jud` being one) and a low measurement of black-market pre-

Table 3.9: Goodness of fit test using fitness index (FI) in political-economic risk data. Values shown are the results of application of MELD $Q^{(2)}(\mathbf{\Phi})$ and $Q^{(3)}(\mathbf{\Phi})$ with first stage estimation.

| $k$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $Q^{(2)}(\mathbf{\Phi})$ | 0.9974 | 0.9996 | 0.9996 | **0.9998** | 0.9927 |
| $Q^{(3)}(\mathbf{\Phi})$ | 0.9181 | 0.9791 | **0.9885** | 0.9861 | 0.9844 |

mium. The first component also has a high probability of observing high levels in `lack.exp.risk` (4, 5) and in `lack.corrup` (3, 4, and 5). The mean of the GDP per worker is highest among the three components. The second component, on the other hand, has a relatively high probability of being zero in `ind.jud` and a large mean value of `blk.mkt`. Both of lack of appropriation risk measurement and lack of corruption measurement put higher weights on lower levels (0, 1 and 2), indicating more risk and higher levels of corruption. The GDP per worker is still high. We might interpret this component as a society being relatively unstable while still having a good economic forecast, meaning that GDP per worker is high, possibly through the black market. The last component has the least judicial independence as quantified by the probability of `ind.jud` being zero. The black-market premium is also low, as is the lack of risk level and lack of corruption level. The GDP per worker is by far the lowest among the three components. We might interpret this component as society being the most unstable with the greatest economic risk. We find although the three components have distinct stratification, each country is a mixture of the three components reflected by the recovered memebership variables shown in Figure 3.10. Assigning a country to a mixture of components allows our method to find clear stratified components from the mixed data type observations.

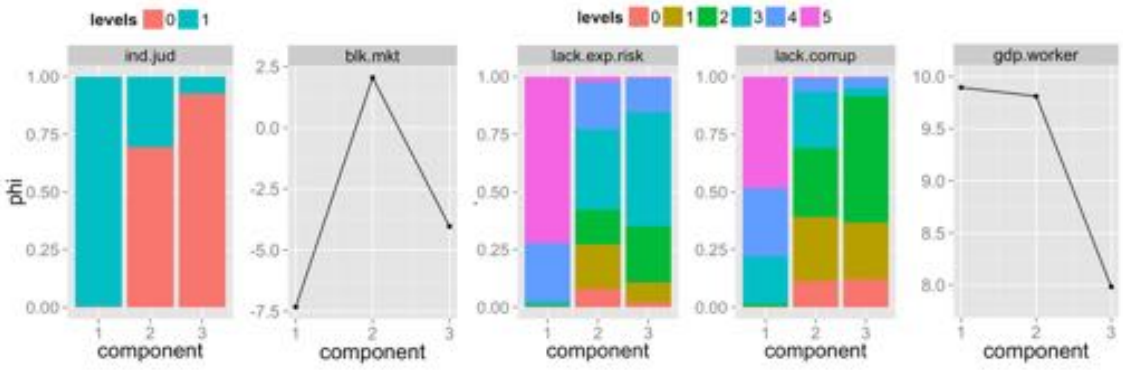


FIGURE 3.9: Estimated component parameters for the political-economic risk data. Results shown from applying MELD to the data set using $Q^{(3)}(\mathbf{\Phi})$ with $k = 3$ components. For the real-valued variables, component mean parameters are plotted. For the categorical variables, component-wise relative proportions are plotted.

### *3.4.3 Gene expression quantitative trait loci mapping*

As a third application we apply MELD to Human HapMap phase 3 (HM3) project genotype and gene expression data (Stranger et al., 2012) to perform a gene expression quantitative trait loci (eQTL) mapping. We hypothize that population structure, which stratifies the distributions of genotype, might act in a similar manner on quantitative traits. Such stratification might be a reason to cause dependence and association between a SNP and a trait. The HM3 data set consists of 608 individuals. We focus on SNP's on chromosome 21 having a minor allele frequency greater than 5%. SNP's are represented by the number of copies of the minor allele at each loci, and we obtain $1,672$ SNP's after selecting every 10th loci to avoid including SNP's in strong linkage. For gene expression data we extract genes on chromosome 21 (237 genes) for analyses.

We employ two analyses. We first apply MELD to SNP data only using $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation. This can be viewed as estimating genotype distributions in different latent sub-populations. Then we apply MELD to both SNP and gene expression data. In this case, we attempt to find how including gene expression traits

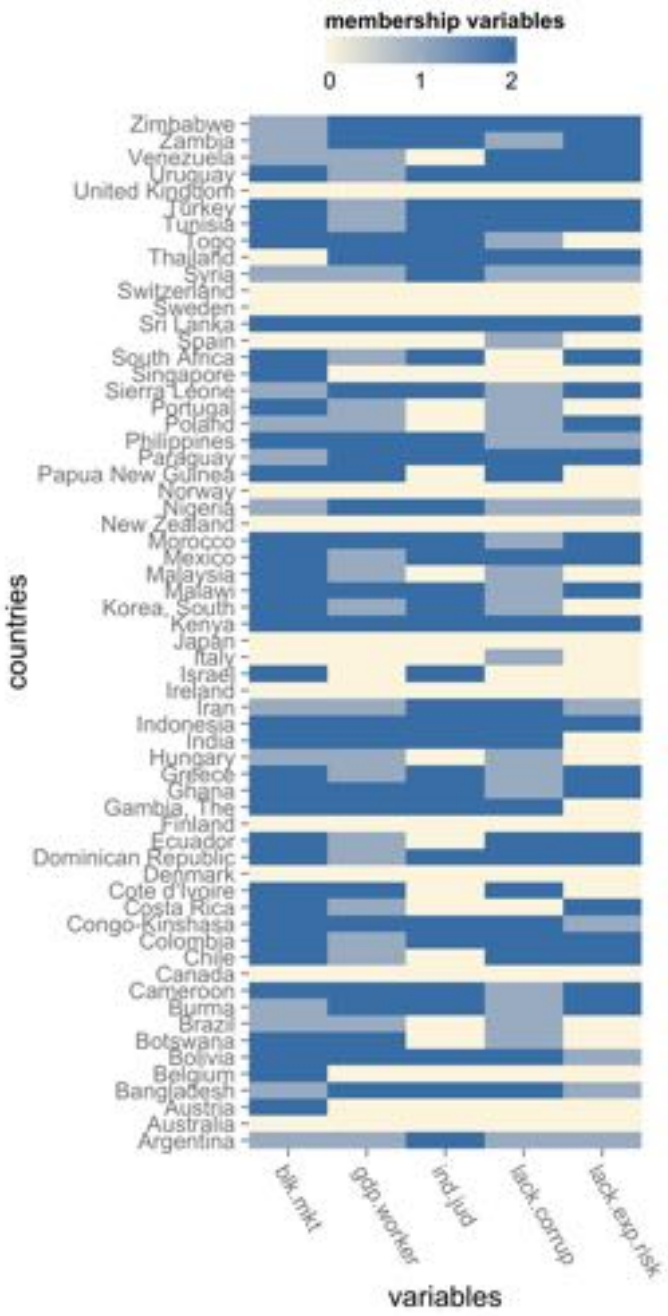


FIGURE 3.10: Recovered membership variables in application of political-economics risk data set. The results shown are the membership variables for political-economics risk data with $k = 3$ using MELD $Q^{(3)}(\mathbf{\Phi})$.

would influence the population stratification and how such stratification is related to eQTL. We consider different number of sub-populations including $k = \{1, 2, 3, 4, 5\}$. FI chooses $k = 2$ sub-populations. Therefore in following analysis we set $k$ to this number.

Table 3.10: Goodness of fit test using fitness index (FI) in HapMap phase 3 data set. Values shown are the results of application of MELD $Q^{(2)}(\mathbf{\Phi})$ with first stage estimation on the selected chromosome 21 data set.

| $k$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| SNP only | 0.987 | **0.995** | 0.973 | 0.867 | 0.762 |
| SNP+expression | 0.993 | **0.997** | 0.988 | 0.941 | 0.896 |

We first analyze the recovered membership variables (Figure 3.11 and 3.12). The results suggest that there is a clear population structure among the 608 individuals in SNP data, reflected by the different bands in the two figures. However we do not observe a clear population structure in gene expression data (Figure 3.12). Then we calculate average KL distance for the $1,672$ SNPs without and with gene expression included (Figure 3.13). Several SNP loci show high values of averaged KL distance, suggesting they have clear differentiated distributions among sub-populations. The averaged KL distances do not change after inclusion of gene expression data. We extract the first five SNP's with highest average KL distance. Their genotypes are shown in Table 3.11. We further examine whether there are significant associations with those SNP's by performing univariate regression tests against the expressions of the 237 genes. Under pvalue cutoff of 0.05 we do not observe any significant associations.

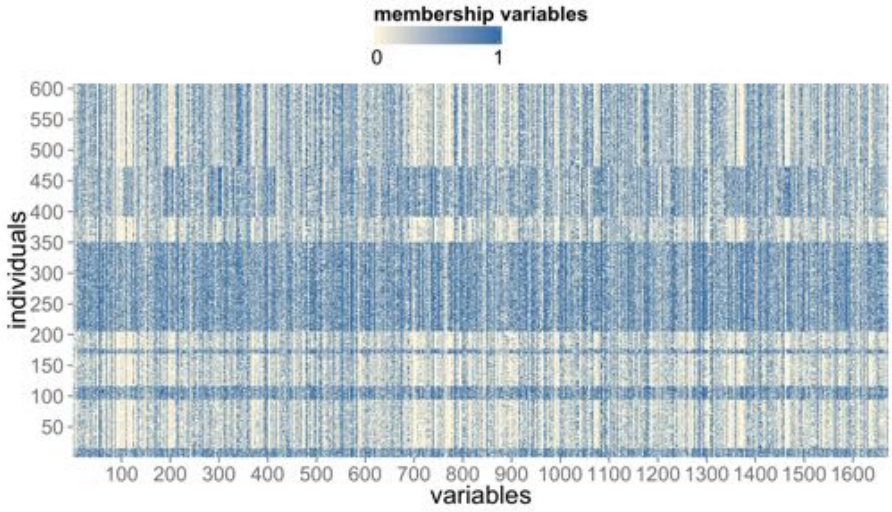


FIGURE 3.11: Recovered membership variables in application of HM3 chromosome 21 data. The results shown are the membership variables for SNP data only with $k = 2$.

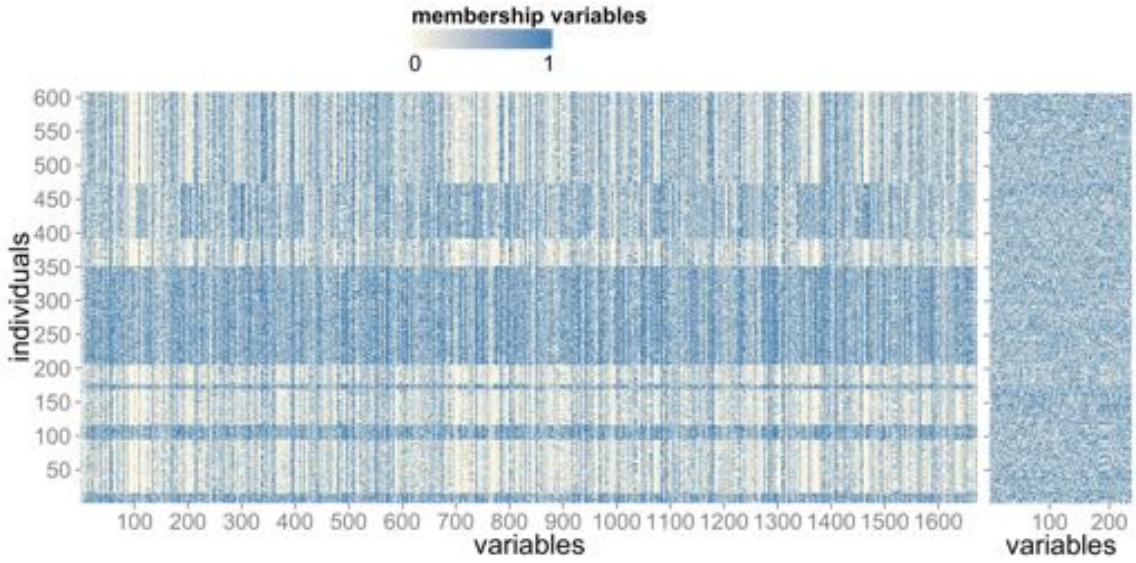


FIGURE 3.12: Recovered membership variables in application of HM3 chromosome 21 data. The results shown are the membership variables of for both SNP and gene expression data with $k = 2$.

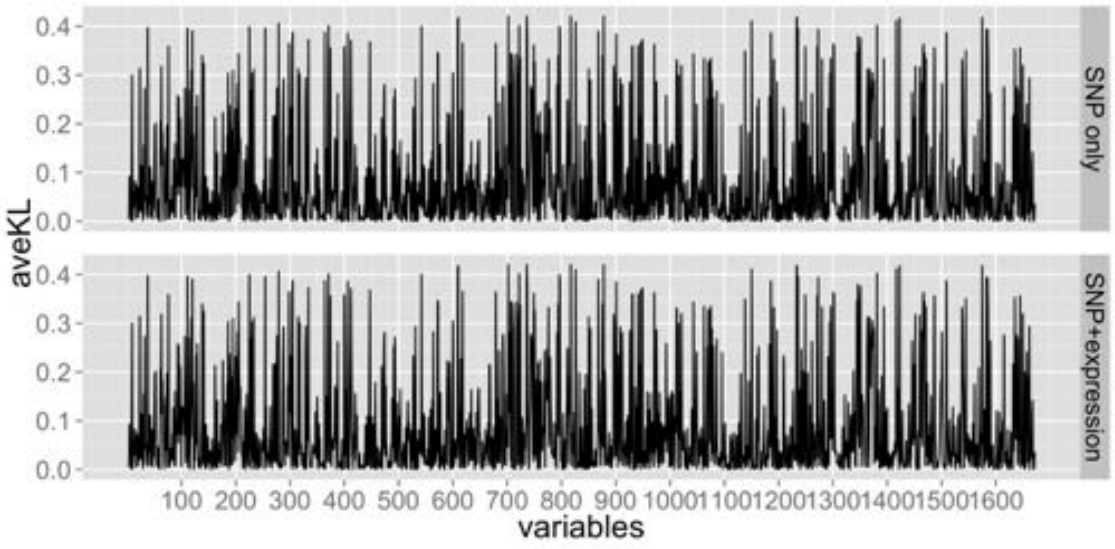


FIGURE 3.13: Averaged Kullback-Leibler distances in HM3 chr21 data. The averaged Kullback-Leibler (KL) distance between estimated component distributions from MELD and marginal frequency for each SNP using equation (3.27) is calculated with $k = 2$.

## 3.5 Discussion and conclusion

In this chapter, we have developed a new class of latent variable models with Dirichlet-distributed latent variables for mixed data types. These generalized Dirichlet latent variable models extend previous mixed membership models such as LDA (Blei et al., 2003) and simplex factor models (Bhattacharya and Dunson, 2012) to allow mixed data types. For this class of models, we develop a fast parameter estimation procedure using generalized methods of moments. Our GMM estimator is consistent, requiring the correct specification of first moment of component distributions. Efficiency can be achieved by deriving an optimal weight matrix.

Our moment functions are similar to the moment tensor approaches developed in recent work (Anandkumar et al., 2014b). The key novelty of our moment functions is they are constructed using heterogeneous low order polynomials instead of homogeneous polynomials. The heterogeneity of the moment functions allows us to

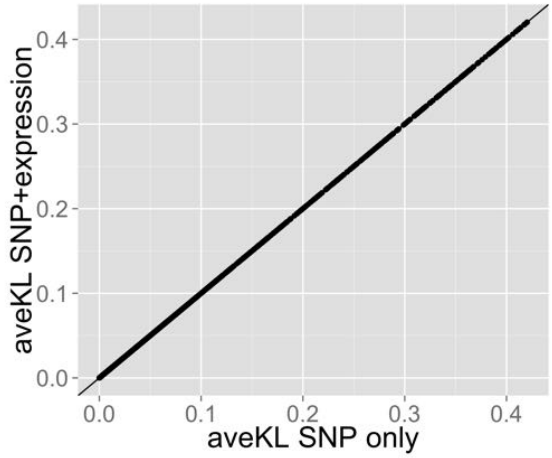


FIGURE 3.14: Averaged KL distance in application of HM3 chromosome 21 data with and without gene expression data included under $k = 2$.

develop a fast Newton-Raphson method for parameter estimation. The computational advantage of MELD over other parameter estimation methods such as EM and MCMC is that parameter estimation does not require the instantiation of the latent variables. We derive population moment conditions after marginalizing out the sample-specific Dirichlet latent variables. Results suggest that the fitness index (FI) (Bentler, 1983) is a reliable metric for selecting the number of components in this framework.

We demonstrate the utility of our approach using simulation studies and three applications. Our results show that MELD is a promising alternative to MCMC or EM methods for parameter estimation, producing fast and robust parameter estimates. Since our method depends only on certain forms of sample moments, parameter estimation does not scale with sample size $n$ after observed data are transformed to the moment statistics. An online method to update moment statistics when new samples arrive would allow re-estimation of the parameters to include new observations. One limitation of our method is that the Newton-Raphson method is of order $O(p^2)$ using second moment functions and order $O(p^3)$ using third moment functions. One

possible approach speed up MELD with large $p$ problems is to use stochastic gradient methods to calculate an approximate gradient in each step.

Table 3.11: Top 5 SNP's with largest averaged KL distances in HM3 chromosome 21 data.

| SNP | Genotype across 608 individuals | aveKL |
|---|---|---|
| SNP1 | 22222222222222200000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000022222222222222222222122<br>00000000000000000000000000000000000000000000000000000000222222<br>22000000000000000000000000000022222222222222222222222222<br>22222222222222222212222222222222212222222222212122212222222<br>22121222221222222222222222222222222222222212222222222222222<br>22000000000000000000000000000000000000000000022222222222222<br>22221221212122222222112221222212222222222222222212222222221<br>22222222100000000000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000000000000000000000000<br>00000000000000000000000000 | 0.4203 |
| SNP2 | 22222211222212200000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000022222222212212222222222<br>00000000000000000000000000000000000000000000000000000000222222<br>12000000000000000000000000000022122222221222222212221222<br>22222222121222222222222122222221221222222221222222222222222<br>22221222222122222222222222222222222222222222222222222222221<br>22000000000000000000000000000000000000000000221222212122222<br>22222222211222211222222212222212222222222122121212222221<br>22222221200000000000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000000000000000000000000<br>00000000000000000000000000 | 0.4202 |
| SNP3 | 22112222222221200000000000000000000000000000000000000000000<br>00000000000000000000000000000000000022122222221222222222222<br>00000000000000000000000000000000000000000000000000000222122<br>22000000000000000000000000000022221212211222222222222222<br>22222222221222222212222222222222222212221222212222222222122<br>21212222212222212222122222221212211222122222222212122222<br>22000000000000000000000000000000000000000000022212222222222<br>22222222222222212222222222122222222222222222222222122221122<br>22222222220000000000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000000000000000000000000<br>00000000000000000000000000 | 0.4202 |
| SNP4 | 22222222222222200000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000022222222222222222222222<br>00000000000000000000000000000000000000000000000000000000122222<br>21000000000000000000000000000022222222222222222222222222<br>22222222222222222222222222212222222222222222222122222222222<br>21222222122222222212222222222222212222222222222222222222222<br>22000000000000000000000000000000000000000000022222222222222<br>22222222212222221222212222222221222222122222222222222122222<br>22222222220000000000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000000000000000000000000<br>00000000000000000000000000 | 0.4201 |
| SNP5 | 22222222222222200000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000022222222222222222222222<br>00000000000000000000000000000000000000000000000000000000222122<br>12000000000000000000000000000022222221222222222222222222<br>22222222122222222222222222222222222222122222222212222222212<br>22212222221222222222122222212221212222212222222212222222222<br>22000000000000000000000000000000000000000000022222212222221 22<br>11121202222222122112211221221221221121211212212211112222<br>22212222200000000000000000000000000000000000000000000000000<br>00000000000000000000000000000000000000000000000000000000000<br>00000000000000000000000000 | 0.4181 |

# 4

# An efficient Monte Carlo method for distributions on manifolds

In Chapter 3 we develop a generalized method of moments (GMM) approach named MELD for the latent Dirichlet variable model with mixed data types. The moment functions specified in MELD are derived from a specific probability model. However the parameter estimation avoids manipulations of the likelihood function. This is distinct from the GMM often used in econometrics literature in which a probability model is avoided. In this chapter, we are going to provide some preliminary investigations of embedding the GMM approach to a likelihood context. Parameter estimations are conducted under Bayesian framework. Using a Bayesian approach for parameter estimation in MELD has at least two advantages. First posterior computations could be performed using efficient MCMC algorithms. Second, the selection of weight matrix $\boldsymbol{A}^{(\cdot)}$ in construction of GMM estimator could be avoided. The weight matrix $\boldsymbol{A}^{(\cdot)}$ in the original GMM is used to penalize moment functions in a way that moment functions with smaller variance or covariance receive larger weights in the objective function, and vice versa. When embedding in Bayesian framework such effects could be achieved by introducing additional penalizing variables. The

posterior distributions of the penalizing variables could be estimated from posterior draws using MCMC algorithms. Moreover, model selections can be evaluated using theories in Bayesian literature.

One major step in the posterior computation is to draw samples from a distribution defined on a probability simplex. Drawing samples from such a density is not trivial. An inefficient sampler could make the posterior computation mix poorly. With this problem in mind, we are motivated to develop an efficient Monte Carlo method that draws samples from distributions defined on Riemannian manifolds. Our method combines Hamiltonian Monte Carlo (HMC) algorithm with a geodesic integrator. The HMC component allows our method to accept bold moves in the parameter space and the geodesic integrator component restricts the moves on a manifold.

The rest of this chapter is arranged as follows. In Section 4.1 we give a brief review of Bayesian generalized method of moments. In Section 4.2 we introduce the pseudo-likelihood functions used for parameter estimation in MELD. The problem of sampling from a distribution defined on the probability simplex appears. In Section 4.3, an efficient Markov chain Monte Carlo method is developed to draw samples from such a distribution. Our method could also be applied to distributions on other types of manifolds with closed form equations of geodesic flow, for example the Stiefel manifold. We perform simulation study to evaluate our method in Section 4.4 and we conclude with a discussion in Section 4.5.

## 4.1 Bayesian generalized method of moments

Both of the method of moments (MM) and generalized method of moments (GMM) introduced in Section 3.2 have been investigated in Bayesian framework by several authors. Zellner (1996) develops a Bayesian method of moments approach for regression problems. In his paper the author proposes first and second order posterior

moment constraints and shows that the relations between the moment estimators and those obtained under a likelihood model with diffuse priors. Posterior densities are derived using maximum entropy criteria. Yin (2009) studies generalized linear models with correlated observations. Such data are often observed in longitudinal studies. The author builds on the generalized estimating equations (GEE) developed by Zeger and Liang (1986) and Liang and Zeger (1986) in longitudinal data analysis and develops a Bayesian GMM by constructing a pseudo-likelihood using GEE. Posteriors of regression coefficients and unknown correlation matrix are estimated using MCMC algorithms with Gibbs sampling. Although the GEE resemble score equations in a likelihood based analysis, the studies mentioned above are different from our Bayesian approach in the sense that those methods completely avoid specification of a likelihood function. In contrast, the Bayesian GMM strategy studied in this chapter uses moment functions derived from a likelihood model.

## 4.2 Pseudo-likelihood and posterior

In this section we are going to introduce the pseudo-likelihood functions developed for MELD. Then we assign prior distributions on parameters and derive their conditional posterior distributions. We first use the second order moment matrices defined in (3.10) to construct a pseudo-likelihood function. We define

$$L^{(2)}(\mathbf{\Phi}) \propto \prod_{j,t,t>j} (\tau_{jt}^{(2)})^{-d_j d_t/2} \exp\left( -\frac{1}{2\tau_{jt}^{(2)}} ||\boldsymbol{E}_{n,jt}^{(2)} - \mathbf{\Phi}_j \mathbf{\Lambda}^{(2)} \mathbf{\Phi}_t^\top||_F^2 \right). \tag{4.1}$$

Here $\boldsymbol{E}_{n,jt}^{(2)}$ is defined in (3.16). The idea behind the likelihood function is that we multiply $||\boldsymbol{F}_{n,jt}^{(2)}(\mathbf{\Phi})||_F^2 = ||\boldsymbol{E}_{n,jt}^{(2)} - \mathbf{\Phi}_j \mathbf{\Lambda}^{(2)} \mathbf{\Phi}_t^\top||_F^2$ by $-1/(2\tau_{jt}^{(2)})$ and exponentiate the result to get a Gaussian kernel for every $j, t$ with $j < t$. The final likelihood function is the product of the $p(p-1)/2$ kernels. The additional parameters $\{\tau_{jt}^{(2)}\}$ give weights to the second order moment matrices and the weights for the functions in

$\text{vec}[\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})]$ are assumed to be equal. Those parameters play a similar role as the weight matrix $\boldsymbol{A}^{(2)}$ in GMM estimation. Following the similar idea, we can define a pseudo-likelihood function using both second moment matrices and third moment tensors in (3.10) and (3.11)

$$L^{(3)}(\boldsymbol{\Phi}) \propto \prod_{j,t,t>j} (\tau^{(2)}_{jt})^{-d_j d_t/2} \exp\left( -\frac{1}{2\tau^{(2)}_{jt}} ||\boldsymbol{E}^{(2)}_{n,jt} - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^\top||^2_F \right)$$

$$\times \prod_{j,s,t,t>s>j} (\tau^{(3)}_{jst})^{-d_j d_s d_t/2} \exp\left( -\frac{1}{2\tau^{(3)}_{jst}} ||\boldsymbol{E}^{(3)}_{n,jst} - \boldsymbol{\Lambda}^{(3)} \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t||^2_F \right). \quad (4.2)$$

Here $\boldsymbol{E}^{(3)}_{n,jst}$ is defined in (3.17). The parameters $\{\tau^{(3)}_{jst}\}$ give weights to the third order moment tensors in a similar manner as $\{\tau^{(2)}_{jt}\}$ to second moment matrices. We let $\boldsymbol{T}^{(2)} = \{\tau^{(2)}_{jt}\}$ and $\boldsymbol{T}^{(3)} = \{\tau^{(3)}_{jst}\}$ and treat them as unknown variables.

To complete the specification, we assign priors on $\boldsymbol{\phi}_{jh}$, $\tau^{(2)}_{jt}$ and $\tau^{(3)}_{jst}$. For $j$th variable being categorical, we give $\boldsymbol{\phi}_{jh}$ a Dirichlet prior $\text{Dir}(\boldsymbol{\beta}_j)$ with $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jd_j})^\top$. The posterior of $\boldsymbol{\phi}_{jh}$ with likelihood function (4.1) can be written as

$$p(\boldsymbol{\phi}_{jh}|-) \propto \exp\left( (\boldsymbol{\xi}^{(2)})^\top \boldsymbol{\phi}_{jh} + \gamma^{(2)} \boldsymbol{\phi}_{jh}^\top \boldsymbol{\phi}_{jh} + \sum_{c=1}^{d_j} (\beta_{jc} - 1) \log(\phi_{jhc}) \right), \quad (4.3)$$

where

$$\boldsymbol{\xi}^{(2)} = \lambda^{(2)}_h \sum_{t=1,t\neq j}^{p} \frac{1}{\tau^{(2)}_{jt}} (\overline{\boldsymbol{E}}^{(2)}_{n,jt} \boldsymbol{\phi}_{th}),$$

$$\gamma^{(2)} = -(\lambda^{(2)}_h)^2 \sum_{t=1,t\neq j}^{p} \frac{1}{2\tau^{(2)}_{jt}} \boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th},$$

and $\overline{\boldsymbol{E}}^{(2)}_{n,jt} = \boldsymbol{E}^{(2)}_{n,jt} - \sum_{h'\neq h} \lambda^{(2)}_{h'} \boldsymbol{\phi}_{jh'} \circ \boldsymbol{\phi}_{th'}$.

The posterior of $\boldsymbol{\phi}_{jh}$ with likelihood (4.2) can be written as

$$p(\boldsymbol{\phi}_{jh}|-) \propto \exp\left( (\boldsymbol{\xi}^{(3)})^\top \boldsymbol{\phi}_{jh} + \gamma^{(3)} \boldsymbol{\phi}_{jh}^\top \boldsymbol{\phi}_{jh} + \sum_{c=1}^{d_j} (\beta_c - 1) \log(\phi_{jhc}) \right), \quad (4.4)$$

where

$$\boldsymbol{\xi}^{(3)} = \lambda_h^{(2)} \sum_{t=1,t\neq j}^{p} \frac{1}{\tau_{jt}^{(2)}} (\overline{\boldsymbol{E}}_{n,jt}^{(2)} \boldsymbol{\phi}_{th}) + \lambda_h^{(3)} \sum_{s=1,s\neq j}^{p} \left[ \sum_{t=1,t\neq s,t\neq j}^{p} \frac{1}{\tau_{jst}^{(3)}} (\overline{\boldsymbol{E}}_{n,jst}^{(3)} \times_2 \boldsymbol{\phi}_{sh} \times_3 \boldsymbol{\phi}_{th}) \right],$$

$$\gamma^{(3)} = -(\lambda_h^{(2)})^2 \sum_{t=1,t\neq j}^{p} \frac{1}{2\tau_{jt}^{(2)}} \boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th} - (\lambda_h^{(3)})^2 \sum_{s=1,s\neq j}^{p} \left[ \sum_{t=1,t\neq s,t\neq j}^{p} \frac{1}{2\tau_{jst}^{(3)}} (\boldsymbol{\phi}_{sh}^\top \boldsymbol{\phi}_{sh})(\boldsymbol{\phi}_{th}^\top \boldsymbol{\phi}_{th}) \right].$$

$\overline{\boldsymbol{E}}_{n,jst}^{(3)}$ is defined as $\overline{\boldsymbol{E}}_{n,jst}^{(3)} = \boldsymbol{E}_{n,jst}^{(3)} - \sum_{h'\neq h} \lambda_{h'}^{(3)} \boldsymbol{\phi}_{jh'} \circ \boldsymbol{\phi}_{sh'} \circ \boldsymbol{\phi}_{th'}$.

The posteriors of $\boldsymbol{\phi}_{jh}$ for categorical variable in (4.3) and (4.4) share a similar following form

$$\exp\left( \boldsymbol{\xi}^\top \boldsymbol{\phi}_{jh} + \gamma \boldsymbol{\phi}_{jh}^\top \boldsymbol{\phi}_{jh} + \sum_{c=1}^{d_j} (\beta_c - 1) \log(\phi_{jhc}) \right). \tag{4.5}$$

They both define a probability distribution on $d_j - 1$ simplex. We are going to introduce a Monte Carlo method that targets the posterior distribution in next section.

When $j$th variable is non-categorical with a support of $\phi_{jh}$ across the real line, we assign a normal prior $\mathrm{N}(\mu_0, \sigma_0^2)$ for $\phi_{jh}$. This prior generates following posterior

$$p(\phi_{jh}|-) \overset{d}{=} \mathrm{N}\left( \frac{\mu_0 + \xi^{(\cdot)}\sigma_0^2}{1 - 2\gamma^{(\cdot)}\sigma_0^2}, \frac{\sigma_0^2}{1 - 2\gamma^{(\cdot)}\sigma_0^2} \right). \tag{4.6}$$

For non-categorical variable with positive value of $\phi_{jh}$, such as Poisson or exponential variable, we assign the prior the same distribution truncated to $\mathbb{R}^+$. The posterior is

$$p(\phi_{jh}|-) \propto \mathrm{N}\left( \frac{\mu_0 + \xi^{(\cdot)}\sigma_0^2}{1 - 2\gamma^{(\cdot)}\sigma_0^2}, \frac{\sigma_0^2}{1 - 2\gamma^{(\cdot)}\sigma_0^2} \right) \mathbf{1}(\phi_{jh} \geqslant 0). \tag{4.7}$$

For $\{\tau_{jt}^{(2)}\}$ and $\{\tau_{jst}^{(3)}\}$ we assign a conjugate prior $\mathrm{Ga}(a_\tau, b_\tau)$ on $(\tau^{(\cdot)})^{-1}$. The resulting posterior is

$$\begin{aligned}
(\tau_{jt}^{(2)})^{-1}|- &\sim \mathrm{Ga}\left( a_\tau + \frac{1}{2} d_j d_t, b_\tau + \frac{1}{2} ||\boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi})||_F^2 \right), \\
(\tau_{jst}^{(3)})^{-1}|- &\sim \mathrm{Ga}\left( a_\tau + \frac{1}{2} d_j d_s d_t, b_\tau + \frac{1}{2} ||\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})||_F^2 \right).
\end{aligned} \tag{4.8}$$

Equations (4.3), (4.4), (4.6), (4.7) and (4.8) complete a Gibbs sampler. The major difficulty comes from drawing samples from (4.3) and (4.4) with the support on $d_j - 1$ simplex. The two equations can be generalized as drawing from a distribution with density proportional to (4.5). We are going to introduce a Monte Carlo method to draw samples from such a distribution.

## 4.3 Drawing from distributions on manifolds

In this section we are going to develop a method to efficiently draw samples from (4.5). Drawing samples from such a density with the support on $d_j - 1$ simplex is not trivial. An inefficient sampler could make the posterior computation mix poorly. We view this problem as drawing samples from a distribution defined on a manifold, which in our case is the $d_j - 1$ simplex. We develop a geodesic Riemannian manifold Hamiltonian Monte Carlo (HMC) algorithm on the parameter manifold (Byrne and Girolami, 2013). To facilitate the algorithm, we first re-parameterize the parameter $\boldsymbol{\phi}_{jh} = (\phi_{jh1}, \ldots, \phi_{jhd_j})^\top$ by letting $\phi_{jhc} = x_{jhc}^2$ for $c = 1, \ldots, d_j$. Then $\boldsymbol{x}_{jh} = (x_{jh1}, \ldots, x_{jhd_j})^\top$ is a point on $d_j - 1$ dimensional sphere $\mathbb{S}^{d_j - 1}$. According to change of variables, the distribution of $\boldsymbol{x}_{jh}$ follows as

$$p(\boldsymbol{x}_{jh}|-) \propto \exp\left( \sum_{c=1}^{d_j} \xi_c x_{jhc}^2 + \gamma \sum_{c=1}^{d_j} x_{jhc}^4 + \sum_{c=1}^{d_j} (2\beta_c - 1) \log(x_{jhc}) \right). \tag{4.9}$$

We are going to develop a HMC algorithm which induces a random walk on $\mathbb{S}^{d_j - 1}$ that targets (4.9) as equilibrium distribution. Transforming back to $\boldsymbol{\phi}_{jh}$ we get posterior draws from (4.5). Before introducing the geodesic Riemannian manifold HMC algorithm, we first give a very brief review about manifold and coordinate embedding.

### *4.3.1 Manifold and embedding*

In this subsection give a very brief summary about manifold and coordinate embedding. More details can be found in the book written by Absil et al. (2009).

*Chart, atlas and coordinate*

Roughly speaking a manifold is a topological space that locally acts like Euclidean space. For any point $\omega \in \mathcal{M}$, its coordinates is defined by a bijective mapping $\phi(\cdot) : \mathcal{M} \to \mathbb{R}^d$ from an open set around $\omega$, denoted by $\mathcal{U}$, to an open set in $\mathbb{R}^d$. The open set and the mapping is defined as a chart $(\mathcal{U}, \phi(\cdot))$ and the image of the mapping is called the coordinate of $\omega$. An atlas is the collection of charts $(\mathcal{U}_\alpha, \phi(\cdot)_\alpha)$ with $\cup_\alpha \mathcal{U}_\alpha = \mathcal{M}$ and $\mathcal{U}_\alpha$ overlap smoothly. The dimension of $\mathcal{M}$ is given by the dimension of the image of $\phi(\cdot)_\alpha$.

*Embedding*

Usually $\omega$ is embedded in a higher dimensional embedding space such as a Euclidean space $\mathbb{R}^d$ with $d \geqslant m$, where $m$ is the dimension of the manifold. Such an embedding space is called an ambient space. The general definition of embedding comes with the mapping between two manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ with dimension $m_1$ and $m_2$. The mapping is immersion when $m_1 \leqslant m_2$ and it is submersion when $m_1 \geqslant m_2$. When the mapping is immersion with $\mathcal{M}_1 \subset \mathcal{M}_2$ and the manifold topology of $\mathcal{M}_1$ coincides with the subspace topology of $\mathcal{M}_2$, then $\mathcal{M}_1$ is called embedded in $\mathcal{M}_2$. In most of examples we face, the embedding space $\mathcal{M}_2$ is a vector space. For example, an element in the Stiefel manifold $\omega \in \mathcal{V}(p, k)$ are embedded in $\mathbb{R}^{p \times k}$ and dimension of $\mathcal{V}(p, k)$ is $pk - p(p+1)/2$. We use $\boldsymbol{\theta}$ to denote the parameters that are represented by embedded coordinates (extrinsic coordinates) in the embedding vector space $\mathbb{R}^d$ of a underlying manifold. For example in our case $\boldsymbol{\phi}_{jh} \in \Delta^{d_j - 1}$ and $\boldsymbol{x}_{jh} \in \mathbb{S}^{d_j - 1}$ are represented by a $d_j$ dimensional vector.

Given an embedded manifold, the tangent can be represented by a vector consisting of coordinate-wise time derivative with respect to any smooth motions defined on the manifold. The tangent space is the set of such vectors and forms a subspace of ambient space of the manifold. For example, the tangent space of a sphere $\mathbb{S}^{d-1}$ with coordinate $\boldsymbol{\theta} \in \mathbb{R}^d$ is formed by $T_{\boldsymbol{\theta}} = \{\boldsymbol{v} \in \mathbb{R}^d \; s.t. \; \boldsymbol{\theta}^\top \boldsymbol{v} = 0\}$.

The Riemannian manifold is a smooth manifold $\mathcal{M}$ equipped with a inner product $g$ defined at every $\boldsymbol{\theta}$ on the manifold as

$$g(\boldsymbol{s}, \boldsymbol{t}) = \boldsymbol{s}^\top \boldsymbol{G}(\boldsymbol{\theta}) \boldsymbol{t}.$$

$\boldsymbol{s}, \boldsymbol{t} \in T_{\boldsymbol{\theta}}$ are two tangent vectors and $\boldsymbol{G}(\boldsymbol{\theta})$ is called metric tensor. We denote a Riemannian manifold and its inner product as $(\mathcal{M}, g)$. The Euclidean space $\mathbb{R}^d$ is also a Riemannian manifold with the inner product as the dot product of vectors.

Let $(\mathcal{M}, g)$ and $(\mathcal{N}, h)$ be two Riemannian manifolds. An isometric embedding is a mapping $\mathcal{M} \to \mathcal{N}$ that preserve the inner product. In particular we are interested in the case where $\mathcal{N}$ is an Euclidean space $\mathbb{R}^d$. In this case, we have

$$\boldsymbol{s}^\top \boldsymbol{G}(\boldsymbol{\theta}) \boldsymbol{t} = \boldsymbol{u}^\top \boldsymbol{v},$$

where $u_i = \sum_j \partial x_i / \partial \theta_j \cdot s_j$ and $v_i = \sum_j \partial x_i / \partial \theta_j \cdot t_j$. Here $\boldsymbol{x}$ is the new coordinate in $\mathbb{R}^d$ of the manifold. This is equivalent to $g_{ij} = \sum_{t=1}^{p} \partial x_t / \partial \theta_i \cdot \partial x_t / \partial \theta_j$. If we let $\boldsymbol{M}$ with $m_{ij} = \partial x_i / \partial \theta_j$, then

$$\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{M}^\top \boldsymbol{M}. \tag{4.10}$$

*Hausdorff measure and Lebesgue measure*

We consider the distribution on manifolds. Therefore the Lebesgue measure defined on Euclidean space should be converted to Hausdorff measure, which is a fundamental concept in geometric measure theory. When the manifold can be embedded into $\mathbb{R}^d$, Hausdorff measure can be interpreted as the surface area on the manifold (Byrne and Girolami, 2013). The relation between Hausdorff measure and Lebesgue measure can

be formulated as follows. Let $\mathcal{H}^d$ be a $d$ dimensional Hausdorff measure and $\lambda^m$ be the Lebesgue measure on $\mathbb{R}^m$. If we could parameterize the manifold by a Lipschitz function $f : \mathbb{R}^m \to \mathbb{R}^d$, then for any $\mathcal{H}^d$ measurable function $g : \mathbb{R}^d \to \mathbb{R}$, we have following basic area equation

**Theorem 4.1.** *If $f : \mathbb{R}^m \to \mathbb{R}^d$ is Lipschitz and $m \leqslant d$, then*

$$\int_A g(f(x)) J_m f(x) \lambda^m(dx) = \int_{\mathbb{R}^d} g(y) |\{x \in A : f(x) = y\}| \mathcal{H}^d(dy). \tag{4.11}$$

Here $J_m f(x)$ is the $m$ dimensional Jacobian of $f$, and in this case it is defined as $(J_m f(x))^2 = |\boldsymbol{D}^\top \boldsymbol{D}|$, where $\boldsymbol{D} \in \mathbb{R}^{d \times m}$ is the derivative matrix with $d_{i,j} = \partial f_i(x)/\partial x_j, 1 \leqslant i \leqslant d, 1 \leqslant j \leqslant m$ (Federer, 1969; Diaconis et al., 2012).

The right hand integral is the surface area integral of $g$ over $f(A)$. The left hand integral shows the surface area integral can be related to the integral of Lebesgue measure in $\mathbb{R}^m$ and the Jacobian. Sampling from the density of normalized $J_m f(x)$ on $\mathbb{R}^m$ and mapping back to the manifold via $f$ gives samples from the area measure (Diaconis et al., 2012).

We use an example studied by Diaconis et al. (2012) to provide an application of (4.11). The Torus manifold

$$\mathcal{M} = \left\{ \Big( [R + r\cos(\theta)]\cos(\varphi), [R + r\cos(\theta)]\sin(\varphi), r\sin(\theta) \Big)^\top \right\}$$

with $0 \leqslant \theta, \varphi \leqslant 2\pi$ for fixed $R > r > 0$ is a two dimensional manifold in $\mathbb{R}^3$. The area of the Torus is $(2\pi)^2 Rr$ therefore the uniform density with Hausdorff area measure is $1/[(2\pi)^2 Rr]$. The Lipschitz function $f : \mathbb{R}^2 \to \mathbb{R}^3$ is

$$f(\theta, \varphi) = \Big( [R + r\cos(\theta)]\cos(\varphi), [R + r\cos(\theta)]\sin(\varphi), r\sin(\theta) \Big)^\top.$$

The derivative matrix $\boldsymbol{D}$ has following form

$$\boldsymbol{D} = \begin{pmatrix} -r\sin(\theta)\cos(\varphi) & -(R + r\cos(\theta))\sin(\varphi) \\ -r\sin(\theta)\sin(\varphi) & (R + r\cos(\theta))\cos(\varphi) \\ r\cos(\theta) & 0 \end{pmatrix}.$$

Therefore the Jacobian $J_2(f(x))^2 = r^2(R+r\cos(\theta))^2$. Thus the corresponding density of $\theta, \varphi$ with Lebesgue measure is

$$p(\theta, \varphi) \propto \frac{1}{4\pi^2}\big(1 + \frac{r}{R}\cos(\theta)\big).$$

Sampling $\theta, \varphi$ from this density and mapping back to $\mathbb{R}^3$ gives the uniform distribution on the Torus manifold.

### *4.3.2 Geodesic Riemann manifold Hamiltonian Monte Carlo*

We now state a Riemann manifold Hamiltonian Monte Carlo method on the $\mathbb{S}^{d_j-1}$ sphere to draw posterior samples of $\boldsymbol{x}_{jh}$ from (4.9). Neal (2011) gives a detailed review of Hamiltonian Monte Carlo (HMC) and Girolami and Calderhead (2011) develops HMC methods on Riemann manifold. We first introduce Hamiltonian Monte Carlo method.

*Hamiltonian Monte Carlo*

Let $\ell(\boldsymbol{\theta})$ be the log density of parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. HMC methods define a Hamiltonian by introducing a kinetic energy term with momentum variables $\boldsymbol{q} \in \mathbb{R}^d$. The Hamiltonian is defined as

$$H(\boldsymbol{\theta}, \boldsymbol{q}) = -\ell(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{q}^\top \boldsymbol{G}^{-1}\boldsymbol{q}. \tag{4.12}$$

From physics point of view, the equation can be thought as a sum of a potential energy term ($-\ell(\boldsymbol{\theta})$) defined at position $\boldsymbol{\theta}$ and a kinetic energy term ($\frac{1}{2}\boldsymbol{q}^\top \boldsymbol{G}^{-1}\boldsymbol{q}$) defined by the momentum variable $\boldsymbol{q}$. It also can be viewed proportional to a log joint density of $\boldsymbol{\theta}$ and auxiliary variable $\boldsymbol{q}$ with

$$H(\boldsymbol{\theta}, \boldsymbol{q}) \propto \log\big(p(\boldsymbol{\theta}, \boldsymbol{q})\big) = \log\big(p(\boldsymbol{\theta})p(\boldsymbol{q})\big),$$

with $p(\boldsymbol{q}) \overset{d}{=} \mathrm{N}(\boldsymbol{0}, \boldsymbol{G})$. The Hamilton's equations for the system (4.12) are

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \frac{\partial H(\boldsymbol{\theta}, \boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{G}^{-1}\boldsymbol{q},$$

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = -\frac{\partial H(\boldsymbol{\theta}, \boldsymbol{q})}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}). \tag{4.13}$$

One can develop a HMC method using following steps to update $(\boldsymbol{\theta}, \boldsymbol{q})$ jointly.

- Draw $\boldsymbol{q}$ from $\mathrm{N}(\mathbf{0}, \boldsymbol{G})$. Due to factorization of $(\boldsymbol{\theta}, \boldsymbol{q})$, this is equivalent to draw from $\boldsymbol{q}|\boldsymbol{\theta}$.

- Perform following Hamiltonian dynamics to propose a new state $(\boldsymbol{\theta}^*, \boldsymbol{q}^*)$

$$\begin{aligned} \boldsymbol{q}' &= \boldsymbol{q} + \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}), \\ \boldsymbol{\theta}^* &= \boldsymbol{\theta} + \epsilon \boldsymbol{G}^{-1} \boldsymbol{q}', \\ \boldsymbol{q}^* &= \boldsymbol{q}' + \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}^*). \end{aligned} \tag{4.14}$$

  Here $\epsilon$ is the integration step size. Above equations are also know as the leapfrog integrator.

- Accept $(\boldsymbol{\theta}^*, \boldsymbol{q}^*)$ with the Metropolis ratio. The Hamiltonian dynamics in (4.14) keep the total energy approximately invariant (errors are generated due to numerical integration), and the Metropolis step corrects this error. Due to the volume preserving property of the Hamiltonian trajectory, Hastings ratio is not needed (Neal, 2011; Girolami and Calderhead, 2011).

Above steps can be viewed as a Gibbs sampler by drawing conditional distributions from the Hamiltonian in (4.12) (Girolami and Calderhead, 2011). Repeating above steps defines reversible moves on joint parameter space of $\boldsymbol{\theta}$ and $\boldsymbol{q}$ and it keeps the Hamiltonian $H(\boldsymbol{\theta}, \boldsymbol{q})$ invariant. Independent draws from $\boldsymbol{q}$ allows HMC to perform large moves in $(\boldsymbol{\theta}, \boldsymbol{q})$. Therefore, HMC allows large moves in $\boldsymbol{\theta}$ and enhances the mixing behavior of the Markov chain. Due to factorization of the joint density, the Markov chain by discarding $\boldsymbol{q}$ targets the distribution $p(\boldsymbol{\theta})$ (Neal, 2011).

The numerical integration methods used in Hamiltonian dynamics have direct connections with Langevin diffusion with stochastic differential equation defined as

$$\mathrm{d}\boldsymbol{\theta}(t) = \frac{1}{2}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta})\mathrm{d}t + \mathrm{d}\boldsymbol{b}(t),$$

where $\boldsymbol{b}(t)$ is a $d$ dimensional Brownian motion (Neal, 2011; Girolami and Calderhead, 2011). The parameter space with the Hamiltonian dynamics (4.13) is endowed with Euclidean space with identity metric by using gradient of the log density.

The generalization of the Euclidean space to a Riemannian manifold has lead to using other metric tensors in the parameter space (Efron, 1975; Amari, 1998; Raskutti and Mukherjee, 2015). In particular, when the parameter $\boldsymbol{\theta}$ is in a Riemannian manifold with metric tensor of $\boldsymbol{G}(\boldsymbol{\theta})$, the Hamiltonian becomes (Girolami and Calderhead, 2011; Byrne and Girolami, 2013)

$$H(\boldsymbol{\theta}, \boldsymbol{q}) = -\ell(\boldsymbol{\theta}) + \frac{1}{2}\log\left(|\boldsymbol{G}(\boldsymbol{\theta})|\right) + \frac{1}{2}\boldsymbol{q}^\top \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{q}. \tag{4.15}$$

The density function in $\ell(\boldsymbol{\theta})$ is defined with respect to Lebesgue measure in some coordinate system $\boldsymbol{\theta}$. The log determinant term on the right hand side of the equation is introduced to satisfy that the marginal density $\boldsymbol{\theta}$ equals to the target. This Hamiltonian can be viewed as letting auxiliary variable $\boldsymbol{q} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{G}(\boldsymbol{\theta}))$. Byrne and Girolami (2013) notice that the first two terms on the right hand side of (4.15) can be combined to generate a density function with respect to Hausdorff measure $\ell_{\mathcal{H}}(\boldsymbol{\theta})$. The resulting Hamiltonian becomes

$$H(\boldsymbol{\theta}, \boldsymbol{q}) = -\ell_{\mathcal{H}}(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{q}^\top \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{q}. \tag{4.16}$$

This is due to equation (4.11) in Theorem 4.1. The Hamiltonian dynamics in (4.13) become

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \frac{\partial H(\boldsymbol{\theta}, \boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{q},$$

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = -\frac{\partial H(\boldsymbol{\theta}, \boldsymbol{q})}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \left( \ell_{\mathcal{H}}(\boldsymbol{\theta}) - \frac{1}{2} \boldsymbol{q}^\top \boldsymbol{G}(\boldsymbol{\theta})^{-1} \boldsymbol{q} \right). \tag{4.17}$$

This Hamiltonian is not separable therefore original leapfrog method does not apply.

*Geodesic integrator*

Byrne and Girolami (2013) construct an integrator by splitting the Hamiltonian in (4.17) to a potential and kinetic term

$$H^{(1)}(\boldsymbol{\theta}, \boldsymbol{q}) = -\ell_{\mathcal{H}}(\boldsymbol{\theta}), \tag{4.18}$$

$$H^{(2)}(\boldsymbol{\theta}, \boldsymbol{q}) = \frac{1}{2} \boldsymbol{q}^\top \boldsymbol{G}(\boldsymbol{\theta})^{-1} \boldsymbol{q}. \tag{4.19}$$

The potential term in (4.18) does not involve $\boldsymbol{q}$ therefore $\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t = \mathbf{0}$. $\mathrm{d}\boldsymbol{q}/\mathrm{d}t$ is simply $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{H}}(\boldsymbol{\theta})$. For the kinetic term in (4.19), it is an Hamiltonian without potential term. Byrne and Girolami (2013) show that the solution in the dynamics is geodesic flow with metric tensor $\boldsymbol{G}(\boldsymbol{\theta})$. In summary the Hamiltonian dynamics for (4.17) become

- Update $\boldsymbol{q}$ according to the solution in (4.18) for $\epsilon/2$

$$\boldsymbol{q}' = \boldsymbol{q} + \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{H}}(\boldsymbol{\theta}), \tag{4.20}$$

- Update $(\boldsymbol{\theta}, \boldsymbol{q})$ according to (4.19) following geodesic flow for $\epsilon$,
- Update $\boldsymbol{q}$ again using (4.20) for $\epsilon/2$.

*Manifold HMC with embedding coordinates*

We re-write the Riemannian manifold HMC in (4.16) by coordinate embedding (Byrne and Girolami, 2013). Given an isometric embedding $\xi : \mathcal{M} \to \mathbb{R}^d$, a motion on the original space $\boldsymbol{\theta}(t)$ with $t > 0$ has an image $\boldsymbol{x}(t)$ on $\mathbb{R}^d$ with $\boldsymbol{x}(t) = \xi[\boldsymbol{\theta}(t)]$. In addition, we have

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \sum_j \frac{\partial x_i}{\partial \theta_j} \frac{\mathrm{d}\theta_j(t)}{\mathrm{d}t},$$

where we let subscript $j$ index the coordinate of $\boldsymbol{\theta}$ and subscript $i$ index the coordinate of $\boldsymbol{x}$. Therefore we get $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{M}\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t$ with $\boldsymbol{M}$ is the derivative matrix defined in (4.10) and $\boldsymbol{G} = \boldsymbol{M}^\top\boldsymbol{M}$. Then if we transform $(\boldsymbol{\theta}, \boldsymbol{q})$ to the embedding space $(\boldsymbol{x}, \boldsymbol{v})$ with $\boldsymbol{v} = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$, we get

$$\boldsymbol{v} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{M}\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \boldsymbol{M}(\boldsymbol{M}^\top\boldsymbol{M})^{-1}\boldsymbol{q}.$$

Furthermore, if we make the transformation, the original Hamiltonian (4.16) becomes (Byrne and Girolami, 2013)

$$H(\boldsymbol{x}, \boldsymbol{v}) = -\ell_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{v}^\top\boldsymbol{v}. \tag{4.21}$$

According to Byrne and Girolami (2013) there is no additional Jacobian term introduced because $\ell_{\mathcal{H}}(\boldsymbol{x})$ is still defined with respect to Hausdorff measure. With this transformation, the solution to the potential term in (4.18) can be written in the new coordinates by change of variables of the operator $\nabla_{\boldsymbol{\theta}} = \boldsymbol{M}^\top\nabla_{\boldsymbol{x}}$

$$\boldsymbol{v}' = \boldsymbol{v} + \frac{\epsilon}{2}\boldsymbol{M}(\boldsymbol{M}^\top\boldsymbol{M})^{-1}\boldsymbol{M}^\top\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x}).$$

The term $\boldsymbol{M}(\boldsymbol{M}^\top\boldsymbol{M})^{-1}\boldsymbol{M}^\top$ is the orthogonal projection matrix by projecting the gradient of $\ell(\boldsymbol{x})$ to the column space of $\boldsymbol{M}$. As shown by Byrne and Girolami (2013), this is the tangent space of the embedded manifold. In our problem, $\boldsymbol{x} \in \mathbb{S}^{d-1}$. For arbitrary vector $\boldsymbol{u}$, the projection of $\boldsymbol{u}$ to the tangent space of $\boldsymbol{x}$ is given by

$$\boldsymbol{u}_{Tx} = (\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{u}.$$

Therefore the solution to the potential term in (4.18) in the new coordinate becomes

$$\boldsymbol{v}' = \boldsymbol{v} + \frac{\epsilon}{2}(\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top)\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x}).$$

For the solution to the kinetic term in (4.19), we have a closed form solution due to our sphere parameterization. The geodesic flow on the sphere is shown to be

$$(\boldsymbol{x}(t), \boldsymbol{v}(t)) = (\boldsymbol{x}(0), \boldsymbol{v}(0))\begin{pmatrix}1 & 0\\ 0 & c^{-1}\end{pmatrix}\begin{pmatrix}\cos(ct) & -\sin(ct)\\ \sin(ct) & \cos(ct)\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & c\end{pmatrix}, \tag{4.22}$$

where $c = \boldsymbol{v}(t)^\top \boldsymbol{v}(t)$ is constant.

We summarize our geodesic Riemannian manifold HMC steps to draw posterior samples from distribution in (4.9) defined on $\mathbb{S}^{d-1}$ in Algorithm 4.1. Note that we do not need to evaluate the normalizing constant of the density $p(\boldsymbol{x}_{jh})$ in (4.9) due to following two reasons. First, when we calculate the gradient of the log density $\ell(\boldsymbol{x}_{jh})$, only the terms in the exponential function matter. Second, when we define the Hamiltonian in (4.15) and (4.16), the normalizing constant does not depends on either $\boldsymbol{\theta}$ or $\boldsymbol{q}$. Therefore it can be removed from the Hamiltonian.

**Algorithm 4.1:** Geodesic Riemannian manifold Hamiltonian Monte Carlo algorithm to draw samples from log density $\ell(\boldsymbol{x})$

**Input** : the number of steps $T$ for integration and step size
**Output**: One draw from density $\ell(\boldsymbol{x})$

$\boldsymbol{v} \sim \mathrm{N}_d(\boldsymbol{0}, \boldsymbol{I})$; $\boldsymbol{v} = (\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{v}$;
$l_0 = \ell(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{v}^\top\boldsymbol{v}$; $\boldsymbol{x}_0 = \boldsymbol{x}$;
**for** $t \leftarrow 1$ **to** $T$ **do**
  $\boldsymbol{v} = \boldsymbol{v} + \frac{\epsilon}{2}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x})$; $\boldsymbol{v} = (\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{v}$;
  Update $(\boldsymbol{x}, \boldsymbol{v})$ following geodesic flow in (4.22) for $\epsilon$;
  $\boldsymbol{v} = \boldsymbol{v} + \frac{\epsilon}{2}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x})$; $\boldsymbol{v} = (\boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{v}$;
**end**
$l_1 = \ell(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{v}^\top\boldsymbol{v}$;
$u \sim \mathrm{Unif}(0, 1)$;
**if** $u < \exp(l_1 - l_0)$ **then**
  return $\boldsymbol{x}$;
**else**
  return $\boldsymbol{x}_0$;
**end**

### *4.3.3 An additional example*

Other than the distribution defined in (4.9), we consider another distribution defined on a specific manifold in this subsection, the Bingham-von Mises-Fisher distribution. This distribution is defined on Stiefel manifold $\mathcal{V}(p, k)$, the set of orthonormal matrix $\boldsymbol{X}$ of dimension $p \times k$ $(k \leqslant p)$ such that $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{I}_k$. This Stiefel manifold is of

dimension $pk - p(p+1)/2$ and it is embedded in the Euclidean space $\mathbb{R}^{p\times k}$ (Absil et al., 2009). When $k = 1$ the Stiefel manifold becomes a sphere. The Bingham-von Mises-Fisher (BMF) distribution specifies an absolutely continuous density on the Stiefel manifold $\mathcal{V}(p, k)$. This distribution is studied by Bingham (1974) for a vector defined on sphere ($k = 1$) and has been considered for matrix form with $k > 1$ by Khatri and Mardia (1977). Recently several papers have investigated applications of such a distribution (Hoff, 2009a,b; Byrne and Girolami, 2013).

The BMF distribution is defined as

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}) &\propto \text{etr}(\boldsymbol{C}^\top \boldsymbol{X} + \boldsymbol{B}\boldsymbol{X}^\top \boldsymbol{A}\boldsymbol{X}) \\ &= \exp\left(\sum_h^k \boldsymbol{c}_h^\top \boldsymbol{x}_h\right) \exp\left(\sum_h^k b_h \boldsymbol{x}_h^\top \boldsymbol{A}\boldsymbol{x}_h\right), \end{aligned} \tag{4.23}$$

where $\boldsymbol{x}_h$ is the $h$th column of $\boldsymbol{X}$, $\boldsymbol{c}_h$ is the $h$th column of $\boldsymbol{C}$ and $b_h$ is the $h$th diagonal entry of $\boldsymbol{B}$. We assume $\boldsymbol{B}$ is a diagonal matrix and $\boldsymbol{A}$ is a symmetric matrix to satisfy the antipodally symmetric property (Hoff, 2009a). The density reduces to Bingham density when $\boldsymbol{C} = \boldsymbol{0}$ and von Mises-Fisher density when either $\boldsymbol{A}$ or $\boldsymbol{B}$ is zero. When $k = 1$ the density defines a distribution on unit sphere. Sampling from this distribution has many potential applications including covariance matrix estimation, orthogonal factor analysis, probabilistic singular value/eigen decomposition and network analysis (Hoff, 2009a,b; Zhong and Girolami, 2012).

To efficiently draw samples from the distribution in (4.23) has been shown a challenging task (Hoff, 2009b; Rao et al., 2014). We use the geodesic Riemannian manifold HMC introduced in previous subsection to draw samples from such a density (Byrne and Girolami, 2013). We introduce the momentum variable $\boldsymbol{V} \in \mathbb{R}^{p\times k}$. We require following two ingredients to implement a geodesic HMC sampler.

*Projection of gradient to tangent space* We first calculate the gradient of the log density as follows

$$\begin{aligned}\nabla_{\boldsymbol{X}} \log[p(\boldsymbol{X})] &= \frac{\partial \log[p(\boldsymbol{X})]}{\partial \boldsymbol{X}} = \frac{1}{p(\boldsymbol{X})}\frac{\partial p(\boldsymbol{X})}{\boldsymbol{X}} \\ &= \frac{1}{p(\boldsymbol{X})} p(\boldsymbol{X}) \frac{\partial \mathrm{tr}(\boldsymbol{C}^\top \boldsymbol{X} + \boldsymbol{B}\boldsymbol{X}^\top \boldsymbol{A}\boldsymbol{X})}{\partial \boldsymbol{X}} \\ &= \boldsymbol{C} + 2\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}.\end{aligned}$$

According to Byrne and Girolami (2013), the projection of arbitrary matrix $\boldsymbol{V} \in \mathbb{R}^{p\times k}$ to the tangent space of $\boldsymbol{X}$ can be written as

$$\boldsymbol{V}_{Tx} = \boldsymbol{V} - \frac{1}{2}\boldsymbol{X}\big(\boldsymbol{X}^\top \boldsymbol{V} + \boldsymbol{V}^\top \boldsymbol{X}\big).$$

*Geodesic flow on Stiefel manifold* The geodesic flow has been shown to have following form (Edelman et al., 1998; Byrne and Girolami, 2013)

$$(\boldsymbol{X}(t), \boldsymbol{V}(t)) = (\boldsymbol{X}(0), \boldsymbol{V}(0)) \exp\left(t \begin{pmatrix} \boldsymbol{A} & -\boldsymbol{S}(0) \\ \boldsymbol{I} & \boldsymbol{A} \end{pmatrix}\right) \begin{pmatrix} \exp(-t\boldsymbol{A}) & \boldsymbol{0} \\ \boldsymbol{0} & \exp(-t\boldsymbol{A}) \end{pmatrix}, \tag{4.24}$$

where $\boldsymbol{A} = \boldsymbol{X}(t)^\top \boldsymbol{V}(t)$ is a skew-symmetric matrix and it is constant over the geodesic. $\boldsymbol{S}(0) = \boldsymbol{V}(0)^\top \boldsymbol{V}(0)$ and $\exp(\cdot)$ is the matrix exponential function.

With the two ingredients, we could define following algorithm to draw samples from the Bingham-von Mises-Fisher distribution

## 4.4 Simulations

### *4.4.1 Bayesian GMM in MELD*

In this subsection we apply our Bayesian GMM approach developed for MELD in two simulation studies performed in Chapter 3. For first simulation study, we consider the low dimensional setting with $p = 20$ and apply our approach with both pseudo-likelihood functions in (4.1) and (4.2). For the second simulation study we consider

**Algorithm 4.2:** Geodesic Riemannian manifold Hamiltonian Monte Carlo algorithm to draw samples from Bingham-von Mises-Fisher distribution (4.23)

**Input** : the number of steps $T$ for integration and step size
**Output**: One draw from density in (4.23)

$\boldsymbol{V} \sim \mathrm{N}_{p\times k}(\boldsymbol{0}, \boldsymbol{I})$; $\boldsymbol{V} = \boldsymbol{V} - \frac{1}{2}\boldsymbol{X}\big(\boldsymbol{X}^\top\boldsymbol{V} + \boldsymbol{V}^\top\boldsymbol{X}\big)$;
$l_0 = \log[p(\boldsymbol{X})] - \frac{1}{2}\mathrm{vec}(\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{V})$; $\boldsymbol{X}_0 = \boldsymbol{X}$;
**for** $t \leftarrow 1$ **to** $T$ **do**
  $\boldsymbol{V} = \boldsymbol{V} + \frac{\epsilon}{2}\nabla_{\boldsymbol{X}} \log[p(\boldsymbol{X})]$; $\boldsymbol{V} = \boldsymbol{V} - \frac{1}{2}\boldsymbol{X}\big(\boldsymbol{X}^\top\boldsymbol{V} + \boldsymbol{V}^\top\boldsymbol{X}\big)$;
  Update $(\boldsymbol{X}, \boldsymbol{V})$ following geodesic flow in (4.24) for $\epsilon$;
  $\boldsymbol{V} = \boldsymbol{V} + \frac{\epsilon}{2}\nabla_{\boldsymbol{X}} \log[p(\boldsymbol{X})]$; $\boldsymbol{V} = \boldsymbol{V} - \frac{1}{2}\boldsymbol{X}\big(\boldsymbol{X}^\top\boldsymbol{V} + \boldsymbol{V}^\top\boldsymbol{X}\big)$;
**end**
$l_1 = \log[p(\boldsymbol{X})] - \frac{1}{2}\mathrm{vec}(\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{V})$;
$u \sim \mathrm{Unif}(0, 1)$;
**if** $u < \exp(l_1 - l_0)$ **then**
  return $\boldsymbol{X}$;
**else**
  return $\boldsymbol{X}_0$;
**end**

the mixed data type setting with categorical, Gaussian and Poisson variables with $p = 100$. Pseudo-likelihood (4.1) with only second moment matrices is used in this case. For all the simulation studies, we set the Dirichlet hyperparameter $\beta_c$ for the prior of $\boldsymbol{\phi}_{jh}$ to 0.5 when $j$th variable is categorical. The hyperparameters of prior for $\phi_{jh}$ for non-categorical variables are set to $\mu_0 = 0$ and $\sigma_0 = 10$. $a_\tau$ and $b_\tau$ are set to 1 and 0.3 respectively. This configuration corresponds to letting $(\tau^{(\cdot)})^{-1}$ to have a prior mean of 3.3 and variance of 11.1. For the geodesic sampler, the integration step size is set to $\epsilon = 0.01$ and the number of steps in the numerical integration is set to 10 for the pseudo-likelihood in (4.1) and 5 for the pseudo-likelihood in (4.2).

*Low dimensional categorical simulations* The process of generating data in this simulation is the same as in Chapter 3. Each of the $p = 20$ categorical variables has $d = 4$ levels. The number of components is set to $k = 3$ and we simulate $n = \{50, 100, 200, 500, 1,000\}$ samples. We contaminate 4% and 10% samples to as-

sess the robustness of our methods. We run our MCMC algorithm on the simulated data for $10,000$ iterations. We set the first $5,000$ as burn-in period and collect a posterior sample every 50 iterations. We use the same method in Chapter 3 to calculate the MSE of estimated mean parameters of $y_{ij}$'s and their true parameters by recovering membership variables (Table 4.1, 4.2, 4.3). Our Bayesian GMM approach with the likelihood $L^{(2)}(\mathbf{\Phi})$ has better MSE's in most of the cases. The log likelihood trajectories with $n = 1,000$ are shown in Figure 4.1 and 4.2. The Markov chains mix well for the likelihood $L^{(2)}(\mathbf{\Phi})$. However with the likelihood of $L^{(3)}(\mathbf{\Phi})$ and $k = 1$ and $k = 2$ the chains do not show a good mixing (Figure 4.2).

Table 4.1: Comparison of mean squared error (MSE) of estimated parameters in categorical simulation with small $p$ using Bayesian GMM method and GMM in MELD. For GMM estimation its standard deviations of MSE's are calculated from ten simulated data sets for each value of $n$. For the Bayesian GMM method the standard deviations of MSE's are calculated using posterior mean estimates of the ten simulated data sets.

| **Methods** | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | GMM $Q^{(3)}(\mathbf{\Phi})$ | | BGMM $L^{(2)}(\mathbf{\Phi})$ | BGMM $L^{(3)}(\mathbf{\Phi})$ |
|---|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage | 2nd stage | 1st stage | 2nd stage | MSE | MSE |
| 50 | 1 | 0.043(0.002) | **0.042(0.002)** | 0.044(0.002) | 0.084(0.040) | 0.042(0.002) | |
| | 2 | 0.035(0.001) | 0.046(0.010) | 0.035(0.002) | 0.075(0.035) | **0.032(0.001)** | |
| | 3 | 0.036(0.002) | 0.038(0.002) | 0.037(0.002) | 0.074(0.034) | **0.029(0.001)** | |
| | 4 | 0.041(0.002) | 0.044(0.002) | 0.042(0.003) | 0.044(0.002) | **0.032(0.002)** | |
| | 5 | 0.045(0.002) | 0.046(0.002) | 0.044(0.002) | 0.048(0.002) | **0.038(0.005)** | |
| 100 | 1 | **0.041(0.001)** | 0.041(0.001) | 0.042(0.001) | 0.068(0.045) | 0.041(0.001) | |
| | 2 | 0.031(0.002) | 0.033(0.005) | 0.031(0.001) | 0.044(0.029) | **0.030(0.001)** | |
| | 3 | 0.033(0.002) | 0.034(0.002) | 0.033(0.002) | 0.050(0.030) | **0.028(0.001)** | |
| | 4 | 0.037(0.002) | 0.038(0.002) | 0.038(0.001) | 0.039(0.002) | **0.031(0.001)** | |
| | 5 | 0.041(0.003) | 0.041(0.003) | 0.039(0.002) | 0.043(0.002) | **0.037(0.004)** | |
| 200 | 1 | **0.041(<0.001)** | 0.041(<0.001) | 0.042(<0.001) | 0.041(<0.001) | 0.041(<0.001) | |
| | 2 | 0.030(0.001) | 0.030(0.001) | **0.029(0.001)** | 0.029(0.001) | 0.029(0.001) | |
| | 3 | 0.033(0.001) | 0.033(0.001) | 0.032(0.001) | 0.033(0.001) | **0.028(0.001)** | |
| | 4 | 0.036(0.001) | 0.037(0.001) | 0.036(0.002) | 0.037(0.001) | **0.031(0.001)** | |
| | 5 | 0.038(<0.001) | **0.036(0.001)** | 0.038(0.001) | 0.039(0.001) | 0.037(0.004) | |
| 500 | 1 | **0.041(<0.001)** | 0.041(<0.001) | 0.041(<0.001) | 0.041(<0.001) | 0.041(<0.001) | |
| | 2 | **0.029(0.001)** | 0.030(0.001) | 0.029(0.001) | 0.029(0.001) | 0.029(0.001) | |
| | 3 | 0.032(0.001) | 0.032(0.001) | 0.032(0.001) | 0.032(0.001) | **0.029(0.001)** | |
| | 4 | 0.035(0.001) | 0.036(0.001) | 0.035(0.001) | 0.036(0.001) | **0.030(0.001)** | |
| | 5 | 0.036(0.001) | **0.035(0.001)** | 0.036(0.001) | 0.038(0.001) | 0.036(0.003) | |
| 1,000 | 1 | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) | **0.041(<0.001)** | |
| | 2 | 0.029(0.001) | 0.029(0.001) | 0.028(0.001) | 0.028(0.001) | **0.028(<0.001)** | |
| | 3 | 0.031(0.001) | 0.031(0.005) | 0.031(0.001) | 0.031(0.001) | **0.029(<0.001)** | |
| | 4 | 0.034(0.001) | 0.034(0.001) | 0.034(0.001) | 0.034(0.001) | **0.030(<0.001)** | |
| | 5 | 0.036(0.001) | **0.034(0.001)** | 0.036(0.001) | 0.037(0.001) | 0.036(0.003) | |

We further monitor the objective functions of $Q_n^{(2)}(\mathbf{\Phi}, \boldsymbol{I})$ and $Q_n^{(3)}(\mathbf{\Phi}, \boldsymbol{I})$ defined in Chapter 3 equation (3.14) and (3.15). With likelihood $L^{(2)}(\mathbf{\Phi})$ the objective function

Table 4.2: Comparison of mean squared error (MSE) of estimated parameters in categorical simulation with small $p$ using Bayesian GMM method and GMM in MELD. The simulated ten data sets are contaminated by setting 4% of the samples to outliers. The MSE's are calculated with the same methods in Table 4.1.

| **Methods** | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | GMM $Q^{(3)}(\mathbf{\Phi})$ | | BGMM $L^{(2)}(\mathbf{\Phi})$ |
|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage | 2nd stage | 1st stage | 2nd stage | MSE |
| 50 | 1 | 0.043(0.002) | 0.043(0.002) | 0.044(0.002) | 0.061(0.024) | **0.042(0.002)** |
| | 2 | 0.036(0.001) | 0.039(0.005) | 0.036(0.003) | 0.058(0.038) | **0.034(0.001)** |
| | 3 | 0.037(0.002) | 0.038(0.002) | 0.037(0.002) | 0.052(0.021) | **0.031(0.001)** |
| | 4 | 0.041(0.001) | 0.044(0.002) | 0.043(0.002) | 0.050(0.007) | **0.035(0.002)** |
| | 5 | 0.046(0.001) | 0.048(0.002) | 0.044(0.002) | 0.050(0.003) | **0.039(0.004)** |
| 100 | 1 | **0.041(0.001)** | 0.041(0.001) | 0.042(0.001) | 0.041(0.002) | 0.041(0.001) |
| | 2 | **0.032(0.001)** | 0.032(0.002) | 0.032(0.002) | 0.032(0.002) | 0.032(0.001) |
| | 3 | 0.033(0.002) | 0.034(0.002) | 0.033(0.002) | 0.036(0.008) | **0.030(0.001)** |
| | 4 | 0.038(0.001) | 0.040(0.001) | 0.038(0.002) | 0.040(0.003) | **0.033(0.001)** |
| | 5 | 0.043(0.002) | 0.044(0.002) | 0.043(0.003) | 0.046(0.003) | **0.040(0.004)** |
| 200 | 1 | **0.041(<0.001)** | 0.041(<0.001) | 0.042(<0.001) | 0.048(0.020) | 0.041(<0.001) |
| | 2 | **0.031(0.001)** | 0.031(0.001) | 0.030(0.001) | 0.032(0.005) | 0.031(0.001) |
| | 3 | 0.033(0.001) | 0.033(0.001) | 0.032(0.001) | 0.033(0.001) | **0.030(0.001)** |
| | 4 | 0.038(0.002) | 0.039(0.002) | 0.038(0.002) | 0.039(0.002) | **0.033(0.001)** |
| | 5 | 0.041(0.001) | 0.041(0.002) | 0.042(0.003) | 0.044(0.003) | **0.039(0.004)** |
| 500 | 1 | **0.041(0.001)** | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) |
| | 2 | **0.030(0.001)** | 0.031(0.001) | 0.029(0.001) | 0.030(0.001) | 0.031(<0.001) |
| | 3 | 0.032(0.001) | 0.033(0.001) | 0.032(0.001) | 0.032(0.001) | **0.031(<0.001)** |
| | 4 | 0.037(0.001) | 0.038(0.001) | 0.038(0.001) | 0.039(0.001) | **0.033(<0.001)** |
| | 5 | 0.040(0.001) | 0.040(0.001) | 0.043(0.003) | 0.045(0.002) | **0.037(0.003)** |
| 1,000 | 1 | **0.041(<0.001)** | 0.041(<0.001) | 0.041(<0.001) | 0.041(<0.001) | 0.041(<0.001) |
| | 2 | **0.030(0.001)** | 0.030(0.001) | 0.029(<0.001) | 0.030(<0.001) | 0.031(<0.001) |
| | 3 | **0.031(<0.001)** | 0.032(<0.001) | 0.031(<0.001) | 0.031(<0.001) | 0.031(<0.001) |
| | 4 | 0.037(0.001) | 0.038(0.001) | 0.038(0.001) | 0.039(0.001) | **0.032(0.001)** |
| | 5 | 0.040(<0.001) | **0.039(0.001)** | 0.043(0.002) | 0.044(0.002) | **0.038(0.002)** |

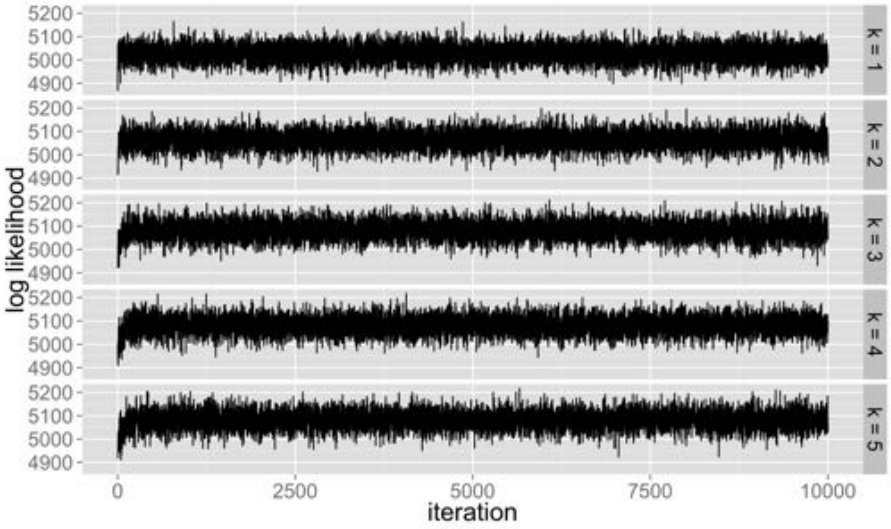


FIGURE 4.1: The trajectories of the log likelihood functions $L^{(2)}(\mathbf{\Phi})$ for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$.

Table 4.3: Comparison of mean squared error (MSE) of estimated parameters in categorical simulation with small $p$ using Bayesian GMM method and GMM in MELD. The simulated ten data sets are contaminated by setting 10% of the samples to outliers. The MSE's are calculated with the same methods in Table 4.1.

| **Methods** | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | GMM $Q^{(3)}(\mathbf{\Phi})$ | | BGMM $L^{(2)}(\mathbf{\Phi})$ |
|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage | 2nd stage | 1st stage | 2nd stage | MSE |
| 50 | 1 | 0.044(0.002) | 0.044(0.001) | 0.044(0.002) | 0.057(0.017) | **0.043(0.002)** |
| | 2 | 0.041(0.003) | 0.044(0.006) | 0.045(0.006) | 0.060(0.025) | **0.040(0.001)** |
| | 3 | 0.045(0.003) | 0.046(0.003) | 0.052(0.006) | 0.068(0.031) | **0.039(0.002)** |
| | 4 | 0.048(0.003) | 0.051(0.003) | 0.056(0.003) | 0.061(0.008) | **0.040(0.003)** |
| | 5 | 0.053(0.002) | 0.059(0.006) | 0.057(0.003) | 0.062(0.004) | **0.041(0.002)** |
| 100 | 1 | **0.042(0.002)** | 0.043(0.003) | 0.042(0.002) | 0.044(0.006) | 0.042(0.002) |
| | 2 | **0.040(0.002)** | 0.040(0.002) | 0.050(0.005) | 0.047(0.005) | 0.040(0.002) |
| | 3 | 0.044(0.002) | 0.044(0.002) | 0.054(0.003) | 0.054(0.004) | **0.041(0.001)** |
| | 4 | 0.047(0.002) | 0.049(0.002) | 0.056(0.002) | 0.058(0.003) | **0.042(0.002)** |
| | 5 | 0.051(0.003) | 0.054(0.003) | 0.055(0.004) | 0.059(0.004) | **0.042(0.002)** |
| 200 | 1 | **0.043(0.001)** | 0.043(0.001) | 0.042(0.001) | 0.043(<0.001) | 0.043(0.001) |
| | 2 | **0.039(0.001)** | 0.039(0.001) | 0.050(0.002) | 0.044(0.001) | 0.039(0.001) |
| | 3 | 0.041(0.002) | 0.042(0.001) | 0.052(0.001) | 0.052(0.002) | **0.039(0.001)** |
| | 4 | 0.047(0.002) | 0.048(0.002) | 0.056(0.002) | 0.057(0.002) | **0.039(0.001)** |
| | 5 | 0.050(0.002) | 0.053(0.002) | 0.054(0.002) | 0.058(0.002) | **0.042(0.001)** |
| 500 | 1 | **0.042(0.001)** | 0.042(0.001) | 0.042(0.001) | 0.042(0.001) | 0.042(0.001) |
| | 2 | **0.039(0.001)** | 0.039(0.001) | 0.050(0.001) | 0.044(0.001) | 0.039(0.001) |
| | 3 | 0.040(0.001) | 0.040(0.001) | 0.051(0.001) | 0.051(0.001) | **0.038(0.001)** |
| | 4 | 0.046(0.001) | 0.048(0.001) | 0.056(0.001) | 0.057(0.001) | **0.039(0.001)** |
| | 5 | 0.050(0.002) | 0.053(0.001) | 0.054(0.001) | 0.058(0.001) | **0.041(0.002)** |
| 1,000 | 1 | **0.042(<0.001)** | 0.042(<0.001) | 0.042(<0.001) | 0.042(<0.001) | 0.043(<0.001) |
| | 2 | 0.040(<0.001) | **0.039(<0.001)** | 0.050(0.001) | 0.045(0.001) | 0.039(0.001) |
| | 3 | 0.039(<0.001) | 0.040(<0.001) | 0.051(<0.001) | 0.051(0.001) | **0.038(0.001)** |
| | 4 | 0.047(<0.001) | 0.048(<0.001) | 0.056(0.001) | 0.057(0.001) | **0.039(0.001)** |
| | 5 | 0.050(0.001) | 0.052(0.001) | 0.054(0.001) | 0.058(0.001) | **0.043(0.001)** |

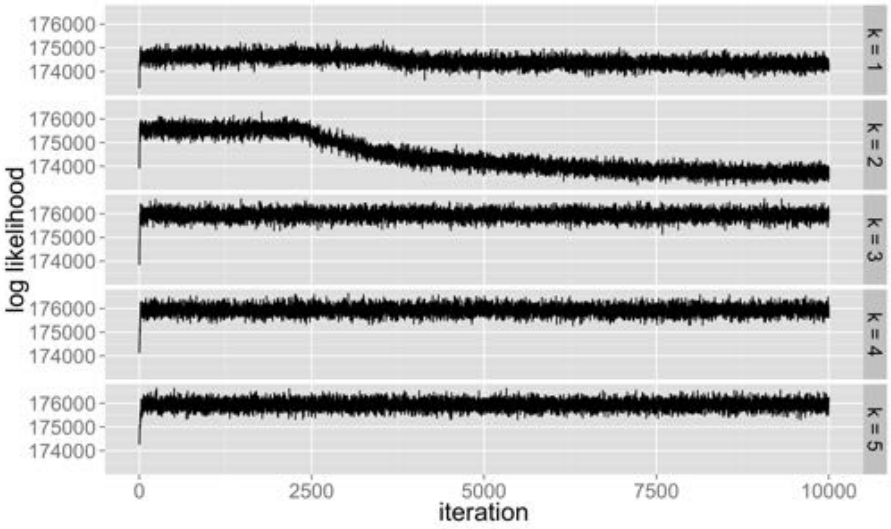


FIGURE 4.2: The trajectories of the log likelihood functions $L^{(3)}(\mathbf{\Phi})$ for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$.

of $Q_n^{(2)}(\mathbf{\Phi}, \boldsymbol{I})$ decreases fast in the first several iterations. Then the objective stays relatively stable. The minimum value is achieved with $k = 3$ (Figure 4.3). With the likelihood $L^{(3)}(\mathbf{\Phi})$, the objective function $Q_n^{(3)}(\mathbf{\Phi}, \boldsymbol{I})$ also decreases dramatically in the first few iterations. Moreover we observe that when the value of $k$ is less than the correct value 3, the objective function diverges after a number of iterations. This results echo their poor mixing behaviors shown in the likelihood trajectories (Figure 4.2). With larger values of $k$, the objective function becomes stable. The minimum value of the objective function is also achieved with the correct value of $k = 3$ (Figure 4.4). The posterior trajectories of one component parameter $\boldsymbol{\phi}_{jh}$ with the two likelihood functions are also plotted in Figure 4.5 and Figure 4.6 respectively.

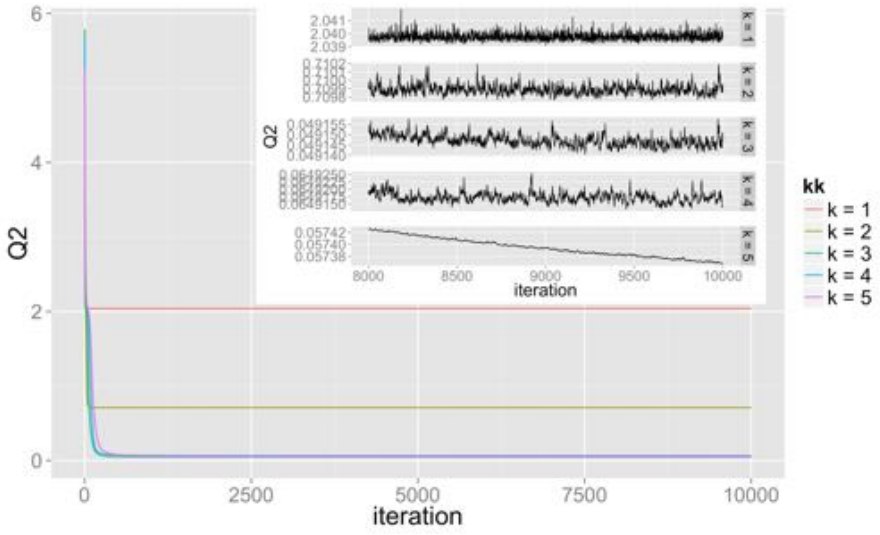


FIGURE 4.3: The trajectories of $Q_n^{(2)}(\mathbf{\Phi}, \boldsymbol{I})$ defined in Chapter 3 equation (3.14) for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$. The last $2,000$ iterations are plotted in the zoomed panel.

*Mixed data types with categorical, Gaussian and Poisson variables* We simulate 100 variables with first 95 being categorical variables with $d = 4$ levels, two Gaussian variables and three Poisson variables. $k$ is set to 2 and $n = \{50, 100, 200, 500, 1,000\}$ samples are generated. We run our MCMC algorithm with likelihood $L^{(2)}(\mathbf{\Phi})$ only

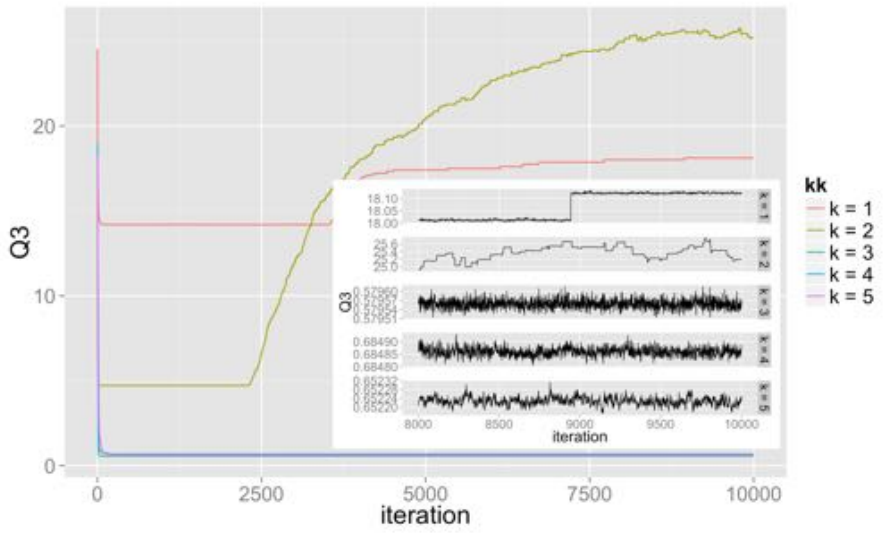


FIGURE 4.4: The trajectories of $Q_n^{(3)}(\boldsymbol{\Phi}, \boldsymbol{I})$ defined in Chapter 3 equation (3.15) for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$. The last $2,000$ iterations are plotted in the zoomed panel.

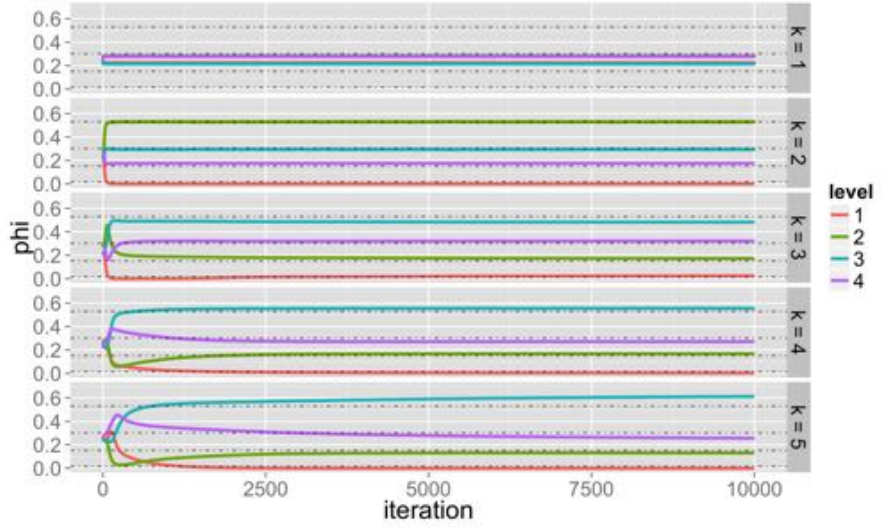


FIGURE 4.5: The trajectories of posterior draws of one component parameter $\boldsymbol{\phi}_{jh}$ with likelihood $L^{(2)}(\boldsymbol{\Phi})$ for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$. Different coordinates in $\boldsymbol{\phi}_{jh}$ are shown by different colors. True values are plotted as dotted lines.

on the data for $10,000$ iterations with first $5,000$ iterations as burn-in. Posterior samples are collected every 50 iterations after the burn-in period. We calculate the

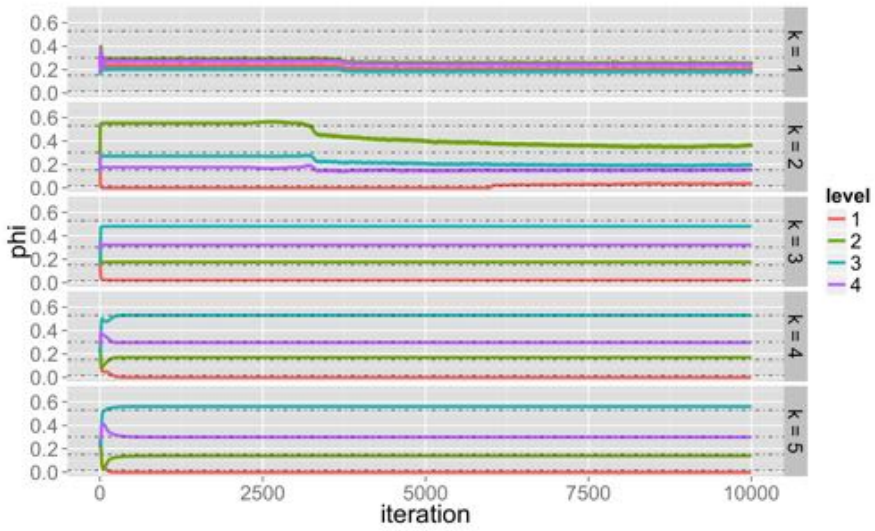


FIGURE 4.6: The trajectories of posterior draws of one component parameter $\boldsymbol{\phi}_{jh}$ with likelihood $L^{(3)}(\boldsymbol{\Phi})$ for the low dimensional categorical simulation with $n = 1,000$ under different values of $k$. Different coordinates in $\boldsymbol{\phi}_{jh}$ are shown by different colors. True values are plotted as dotted lines.

MSE of estimated mean parameters of $y_{ij}$'s and their true parameters by recovering their membership variables (Table 4.4). For non-categorical data squared Euclidean distance is used to recover membership variable using equation (3.23). The Bayesian GMM approach achieves smallest MSE's for categorical data under different values of $n$ when $k > 1$. For $k = 1$ the GMM first stage estimation has the smallest MSE's. For Gaussian variables the Bayesian GMM approach has smallest MSE's under small values of $n$ (50, 100 and 200) with the correct value of $k = 2$. When $n$ is large, GMM with second stage estimation outperforms alternatives with $k = 2$. For Poisson variables, the Bayesian GMM approach consistently has smallest MSE's under different values of $n$ when $k = 2$.

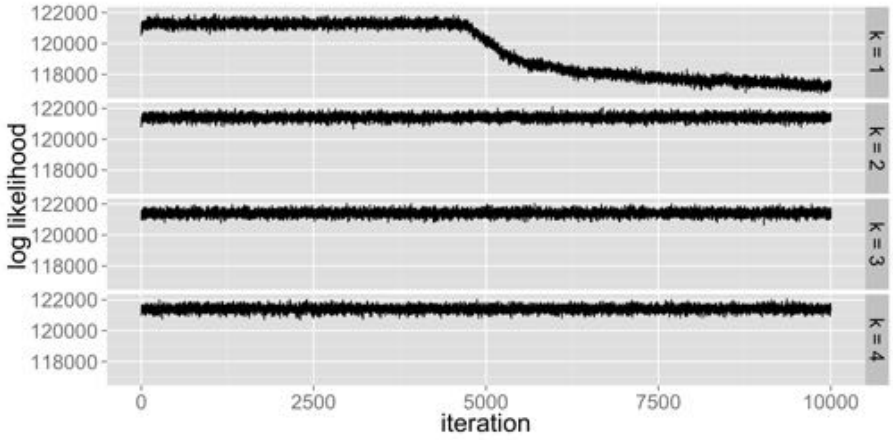


FIGURE 4.7: The trajectories of the log likelihood functions $L^{(2)}(\mathbf{\Phi})$ for mixed data type simulation with $n = 1,000$ under different values of $k$.

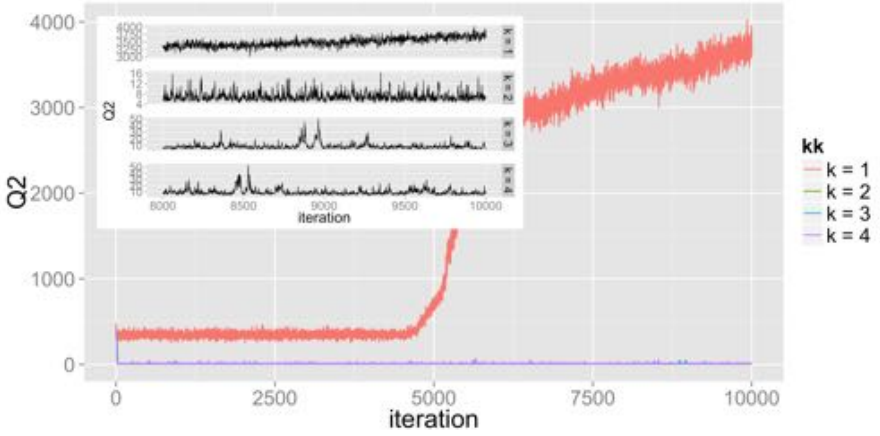


FIGURE 4.8: The trajectories of $Q_n^{(2)}(\mathbf{\Phi}, \boldsymbol{I})$ defined in Chapter 3 equation (3.14) for the mixed data type simulation with $n = 1,000$ under different values of $k$. The last $2,000$ iterations are plotted in the zoomed panel.

Table 4.4: Mean squared error (MSE) of parameter estimation in simulation with categorical, Gaussian, Poisson mixed variables using Bayesian GMM and GMM in MELD. For GMM estimation its standard deviations of MSE's are calculated from parameter estimates in ten data sets, and are provided in parentheses of MSE column. For the Bayesian GMM method the standard deviations of MSE's are calculated from posterior mean estimates of the ten data sets. For non-categorical data squared Euclidean distance is used to recover membership variable.

| | | Categorical | | | Gaussian | | | Poisson | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | | GMM $Q^{(2)}(\mathbf{\Phi})$ | | |
| $n$ | $k$ | 1st stage | 2nd stage | BGMM $L^{(2)}(\mathbf{\Phi})$ | 1st stage | 2nd stage | BGMM $L^{(2)}(\mathbf{\Phi})$ | 1st stage | 2nd stage | BGMM $L^{(2)}(\mathbf{\Phi})$ |
| 50 | 1 | **0.010(1.3e-3)** | 0.010(2.3e-3) | 0.013(0.4e-3) | 9.403(0.448) | 9.419(2.270) | **8.742(0.017)** | **6.799(0.440)** | 7.125(2.144) | 7.928(0.343) |
| | 2 | 0.021(3.2e-3) | 0.021(4.7e-3) | **0.011(0.2e-3)** | 0.633(0.880) | 1.070(3.389) | **0.271(0.066)** | 4.444(0.593) | 5.165(5.323) | **3.984(0.153)** |
| | 3 | 0.036(5.4e-3) | 0.035(4.4e-3) | **0.019(0.4e-3)** | 1.354(0.838) | **1.346(0.841)** | 1.430 (0.361) | 5.674(0.946) | 5.696(0.947) | **5.509(0.600)** |
| | 4 | 0.048(5.4e-3) | 0.047(5.2e-3) | **0.026(0.6e-3)** | 2.435(1.145) | 1.998(1.164) | **1.135 (0.581)** | **6.559(1.278)** | 6.587(1.304) | 7.916(0.892) |
| 100 | 1 | **0.006(0.5e-3)** | 0.007(5.5e-3) | 0.013(1.1e-3) | 9.815(0.438) | 9.503(0.337) | **8.998(0.015)** | **6.718(0.281)** | 7.677(7.146) | 10.445(0.582) |
| | 2 | 0.011(1.2e-3) | 0.011(1.2e-3) | **0.006(0.1e-3)** | 0.880(0.288) | 0.345(0.283) | **0.114(0.091)** | 4.780(0.501) | 4.772(0.503) | **4.111(0.172)** |
| | 3 | 0.020(2.1e-3) | 0.020(2.0e-3) | **0.010(0.5e-3)** | **0.838(0.651)** | 1.494(0.668) | 2.065(0.430) | **5.658(0.785)** | 5.680(0.770) | 5.710(0.561) |
| | 4 | 0.029(1.9e-3) | 0.028(1.8e-3) | **0.015(0.6e-3)** | **1.145(0.668)** | 2.462(0.675) | 1.967(1.424) | 7.480(0.951) | 7.493(0.934) | **7.386(0.909)** |
| 200 | 1 | **0.004(0.2e-3)** | 0.005(0.1e-3) | 0.012(1.1e-3) | 9.992(0.357) | 9.647(0.430) | **9.008(0.009)** | 6.522(0.154) | **6.451(0.126)** | 10.095(0.580) |
| | 2 | 0.006(0.4e-3) | 0.006(0.1e-3) | **0.004(0.1e-3)** | 0.121(0.140) | 0.117(0.135) | **0.111(0.062)** | 4.606(0.266) | 4.607(0.267) | **4.305(0.227)** |
| | 3 | 0.011(1.2e-3) | 0.010(0.6e-3) | **0.006(0.5e-3)** | **1.566(0.490)** | 1.586(0.496) | 1.695(0.417) | 6.203(0.597) | **6.200(0.597)** | 6.496(0.910) |
| | 4 | 0.018(1.7e-3) | 0.018(1.1e-3) | **0.009(0.7e-3)** | 3.243(0.660) | 3.257(0.665) | **0.891(0.600)** | **7.121(0.763)** | 7.123(0.769) | 8.386(1.406) |
| 500 | 1 | **0.003(0.1e-3)** | 0.003(0.1e-3) | 0.009(2.1e-3) | 9.726(0.240) | 9.447(0.185) | **9.033(0.016)** | **6.442(0.110)** | 6.454(0.493) | 9.349(1.429) |
| | 2 | 0.003(0.1e-3) | 0.003(0.8e-3) | **0.002(<0.1e-3)** | 0.083(0.046) | **0.083(0.045)** | 0.160(0.056) | 4.276(0.162) | 4.273(0.163) | **4.228(0.211)** |
| | 3 | 0.006(0.6e-3) | 0.006(1.3e-3) | **0.004(0.2e-3)** | **1.521(0.355)** | 1.525(0.354) | 2.222(0.532) | **5.288(0.275)** | 5.289(0.278) | 5.368(0.843) |
| | 4 | 0.011(1.2e-3) | 0.010(0.8e-3) | **0.005(0.4e-3)** | 2.855(0.243) | 2.870(0.245) | **1.974(1.337)** | 5.083(0.426) | **5.082(0.430)** | 5.984(0.935) |
| 1,000 | 1 | **0.002(0.1e-3)** | 0.003(2.0e-3) | 0.010(1.7e-3) | 9.696(0.146) | 9.413(0.106) | **9.002(0.008)** | 6.376(0.060) | **6.324(0.044)** | 9.951(1.234) |
| | 2 | 0.002(0.1e-3) | 0.002(0.1e-3) | **0.002(<0.1e-3)** | 0.170(0.040) | **0.167(0.036)** | 0.191(0.064) | 4.697(0.113) | 4.698(0.113 | **4.511(0.243)** |
| | 3 | 0.005(0.5e-3) | 0.005(0.5e-3) | **0.003(0.2e-3)** | 1.969(0.191) | **1.967(0.193)** | 2.298(0.763) | 5.865(0.239) | 5.871(0.238) | **5.667(0.713)** |
| | 4 | 0.007(0.9e-3) | 0.007(0.8e-3) | **0.003(0.3e-3)** | 3.520(0.177) | 3.533(0.177) | **2.654(1.789)** | **5.164(0.247)** | 5.182(0.247) | 6.316(1.039) |

We further plot the log likelihood of $L^{(2)}(\mathbf{\Phi})$ (Figure 4.7), the objective function $Q^{(2)}(\mathbf{\Phi})$ (Figure 4.8) and trajectories of mean parameters for categorical, Gaussian and Poisson variables (Figure 4.9, 4.10 and 4.11). With $k = 1$, the Markov chain mixes poorly. The objective function also diverges after a number of iterations. With other values of $k$, the Markov chains mix well. The values of objective function decrease greatly after a few iterations and then they stay stable. The smallest values of the objective function are achieved under the correct value of $k = 2$. The trajectories of the three types of variables also indicate good mixing when $k > 1$. With $k = 2$, the values of mean parameter converge to the true values.

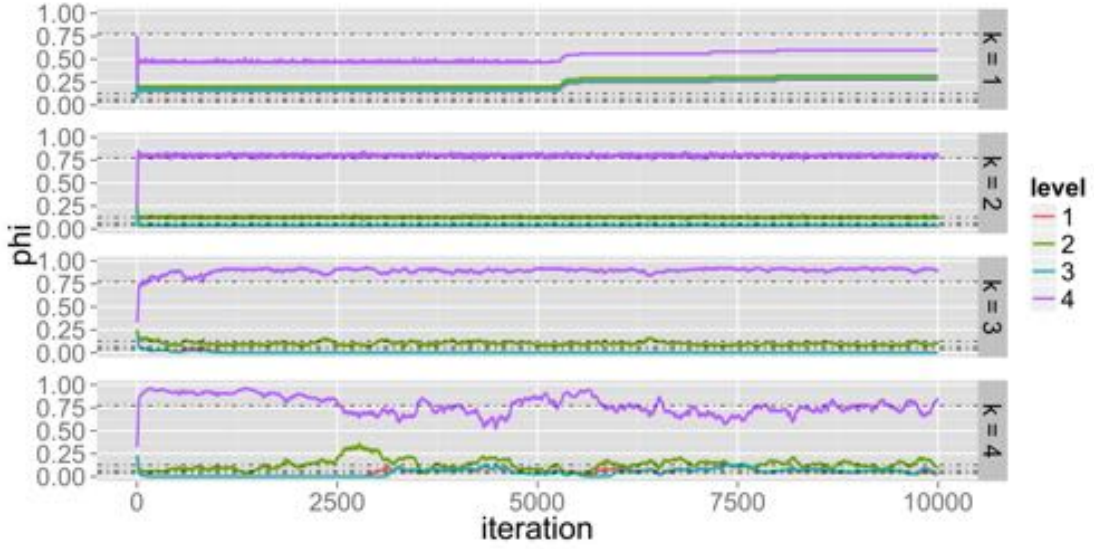


FIGURE 4.9: The trajectories of posterior draws of one component parameter $\boldsymbol{\phi}_{jh}$ for a categorical variable under different values of $k$ are plotted. Results shown are using likelihood $L^{(2)}(\mathbf{\Phi})$ with $n = 1,000$. Different coordinates in $\boldsymbol{\phi}_{jh}$ are shown by different colors. True values are plotted as dotted lines.

### *4.4.2 Joint orthogonal diagonalization*

To test the performance of our MCMC sampler on other manifolds, we diverge from MELD framework for a while and consider a problem of joint diagonalization of multiple matrices (Zhong and Girolami, 2012). This problem requires drawing samples

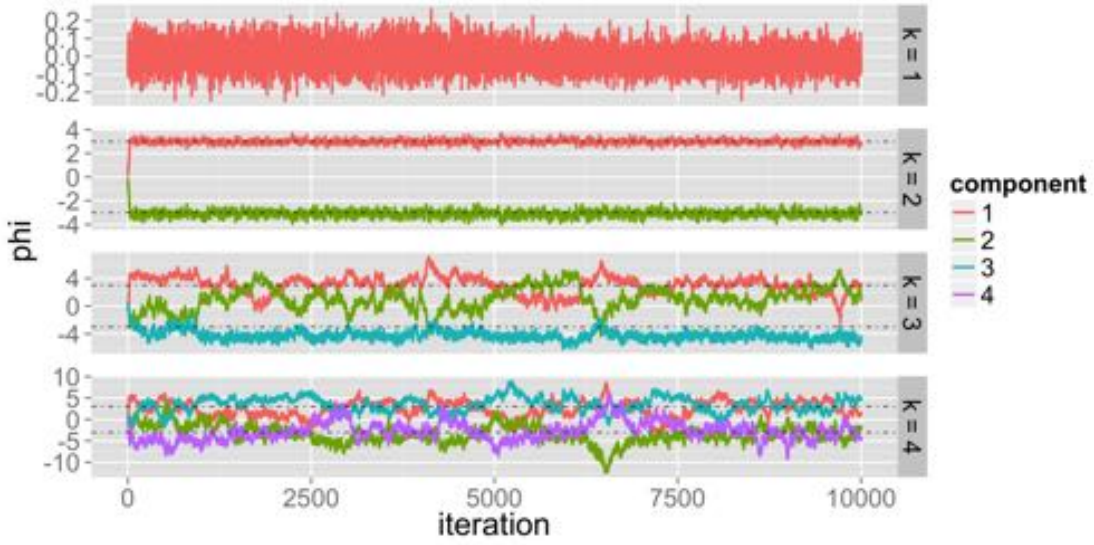


FIGURE 4.10: The trajectories of posterior draws of mean parameter $\phi_{jh}$ under different values of $k$ for a Gaussian variable are plotted. Results shown are using likelihood $L^{(2)}(\mathbf{\Phi})$ with $n = 1,000$. Different components of the mean parameter are shown by different colors. True values are plotted as dotted lines.

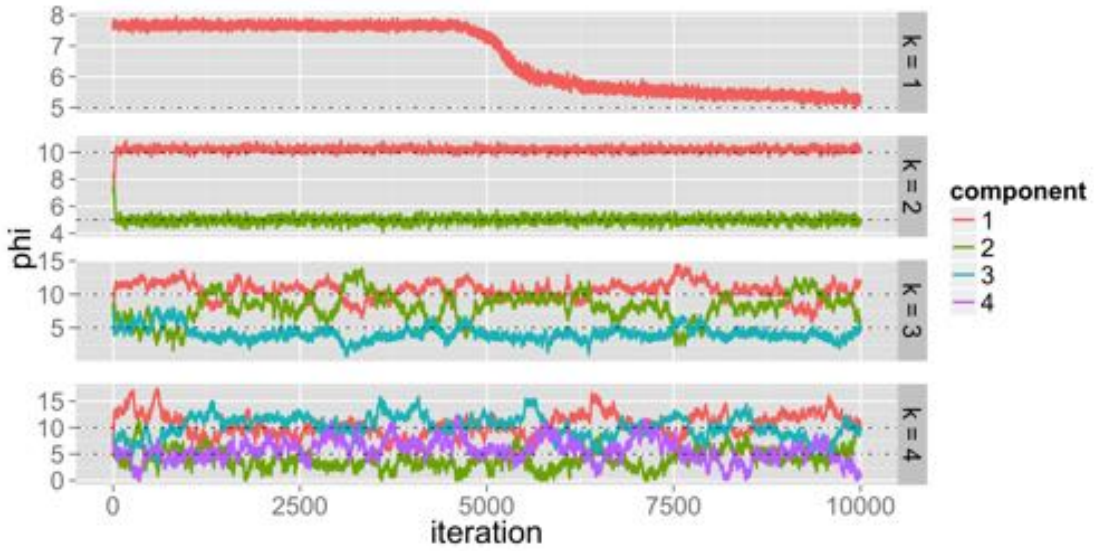


FIGURE 4.11: The trajectories of posterior draws of mean parameter $\phi_{jh}$ under different values of $k$ for a Poisson variable are plotted. Results shown are using likelihood $L^{(2)}(\mathbf{\Phi})$ with $n = 1,000$. Different components of the mean parameter are shown by different colors. True values are plotted as dotted lines.

from the Stiefel manifold $\mathcal{V}(p,k)$. Let $\boldsymbol{M}^{(1)},\cdots,\boldsymbol{M}^{(m)} \in \mathbb{R}^{p\times p}$ be $m$ matrices that we want to perform joint diagonalization. We write following probabilistic model

$$\boldsymbol{M}^{(v)} = \boldsymbol{U}\boldsymbol{\Lambda}^{(v)}\boldsymbol{U}^{\top} + \boldsymbol{E}^{(v)}, \text{ for } v = 1,\ldots,m, \tag{4.25}$$

where $\boldsymbol{U} \in \mathcal{V}(p,k)$, $\boldsymbol{\Lambda}^{(v)} = \text{diag}(\lambda_1^{(v)},\cdots,\lambda_k^{(v)})$ is a diagonal matrix and the entries of $\boldsymbol{E}^{(v)}$ follows independent Gaussian distribution $\text{N}(0,\sigma_v^2)$. This model assumes the $m$ matrices are noisy version of underlying $m$ symmetric matrices with the same set of eigenvectors. The likelihood can be shown to proportional to following term

$$\prod_{v=1}^{m}(\sigma_v^2)^{-p^2/2}\exp\bigg(-\frac{1}{2\sigma_v^2}||\boldsymbol{M}^{(v)} - \boldsymbol{U}\boldsymbol{\Lambda}^{(v)}\boldsymbol{U}^{\top}||_F^2\bigg).$$

This problem has many applications. When those matrices are symmetric sample covariance matrices under different conditions, this problem is related to the common principal components analysis studied by Flury (1984). The moment tensor approach introduced in Chapter 3 can also be treated as a joint diagonalization problem when the third order moment tensor is projected to matrices by different random projection vectors (Anandkumar et al., 2012b).

We define following prior distributions for the likelihood model 4.25. For $\lambda_h^{(v)}$ we assign $\lambda_h^{(v)} \sim \text{N}(0,\sigma_v^2\tau_h^2)$. The parameter $\tau_h^2$ can be viewed as the signal-to-noise ratio of the $h$th orthogonal latent factor. For $\boldsymbol{U}$ we assign an improper uniform prior $p(\boldsymbol{U}) \propto 1$. We further assign hyper-prior distributions for $\sigma_v^2$ and $\tau_h^2$ as

$$\sigma_v^{-2} \sim \text{Ga}(a_\sigma, b_\sigma), \qquad \tau_h^{-2} \sim \text{Ga}(a_\tau, b_\tau).$$

With this setup, the posterior distributions for our model parameters can be written as

- Posterior of $\lambda_h^{(v)}$

$$p(\lambda_h^{(v)}|-) \sim \text{N}\bigg(\frac{\tau_h^2(\boldsymbol{u}_h^{\top}\overline{\boldsymbol{M}}^{(v)}\boldsymbol{u}_h)}{\tau_h^2+1}, \frac{\sigma_v^2\tau_h^2}{\tau_h^2+1}\bigg),$$

  where $\overline{\boldsymbol{M}}^{(v)} = \left(\boldsymbol{M}^{(v)} + (\boldsymbol{M}^{(v)})^{\top}\right)/2$ and $\boldsymbol{u}_h$ is the $h$th column of $\boldsymbol{U}$.

- Posterior of $\boldsymbol{U}$

$$p(\boldsymbol{U}|-) \propto \prod_{v=1}^{m} \exp\left(\frac{1}{\sigma_v^2}\mathrm{tr}(\overline{\boldsymbol{M}}^{(v)}\boldsymbol{U}\boldsymbol{\Lambda}^{(v)}\boldsymbol{U}^\top)\right)$$

$$= \exp\left(\sum_{v=1}^{m}\sum_{h=1}^{k} \boldsymbol{u}_h^\top \frac{\overline{\boldsymbol{M}}^{(v)}\lambda_h^{(v)}}{\sigma_v^2}\boldsymbol{u}_h\right)$$

$$= \exp\left[\sum_{h=1}^{k} \boldsymbol{u}_h^\top\left(\sum_{v=1}^{m} \frac{\overline{\boldsymbol{M}}^{(v)}\lambda_h^{(v)}}{\sigma_v^2}\right)\boldsymbol{u}_h\right].$$

  Note that his posterior distribution is not a standard BMF distribution in (4.23) because each $\boldsymbol{u}_h$ has its own coefficient matrix in the quadratic form.

- Posterior of $\sigma_v^{-2}$

$$p(\sigma_v^{-2}|-) \sim \mathrm{Ga}\left(a_\sigma + \frac{k}{2} + \frac{p^2}{2}, b_\sigma + \frac{1}{2}||\boldsymbol{M}^{(v)} - \boldsymbol{U}\boldsymbol{\Lambda}^{(v)}\boldsymbol{U}^\top||_F^2 + \frac{1}{2}\sum_{h=1}^{k}\frac{(\lambda_h^{(v)})^2}{\tau_h^2}\right).$$

- Posterior of $\tau_h^{-2}$

$$p(\tau_h^{-2}|-) \sim \mathrm{Ga}\left(a_\tau + \frac{m}{2}, b_\tau + \frac{1}{2}\sum_{v=1}^{m}\frac{(\lambda_h^{(v)})^2}{\sigma_v^2}\right).$$

We use the geodesic Riemannian manifold Hamiltonian Monte Carlo sampler in Algorithm 4.2 to draw posterior samples of $\boldsymbol{U}$ from $p(\boldsymbol{U}|-)$. We have shown the geodesic flow on $\mathcal{V}(p,k)$. To construct the sampler we only need to derive the gradient of its log density. Let $\boldsymbol{A}_h = \sum_{v=1}^{m}\overline{\boldsymbol{M}}^{(v)}\lambda_h^{(v)}/\sigma_v^2$. Then the gradient of the log posterior has the form

$$\frac{\partial \log[p(\boldsymbol{U}|-)]}{\partial \boldsymbol{U}} = 2\left(\boldsymbol{A}_1\boldsymbol{u}_1, \cdots, \boldsymbol{A}_k\boldsymbol{u}_k\right).$$

Our method distinguishes from previous joint diagonalization algorithms in that it allows us to sample $\boldsymbol{U}$ jointly without using conditional Gibbs sampler for $\boldsymbol{u}_h$ (Zhong

and Girolami, 2012), resulting an efficient sampler with fast mixing behavior as shown in this simulation.

To evaluate the performance of our method, we generate $m = 10$ matrices with axis dimension $p = 10$. We set $k = 3$ and generate $\boldsymbol{U} \in \mathcal{V}(p, k)$ as follows. We first draw $p$ samples from a standard $p$ dimensional Gaussian distribution. Then we calculate their sample covariance matrix. The first $k$ eigenvectors of the sample covariance matrix are used to construct $\boldsymbol{U}$. The value of $\lambda_h^{(v)}$ is draw from $\mathrm{N}(0, 5^2)$ and $\sigma_v^2$ is generated from the uniform distribution on $(0, 1)$. Errors are further generated from $\mathrm{N}(0, \sigma_v^2)$ for each matrix to form the observed $\boldsymbol{M}^{(v)}$. The parameters for the gamma distributions of $\sigma_v^{-2}$ and $\tau_h^{-2}$ are set to $a_\sigma = a_\tau = 1$, $b_\sigma = b_\tau = 0.3$. We apply our method on the simulated data. For the geodesic sampler, the number of numerical integration steps is set to 10 and the step size is set to $\epsilon = 0.01$. Figure 4.12 shows the results. The trajectory of the log likelihood function demonstrates the overall good mixing behavior of our sampler. The Markov chain converges to the high likelihood region after few iterations (Figure 4.12A). The posterior draws of the first coordinate of $\boldsymbol{U}$ show that the geodesic HMC method is efficient in sampling from the Stiefel manifold $\mathcal{V}(10, 3)$. The values of the posterior draws cover the true values of $\boldsymbol{U}$ well (Figure 4.12B). The trajectories of $\boldsymbol{\Lambda}^{(1)}$ also show good mixing (Figure 4.12C). The values of $\lambda_1^{(1)}$ and $\lambda_3^{(1)}$ are shrunk to zero compared with their true values. This can be seen from the posteriors of $\tau_1^2$ and $\tau_3^2$ in Figure 4.12D: Both of $\tau_1^2$ and $\tau_3^2$ are concentrated on small values, making the variances of $\lambda_1^{(\cdot)}$ and $\lambda_3^{(\cdot)}$ small. In comparison, the value of $\lambda_2^{(1)}$ is not shrunk and the posterior of $\tau_2^2$ has support on large values.

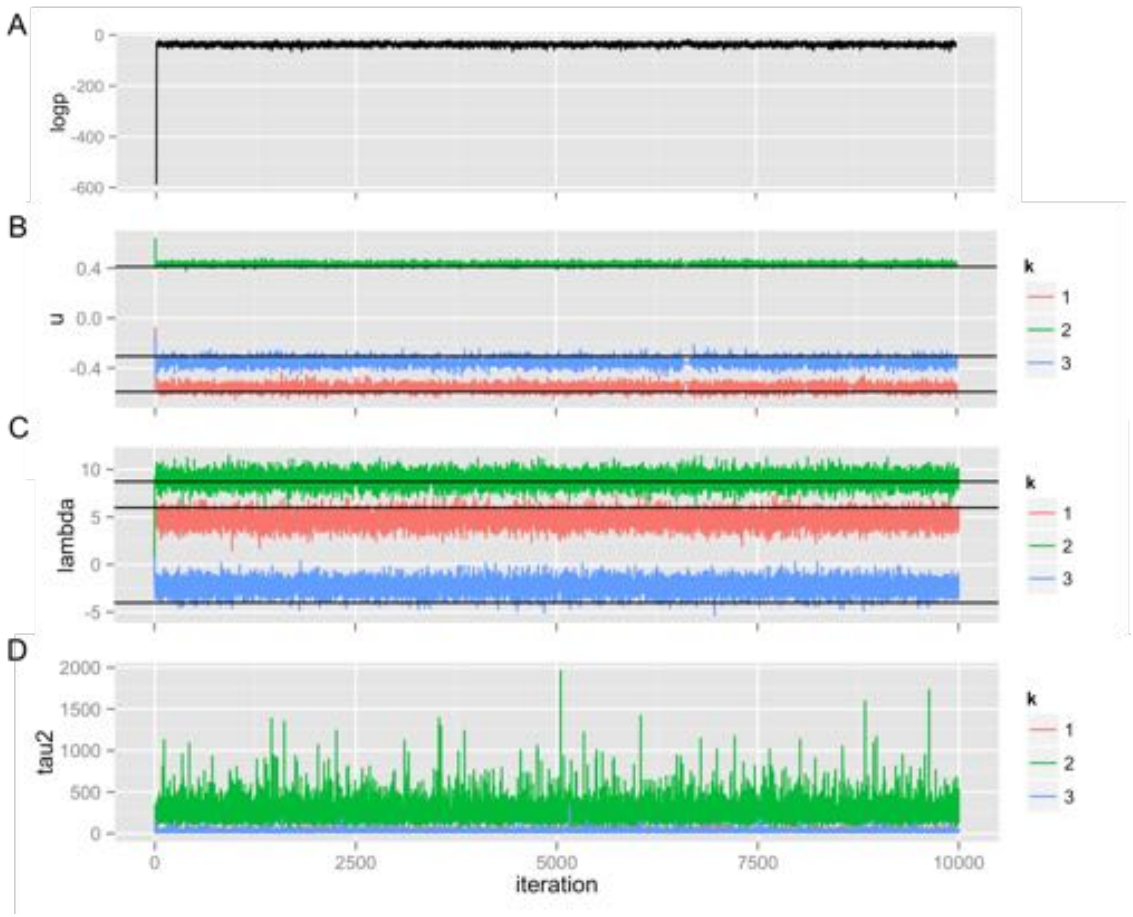


FIGURE 4.12: Results of running geodesic HMC method for joint orthogonal diagonalization simulation. The number of latent orthogonal components $k$ is set to 3. Panel A: The trajectory of log likelihood. Panel B: Posterior draws of the first coordinate of $\boldsymbol{U}$ in the three components. Panel C: Posterior draws of the three diagonal entries in $\boldsymbol{\Lambda}^{(1)}$. Panel D: Posterior draws of hyperparameter $\tau^2$ in the three components.

## 4.5 Discussion and conclusion

In this chapter we have proposed an efficient Monte Carlo method for distributions defined on manifolds. This is motivated by the aim of developing a Bayesian GMM approach for the generalized Dirichlet latent variable model proposed in Chapter 3. Using Bayesian GMM for MELD avoids specification of weight matrices in the objectives. Instead the weights are treated as unknown variables and their posteriors

are computed using MCMC algorithms. In addition, Bayesian model selection methods can be used to assess the model fitness with different values of $k$. For example we might be able to use nonparametric priors such as Dirichlet process prior in the pseudo-likelihood framework.

The Monte Carlo method proposed in this chapter combines the Hamiltonian Monte Carlo algorithm with a geodesic integrator. With the new method we are able to sample from distributions defined on different manifolds, for example the probability simplex and the Stiefel manifold consisting of orthonormal $p \times k$ matrices. We use the method in the posterior computations of the Bayesian GMM for MELD. Results show the superior performance of our Bayesian GMM compared with original GMM in simulations with categorical data and mixed data types. Then we use the algorithm in another problem where multiple matrices are jointly diagonalized. The problem can be further extended to common principal components analysis and moment tensor decompositions.

# 5

# Concluding remarks

## 5.1 Summary

This dissertation addresses three important problems in modeling large scale multivariate data. The first problem is how to combinatorially model covariance structures among multiple couple observations. The motivation for this research is that many modern data sets are collected in a coupled manor. For example expression levels for multiple genes could be measured under different conditions. Data of such kind could be represented by multiple coupled matrices, with each matrix is also known as a view. Researchers are particular interested in modeling covariance specific to each view and covariance among combinations of views. We address this problem by developing a Bayesian group factor analysis model. The model assigns a shrinkage prior organized into three levels on the loading matrix. The prior induces both element-wise sparsity (variable selection) and column-wise sparsity (view selection). The combination of both variable selection and view selection is a key innovation of our new model. Variable selection generates interpretable factors and view selection allows us to combinatorially model covariance structure among mul-

tiple data sets. Using Bayesian shrinkage prior to achieve structured sparsity has many advantages over the frequentist approach of generating sparse solutions using regularization norms. First the uncertainty of parameter estimates could be assessed by posterior computation using MCMC algorithms. Second, Bayesian approach allows information to be borrowed through hierarchical parameterizations, generating adapted shrinkage. Third, Bayesian approach using continuous priors generates a continuous shrinkage across the real line. This continuous solution allows the priors to have more customizable behaviors both around zero and at tails far away from zero. Those advantages have been demonstrated by comparing our new model with other models in simulation studies. We use our new model to real world applications including multivariate response prediction, condition specific gene co-expression network construction and document data analysis.

The second problem this dissertation aims to address is how to develop efficient statistical models and fast parameter estimation methods to model mixed data types. This is motivated by many real world applications. For example in genetics researchers are particularly interested in analyzing the association between genotypes and heterogeneous traits of varying data types. We develop a new mixed membership model named generalized latent Dirichlet model for this task. The model assumes each variable of a observed multivariate vector follows a mixture of $k$ components with distribution particularly specified for that variable. The mixture weights are shared by all variables in the observed multivariate vector. The new model reduces to several well known models as special cases when the distributions of variables are specified. For this new model we develop a generalized method of moment (GMM) approach for parameter estimation. Our GMM approach does not require the instantiation of latent variables, therefore it avoids the needs to alternatively update latent variables and population parameters, which could not be avoided using other estimation methods such as EM and MCMC. In addition our GMM approach only requires

the correct specification of first moments of component distributions. Our approach is orders of magnitude faster than alternative methods, and at the same time it achieves higher estimation accuracy in the existence of outliers. We demonstrate our new approach in several real world applications including promoter sequence analysis, political-economic risk analysis and eQTL study.

The last problem this dissertation tries to solve is how to draw samples from distributions defined on different manifolds. Many high dimensional statistical applications require drawing samples from a manifold. Such applications include probabilistic singular value decomposition and orthogonal factor analysis. In addition, drawing samples from a sphere is required when we embed the GMM approach developed in Chapter 3 in a Bayesian framework. To this end we develop a Monte Carlo method that combines Hamiltonian Monte Carlo (HMC) algorithm with a geodesic integrator. The HMC component allows distant moves in the parameter space to be accepted and the geodesic integrator component restricts the moves to the parameter manifold. We apply the method in two cases: a matrix joint orthogonal diagonalization problem and the posterior computation of the Bayesian GMM method for Chapter 3.

## 5.2 Future directions

There are many potential extensions of previous chapters. Some of the extensions are described below.

### *5.2.1 Chapter 2*

The group factor model BASS can be viewed as achieving structured variable selection at both variable level and view level. This concept is related to the structured variable selection which has been investigated recently either using penalization with structured norms (Jenatton et al., 2011) or developing Bayesian approaches using

graphical priors (Li and Zhang, 2010). This new research direction is motivated by the fact that high dimensional observations are often represented in a structured way. For example in a genome-wide association study, nearby genetic variants are often highly correlated due to linkage disequilibrium. In a functional MRI study, observed images are spatially correlated due to the structures of the brain. In a gene expression analysis, products of genes can be annotated to form a structured ontology that is represented by a directed graph. Given these highly structured data, it is interesting to ask whether variable selection could leverage those information to generate better results. An interesting extension of BASS could be to organize the $p$ variables into a tree and to perform variable selections at different levels/branches of the tree. For example, when we analyze gene expression data, genes could be organized into a tree structure according to the functional annotations of their products. We could develop new efficient structured variable selection methods that could scale to massive data sets current available.

### *5.2.2 Chapter 3*

The original coordinate descent algorithm developed in MELD scales with $O(p^2)$. This complexity hinders its application when $p$ is large. A direct extension of the algorithm is to use stochastic gradient methods by calculating an approximate gradient in each step when we perform parameter updates.

Another extension of Chapter 3 is that the GMM approach developed in MELD could be extended to allow variables taking network type data. Community or modular structure detection in network data has become increasingly important in recent years. Such network modules correspond to functional units in a network (Newman, 2012). For example in a protein-protein interaction network, highly connected proteins in a module indicate they might have similar functions. Most recently the mixed membership stochastic block (MMSB) model is proposed (Airoldi et al., 2008). This

new model extends previous stochastic block model which assumes each node in a network to belong to a single latent module. Instead, MMSB allows each node to partially belong to different modules, reflecting the fact that in real world applications a node often has multiple roles. For example a protein might exert distinct functions by forming different protein complexes with different partners. Based on MMSB model, a fast parameter estimation method using moment tensor decomposition has been proposed for MMSB model (Anandkumar et al., 2014a). It might be possible to extend our GMM framework to include network data types by adapting the network moment tensor approach developed by Anandkumar et al. (2014a). An direct implication of this extension is that it allows us to combine network data with additional node information to better detect modular structure among nodes. Such node information could be a feature vector for every node. For example, the expression values of proteins could be measured under different conditions in addition to their interaction network.

### *5.2.3 Chapter 4*

First, the Bayesian GMM framework could be extended by assigning nonparametric priors to the component mean parameters. Assigning a nonparametric prior such as the Dirichlet process prior to the component mean parameters could allow the component number to be estimated from data. Second, the joint orthogonal diagonalization problem could be extended to the multi-view problem where each view represents measurements of a same set of variables under different conditions, which is related to the pooled covariance estimation problem studied by Hoff (2009a). Moreover the joint orthogonal diagonalization problem could be used to perform third order moment tensor decomposition: by projecting the third order moment tensor to matrices using different random projection vectors, the decomposition problem can be written as a joint orthogonal diagonalization problem (Anandkumar et al., 2012b).

# Appendix A

## Appendix for a scalable Bayesian group factor analysis model

### A.1 Markov chain Monte Carlo (MCMC) algorithm for posterior inference of BASS

We derive an MCMC algorithm with Gibbs sampling steps for our Bayesian group factor analysis model in Chapter 2. Write the joint distribution of the full model as

$$\begin{aligned} & p(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{Z}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ & \quad = p(\boldsymbol{Y}|\boldsymbol{\Lambda}, \boldsymbol{X}, \boldsymbol{\Sigma}) p(\boldsymbol{X}) \\ & \qquad \times p(\boldsymbol{\Lambda}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{\Delta}, \boldsymbol{Z}, \boldsymbol{\Phi}) p(\boldsymbol{\Delta}|\boldsymbol{\Phi}) p(\boldsymbol{\Phi}|\boldsymbol{T}) p(\boldsymbol{T}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\boldsymbol{\gamma}) \\ & \qquad \times p(\boldsymbol{\Sigma}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}), \end{aligned}$$

where $\boldsymbol{\Theta} = \{\theta_{jh}^{(v)}\}$, $\boldsymbol{\Delta} = \{\delta_{jh}^{(v)}\}$, $\boldsymbol{\Phi} = \{\phi_h^{(v)}\}$, $\boldsymbol{T} = \{\tau_h^{(v)}\}$, $\boldsymbol{\eta} = \{\eta^{(v)}\}$ and $\boldsymbol{\gamma} = \{\gamma^{(v)}\}$ are the collections of the prior parameters in equation (2.14)

Update latent factors

$$\boldsymbol{x}_i|- \sim \mathrm{N}_k\Big((\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda} + \boldsymbol{I})^{-1}\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}_i, (\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda} + \boldsymbol{I})^{-1}\Big),$$

Update the $j$th row of the loading

$$\boldsymbol{\lambda}_{j\cdot}^T|- \sim \mathrm{N}_k\bigg((\sigma_j^{-2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{D}_j^{-1})^{-1}\sigma_j^{-2}\boldsymbol{X}\boldsymbol{y}_{j\cdot}^T, (\sigma_j^{-2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{D}_j^{-1})^{-1}\bigg),$$

where

$$\boldsymbol{D}_j^{-1} = \mathrm{diag}\bigg((\theta_{j1}^{(v_j)})^{I(z_1^{(v_j)}=1)}(\phi_1^{(v_j)})^{I(z_1^{(v_j)}=0)}, \cdots, (\theta_{jk}^{(v_j)})^{I(z_k^{(v_j)}=1)}(\phi_k^{(v_j)})^{I(z_k^{(v_j)}=0)}\bigg),$$

and $v_j$ represents the view $j$th row belongs to.

Update $\theta_{jh}^{(v)}$, $\delta_{jh}^{(v)}$ and $\phi_h^{(v)}$ with $z_h^{(v)} = 1$

$$\theta_{jh}^{(v)}|- \sim \mathrm{GIG}(0, 2\delta_{jh}^{(v)}, (\lambda_{jh}^{(v)})^2),$$

$$\delta_{jh}^{(v)}|- \sim \mathrm{Ga}(1, \phi_h^{(v)} + \theta_{jh}^{(v)}),$$

$$\phi_h^{(v)}|- \sim \mathrm{Ga}(1/2p_v + 1/2, \sum_{j=1}^{p_v}\delta_{jh}^{(v)} + \tau_h^{(v)}),$$

where GIG is the generalized inverse Gaussian distribution.

Update $\phi_h^{(v)}$ with $z_h^{(v)} = 0$

$$\phi_h^{(v)}|- \sim \mathrm{GIG}(1/2 - p_w/2, 2\tau_h^{(v)}, \sum_{j=1}^{p_v}(\lambda_{jh}^{(v)})^2).$$

Update the rest parameters in the loading prior

$$\tau_h^{(v)}|- \sim \mathrm{Ga}(1, \phi_h^{(v)} + \eta^{(v)}),$$

$$\eta^{(v)}|- \sim \mathrm{Ga}(1/2k + 1/2, \gamma^{(v)} + \sum_{h=1}^{k}\tau_h^{(v)}),$$

$$\gamma^{(v)}|- \sim \mathrm{Ga}(1, \eta^{(v)} + 1),$$

$$\pi^{(v)}|- \sim \mathrm{Be}(1 + \sum_{h=1}^{k}z_h^{(v)}, 1 + k - \sum_{h=1}^{k}z_h^{(v)}).$$

The full conditional of $z_h^{(v)}$ is

$$\Pr(z_h^{(v)} = 1|-) \propto \pi^{(v)}\prod_{j=1}^{p_v}\mathrm{N}(\lambda_{jh}^{(v)}; 0, \theta_{jh}^{(v)})\mathrm{Ga}(\theta_{jh}^{(v)}; a, \delta_{jh}^{(v)})\mathrm{Ga}(\delta_{jh}^{(v)}; b, \phi_h^{(v)}),$$

$$\Pr(z_h^{(v)}=0|-)\propto(1-\pi^{(v)})\prod_{j=1}^{p_v}\mathrm{N}(\lambda_{jh}^{(v)};0,\phi_h^{(v)}).$$

We further integrate out $\delta_{jh}^{(v)}$ in $\Pr(z_h^{(v)}=1|-)$

$$\begin{aligned}\Pr(z_h^{(v)}=1|-)&\propto\pi^{(v)}\prod_{j=1}^{p_v}\int\mathrm{N}(\lambda_{jh}^{(v)};0,\theta_{jh}^{(v)})\mathrm{Ga}(\theta_{jh}^{(v)};a,\delta_{jh}^{(v)})\mathrm{Ga}(\delta_{jh}^{(v)};b,\phi_h^{(v)})\mathrm{d}\delta_{jh}^{(v)}\\&=\pi^{(v)}\prod_{j=1}^{p_v}\mathrm{N}(\lambda_{jh}^{(v)};0,\theta_{jh}^{(v)})\frac{\Gamma(1)}{\Gamma(1/2)\Gamma(1/2)}\frac{(\theta_{jh}^{(v)})^{-1/2}(\theta_h^{(v)})^{1/2}}{(\theta_{jh}^{(v)}+\phi_h^{(v)})}.\end{aligned}$$

Update $\sigma_j^{-2}$

$$\sigma_j^{-2}|-\sim\mathrm{Ga}\bigg(n/2+a_\sigma,1/2(\boldsymbol{y}_{j\cdot}-\boldsymbol{\lambda}_{j\cdot}\boldsymbol{X})(\boldsymbol{y}_{j\cdot}-\boldsymbol{\lambda}_{j\cdot}\boldsymbol{X})^T+b_\sigma\bigg).$$

## A.2 EM updates of loading prior parameters in BASS

We list the parameter updates for loading prior parameters in developed in Chapter 2 equation (2.14) below

$$\hat{\theta}_{jh}^{(v)}=\frac{2a-3+\sqrt{(2a-3)^2+8(\lambda_{jh}^{(v)})^2\delta_{jh}^{(v)}}}{4\delta_{jh}^{(v)}},$$

$$\hat{\delta}_{jh}^{(v)}=\frac{a+b}{\theta_{jh}^{(v)}+\phi_h^{(v)}},$$

$$\hat{\phi}_h^{(v)}=\frac{p'-1+\sqrt{(p'-1)^2+a'b'}}{a'},\text{with}$$

$$p'=\rho_h^{(v)}p_vb-(1-\rho_h^{(v)})p_v/2+c,$$

$$a'=2(\rho_h^{(v)}\sum_{j=1}^{p_v}\delta_{jh}^{(v)}+\tau_h^{(v)}),b'=(1-\rho_h^{(v)})\sum_{j=1}^{p_v}(\lambda_{jh}^{(v)})^2$$

$$\hat{\tau}_h^{(v)}=\frac{c+d}{\phi_h^{(v)}+\eta^{(v)}},$$

$$\hat{\eta}^{(v)}=\frac{dk+e}{\gamma^{(v)}+\sum_{h=1}^k\tau_h^{(v)}},$$

$$\hat{\gamma}^{(v)} = \frac{e+f}{\eta^{(v)}+\nu},$$

$$\hat{\pi}^{(v)} = \frac{\sum_{h=1}^{k}\rho_h^{(v)}}{k},$$

$$\hat{\sigma}_j^{-2} = \frac{n/2 + a_\sigma - 1}{1/2(\boldsymbol{y}_{j\cdot} - \boldsymbol{\lambda}_{j\cdot}\langle \boldsymbol{X}\rangle)(\boldsymbol{y}_{j\cdot} - \boldsymbol{\lambda}_{j\cdot}\langle \boldsymbol{X}\rangle)^T + b_\sigma}.$$

## A.3 Parameter expanded EM (PX-EM) algorithm for MAP estimate

We introduce a positive semidefinite matrix $\boldsymbol{A}$ in our original BASS model defined in Chapter 2 to obtain a parameter expanded version

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{A}_L^{-1}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i,$$

$$\boldsymbol{x}_i \sim \mathrm{N}_k(\boldsymbol{0}, \boldsymbol{A}),$$

$$\boldsymbol{\epsilon}_i \sim \mathrm{N}_k(\boldsymbol{0}, \boldsymbol{\Sigma}).$$

Here $\boldsymbol{A}_L$ is the lower triangular part of Cholesky decomposition of $\boldsymbol{A}$. Marginally the covariance matrix is still $\boldsymbol{\Omega} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma}$ and this additional parameter keeps the likelihood invariant. This new additional parameter reduces the coupling effects between the updates of loading matrix and latent factors (Liu et al., 1998; van Dyk and Meng, 2001) and serves to connect different posterior modes with equal likelihood curves indexed by $\boldsymbol{A}$ (Ročková and George, 2015).

Let $\boldsymbol{\Lambda}^\star = \boldsymbol{\Lambda}\boldsymbol{A}_L^{-1}$ and $\boldsymbol{\Xi}^\star = \{\boldsymbol{\Lambda}^\star, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$. Then the parameters of our expanded model are $\{\boldsymbol{\Xi}^\star \cup \boldsymbol{A}\}$. We assign our structured prior on $\boldsymbol{\Lambda}^\star$. Therefore the updates of $\boldsymbol{\Xi}^\star$ are unchanged given the estimates of first and second moments of $\boldsymbol{X}$. The estimates of $\langle \boldsymbol{X}\rangle$ and $\langle \boldsymbol{X}\boldsymbol{X}^T\rangle$ can still be caculated using corresponding equations in Appendix A.2 after mapping back te loading matrix to the original model by $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^\star\boldsymbol{A}_L$. The rest is to find the estimate of $\boldsymbol{A}$.

Write the $Q$ function in the expanded model as

$$Q(\boldsymbol{\Xi}^\star, \boldsymbol{A}|\boldsymbol{\Xi}_{(s)}) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Xi}_{(s)},\boldsymbol{Y},\boldsymbol{A}_{(s)}} \log\big(p(\boldsymbol{\Xi}^\star, \boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{Y})\big).$$

With $\boldsymbol{A}_{(s)}$ initialized to $\boldsymbol{I}_k$, the only term involving $\boldsymbol{A}$ is $p(\boldsymbol{X})$. Therefore the $\boldsymbol{A}$ that maximizes the function can be solved as

$$\boldsymbol{A}_{(s+1)} = \text{argmax}_{\boldsymbol{A}} Q(\boldsymbol{\Xi}^{\star}, \boldsymbol{A}|\boldsymbol{\Xi}_{(s)}) = \text{argmax}_{\boldsymbol{A}} \bigg( \text{const} - \frac{n}{2} \log |\boldsymbol{A}| - \frac{1}{2} \text{tr}\big(\boldsymbol{A}^{-1} \boldsymbol{S}^{XX}\big) \bigg).$$

The solution is simply $\boldsymbol{A}_{(s+1)} = \frac{1}{n} \boldsymbol{S}^{XX}$.

The EM algorithm in this expanded parameter space generates a sequence $\{\boldsymbol{\Xi}^{\star}{}_{(1)} \cup \boldsymbol{A}_{(1)}, \boldsymbol{\Xi}^{\star}{}_{(2)} \cup \boldsymbol{A}_{(2)}, \cdots\}$. This sequence corresponds to a sequence of parameter estimations in original space $\{\boldsymbol{\Xi}_{(1)}, \boldsymbol{\Xi}_{(2)}, \cdots\}$ with $\boldsymbol{\Lambda}$ in the original space equals to $\boldsymbol{\Lambda}^{\star} \boldsymbol{A}_L$ (Ročková and George, 2015).

Table A.1: First ten words in the specific factors for different newsgroups.

| alt.atheism | | comp.graphics | | comp.os.ms-windows.misc | | comp.sys.ibm.pc.hardware | | comp.sys.mac.hardware | |
|---|---|---|---|---|---|---|---|---|---|
| islam | atheism | graphics | polygon | windows | file | ide | drive | mac | powerbook |
| keith | mathew | 3d | gif | thanks | go | scsi | motherboard | apple | quadra |
| okcforum | atheists | tiff | images | of | dos | controller | thanks | quadra | iisi |
| atheism | livesey | image | format | cica | microsoft | vlb | ide | duo | centris |
| livesey | of | image | pov | dos | the | bios | isa | centris | mac |
| comp.windows.x | | misc.forsale | | rec.autos | | rec.motorcycles | | rec.sport.baseball | |
| window | mit | sale | offer | car | dealer | dod | bmw | baseball | hitter |
| motif | lcs | sale | forsale | cars | oil | bike | riding | braves | ball |
| server | motif | for | the | engine | toyota | motorcycle | bikes | runs | year |
| widget | xterm | sell | shipping | ford | eliot | ride | dod | phillies | players |
| lcs | code | condition | offer | cars | cars | bike | bike | sox | players |
| rec.sport.hockey | | sci.crypt | | sci.electronics | | sci.med | | sci.space | |
| hockey | bruins | encryption | crypto | circuit | radio | geb | msg | it | people |
| nhl | pens | clipper | nsa | voltage | copy | medical | doctor | space | orbit |
| game | detroit | chip | nsa | amp | battery | diet | disease | for | henry |
| team | season | key | pgp | electronics | tv | cancer | geb | digex | moon |
| leafs | espn | des | tapped | audio | power | photography | doctor | for | shuttle |
| soc.religion.christian | | talk.politics.guns | | talk.politics.mideast | | talk.politics.misc | | talk.religion.misc | |
| god | sin | atf | fbi | israeli | israeli | cramer | government | sandvik | morality |
| clh | bible | firearms | stratus | jews | armenians | optilink | drugs | koresh | jesus |
| church | petch | guns | batf | israel | armenian | kaldis | president | sandvik | religion |
| christian | mary | gun | stratus | arab | jake | clinton | br | bible | god |
| heaven | church | handheld | waco | armenians | jewish | cramer | tax | christian | objective |

# Appendix B

## Appendix for fast moment estimation for generalized latent Dirichlet models

### B.1 Proof of Theorem 3.1

*Proof.* We start with the case where $y_{ij}$ is a categorical data with $d_j$ different levels. The latent probability vector $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ik})^T \in \Delta^{k-1}$ defines the mixture proportion of individual $i$. We assume $\boldsymbol{x}_i \sim \text{Dir}(\alpha_1, \cdots, \alpha_k)$. Define $\alpha_0 = \sum_h \alpha_h$ and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_k)^T$.

We use the standard basis for encoding. We encode $y_{ij} = c_j$ as $\boldsymbol{b}_{ij} \in \mathbb{R}^{d_j}$ a binary (0/1) vector with the $c_j$th coordinate being 1 and all others being 0. Similarly, we encode the membership variable $m_{ij}$ as a $k$ dimensional binary vector $\boldsymbol{m}_{ij} \in \mathbb{R}^k$. Consider the first moment of $\boldsymbol{b}_{ij}$.

$$\boldsymbol{\mu}_j = \mathbb{E}(\boldsymbol{b}_{ij}) = \mathbb{E}[\mathbb{E}(\boldsymbol{b}_{ij}|\boldsymbol{m}_{ij})] = \mathbb{E}(\boldsymbol{\Phi}_j \boldsymbol{m}_{ij}) = \mathbb{E}[\mathbb{E}(\boldsymbol{\Phi}_j \boldsymbol{m}_{ij}|\boldsymbol{x}_i)] = \mathbb{E}(\boldsymbol{\Phi}_j \boldsymbol{x}_i) = \boldsymbol{\Phi}_j \frac{\boldsymbol{\alpha}}{\alpha_0},$$

where $\boldsymbol{\Phi}_j = (\boldsymbol{\phi}_{j1}, \cdots \boldsymbol{\phi}_{jk})$.

We consider second order moment conditions. There are four types of second moments: same variable same subject (type SS), same variable cross subject (type SC), cross variable same subject (type CS), and cross variable cross subject (type

CC). Of the four types, only the CS type is needed to prove the theorem. The CS type second moment for $\boldsymbol{b}_{ij}$ and $\boldsymbol{b}_{it}$ $(j \neq t)$ can be written as

$$\mathbb{E}(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it}) = \boldsymbol{\Phi}_j \mathbb{E}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{it}) \boldsymbol{\Phi}_t^T = \boldsymbol{\Phi}_j \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i) \boldsymbol{\Phi}_t^T.$$

For a Dirichlet distributed variable,

$$\begin{aligned}\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i) &= \operatorname{cov}(\boldsymbol{x}_i) + \mathbb{E}(\boldsymbol{x}_i) \circ \mathbb{E}(\boldsymbol{x}_i) \\ &= \frac{1}{\alpha_0(\alpha_0+1)} \operatorname{diag}(\boldsymbol{\alpha}) + \frac{\alpha_0}{\alpha_0^2(\alpha_0+1)} \boldsymbol{\alpha} \circ \boldsymbol{\alpha}.\end{aligned}$$

Then we have

$$\begin{aligned}\mathbb{E}(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it}) &= \boldsymbol{\Phi}_j \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i) \boldsymbol{\Phi}_t^T \\ &= \frac{1}{\alpha_0(\alpha_0+1)} \sum_{h=1}^{k} \alpha_h \boldsymbol{\phi}_{jh} \circ \boldsymbol{\phi}_{th} + \frac{\alpha_0}{\alpha_0+1} \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_t. \qquad \text{(B.1)}\end{aligned}$$

We next consider third order moment conditions. There are eight different types of third order moments for $\boldsymbol{b}_{ij}$. Only the moments with different variables for the same subject are needed to prove the theorem.

We consider the third cross moment for $\boldsymbol{b}_{ij}$, $\boldsymbol{b}_{it}$ and $\boldsymbol{b}_{is}$ with $j \neq t \neq s$ for the same subject. First we calculate $\mathbb{E}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{is} \circ \boldsymbol{m}_{it})$.

$$\begin{aligned}\mathbb{E}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{is} \circ \boldsymbol{m}_{it}) &= \mathbb{E}[\mathbb{E}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{is} \circ \boldsymbol{m}_{it} | \boldsymbol{x}_i)] \\ &= \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i) \\ &= \frac{1}{\alpha_0(\alpha_0+1)(\alpha_0+2)} \Bigg( (\boldsymbol{\alpha} \circ \boldsymbol{\alpha} \circ \boldsymbol{\alpha}) + \sum_{h=1}^{k} \alpha_h (\boldsymbol{e}_h \circ \boldsymbol{e}_h \circ \boldsymbol{\alpha}) \\ &\qquad + \sum_{h=1}^{k} \alpha_h (\boldsymbol{e}_h \circ \boldsymbol{\alpha} \circ \boldsymbol{e}_h) + \sum_{h=1}^{k} \alpha_h (\boldsymbol{\alpha} \circ \boldsymbol{e}_h \circ \boldsymbol{e}_h) \\ &\qquad + 2 \sum_{h=1}^{k} \alpha_h (\boldsymbol{e}_h \circ \boldsymbol{e}_h \circ \boldsymbol{e}_h) \Bigg).\end{aligned}$$

Here $\boldsymbol{e}_h$ is standard basis vector of length $k$ with $h$th coordinate being one. The third order moment tensor of $\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it}$ can be derived as

$$
\begin{aligned}
&\mathbb{E}(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it}) = \mathbb{E}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{is} \circ \boldsymbol{m}_{it}) \times \{\boldsymbol{\Phi}_j, \boldsymbol{\Phi}_s, \boldsymbol{\Phi}_t\} \\
&= \frac{1}{\alpha_0(\alpha_0+1)(\alpha_0+2)} \Bigg( (\boldsymbol{\Phi}_j\boldsymbol{\alpha}) \circ (\boldsymbol{\Phi}_s\boldsymbol{\alpha}) \circ (\boldsymbol{\Phi}_t\boldsymbol{\alpha}) + \sum_{h=1}^{k} \alpha_h [\boldsymbol{\phi}_{jh} \circ \boldsymbol{\phi}_{sh} \circ (\boldsymbol{\Phi}_t\boldsymbol{\alpha})] \\
&\qquad + \sum_{h=1}^{k} \alpha_h [\boldsymbol{\phi}_{jh} \circ (\boldsymbol{\Phi}_s\boldsymbol{\alpha}) \circ \boldsymbol{\phi}_{th}] + \sum_{h=1}^{k} \alpha_h [(\boldsymbol{\Phi}_j\boldsymbol{\alpha}) \circ \boldsymbol{\phi}_{sh} \circ \boldsymbol{\phi}_{th}] \\
&\qquad + 2 \sum_{h=1}^{k} \alpha_h \boldsymbol{\phi}_{jh} \circ \boldsymbol{\phi}_{sh} \circ \boldsymbol{\phi}_{th} \Bigg) \\
&= \frac{1}{\alpha_0(\alpha_0+1)(\alpha_0+2)} \Bigg( \alpha_0^3 \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_s \circ \boldsymbol{\mu}_t \\
&\qquad + \alpha_0^2(\alpha_0+1)\mathbb{E}(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{\mu}_t) - \alpha_0^3 \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_s \circ \boldsymbol{\mu}_t \\
&\qquad + \alpha_0^2(\alpha_0+1)\mathbb{E}(\boldsymbol{\mu}_j \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it}) - \alpha_0^3 \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_s \circ \boldsymbol{\mu}_t \\
&\qquad + \alpha_0^2(\alpha_0+1)\mathbb{E}(\boldsymbol{b}_{ij} \circ \boldsymbol{\mu}_s \circ \boldsymbol{b}_{it}) - \alpha_0^3 \boldsymbol{\mu}_j \circ \boldsymbol{\mu}_s \circ \boldsymbol{\mu}_t \\
&\qquad + 2 \sum_{h=1}^{k} \alpha_h \boldsymbol{\phi}_{jh} \circ \boldsymbol{\phi}_{sh} \circ \boldsymbol{\phi}_{th} \Bigg). \qquad \text{(B.2)}
\end{aligned}
$$

The theorem follows Equations (B.1) and (B.2) with $\boldsymbol{\mu}_j = \mathbb{E}(\boldsymbol{b}_{ij})$ plugged in. For non-categorical data, we let $\boldsymbol{b}_{ij} \equiv y_{ij}$ and $\boldsymbol{\phi}_{jh}$ is a scalar mean parameter for $y_{ij}$. Equations B.1 and B.2 still hold. □

## B.2 Proof of Theorem 3.2

*Proof.* For notation simplicity we suppress $\boldsymbol{A}_n^{(\cdot)}$ in $Q_n^{(\cdot)}(\boldsymbol{\Phi}; \boldsymbol{A}_n^{(\cdot)})$ and $\boldsymbol{A}^{(\cdot)}$ in $Q_0^{(\cdot)}(\boldsymbol{\Phi}; \boldsymbol{A}^{(\cdot)})$. Lemma 3.1 implies $\lim_{n\to\infty} \Pr[|Q_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) - Q_0^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})| < \epsilon/3] = 1$ and $\lim_{n\to\infty} \Pr[|Q_n^{(\cdot)}(\boldsymbol{\Phi}_0) - Q_0^{(\cdot)}(\boldsymbol{\Phi}_0)| < \epsilon/3] = 1$ for $\epsilon > 0$. This result also implies

$$
\lim_{n\to\infty} \Pr[Q_0^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) < Q_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) + \epsilon/3] = 1. \qquad \text{(B.3)}
$$

$$\lim_{n\to\infty} \Pr[Q_n^{(\cdot)}(\mathbf{\Phi}_0) < Q_0^{(\cdot)}(\mathbf{\Phi}_0) + \epsilon/3] = 1. \tag{B.4}$$

On the other hand, $\widehat{\mathbf{\Phi}}^{(\cdot)}$ minimizes $Q_n^{(\cdot)}(\mathbf{\Phi})$, therefore

$$\lim_{n\to\infty} \Pr[Q_n^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) < Q_n^{(\cdot)}(\mathbf{\Phi}_0) + \epsilon/3] = 1. \tag{B.5}$$

Equations B.3 and B.5 imply

$$\lim_{n\to\infty} \Pr[Q_0^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) < Q_n^{(\cdot)}(\mathbf{\Phi}_0) + 2\epsilon/3] = 1.$$

Together with Equation (B.4), we get

$$\lim_{n\to\infty} \Pr[Q_0^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) < Q_0^{(\cdot)}(\mathbf{\Phi}_0) + \epsilon] = 1.$$

Therefore

$$\lim_{n\to\infty} \Pr[0 \leqslant Q_0^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) < \epsilon] = 1 \tag{B.6}$$

follows with $Q_0^{(\cdot)}(\mathbf{\Phi}_0) = 0$ and $Q_0^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) \geqslant 0$. Next, we choose a neighborhood $N$, which contains $\mathbf{\Phi}_0$ in $\mathbf{\Theta}$. Due to the compactness of $\mathbf{\Theta}$ the neighborhood $N^C$ is also compact. The continuousness of $Q_0^{(\cdot)}(\mathbf{\Phi})$ implies the existence of $\inf_{\mathbf{\Phi}\in N^C} Q_0^{(\cdot)}(\mathbf{\Phi})$ and it is positive. Let $\epsilon = \inf_{\mathbf{\Phi}\in N^C} Q_0^{(\cdot)}(\mathbf{\Phi})$, then we get

$$\lim_{n\to\infty} \Pr[0 \leqslant Q_0^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) < \inf_{\mathbf{\Phi}\in N^C} Q_0^{(\cdot)}(\mathbf{\Phi})] = 1. \tag{B.7}$$

Therefore $\lim_{n\to\infty} \Pr(\widehat{\mathbf{\Phi}}^{(\cdot)} \notin N^C) = 1$, which suggests $\lim_{n\to\infty} \Pr(\widehat{\mathbf{\Phi}}^{(\cdot)} \in N) = 1$. Shrinking the neighborhood size of $N$ we get

$$\lim_{n\to\infty} \Pr(\widehat{\mathbf{\Phi}}^{(\cdot)} = \mathbf{\Phi}_0) = 1.$$

□

## B.3 Proof of Theorem 3.3

*Proof.* We approximate $\boldsymbol{f}_n^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)})$ using first order Taylor expansion

$$\boldsymbol{f}_n^{(\cdot)}(\widehat{\mathbf{\Phi}}^{(\cdot)}) = \boldsymbol{f}_n^{(\cdot)}(\mathbf{\Phi}_0) + \boldsymbol{G}_n^{(\cdot)}(\mathbf{\Phi}_0)[\mathrm{vec}(\widehat{\mathbf{\Phi}}^{(\cdot)}) - \mathrm{vec}(\mathbf{\Phi}_0)] + O\{[\mathrm{vec}(\widehat{\mathbf{\Phi}}^{(\cdot)}) - \mathrm{vec}(\mathbf{\Phi}_0)]^2\}$$

Ignoring the high order term, we left multiply both sides by $[\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}$. Then we get

$$[\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}\boldsymbol{f}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) \approx [\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}\boldsymbol{f}_n^{(\cdot)}(\boldsymbol{\Phi}_0) + [\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}\boldsymbol{G}_n^{(\cdot)}(\boldsymbol{\Phi}_0)[\mathrm{vec}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) - \mathrm{vec}(\boldsymbol{\Phi}_0)].$$

The fact that estimator $\widehat{\boldsymbol{\Phi}}^{(\cdot)}$ minimizes $Q_n^{(\cdot)}(\boldsymbol{\Phi}, \boldsymbol{A}_n^{(\cdot)})$ implies the left hand side equals to zero. Therefore we get

$$n^{1/2}[\mathrm{vec}(\widehat{\boldsymbol{\Phi}}^{(\cdot)}) - \mathrm{vec}(\boldsymbol{\Phi}_0)] \approx -\{[\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}\boldsymbol{G}_n^{(\cdot)}(\boldsymbol{\Phi}_0)\}^{-1}[\boldsymbol{G}_n^{(\cdot)}(\widehat{\boldsymbol{\Phi}}^{(\cdot)})]^T\boldsymbol{A}_n^{(\cdot)}n^{1/2}\boldsymbol{f}_n^{(\cdot)}(\boldsymbol{\Phi}_0).$$

The theorem follows with $n^{1/2}\boldsymbol{f}_n^{(\cdot)}(\boldsymbol{\Phi}_0) \xrightarrow{p} \mathrm{N}(\boldsymbol{0}, \boldsymbol{S}^{(\cdot)})$ and Assumptions 3.1 and 3.2. □

## B.4 Derivatives of moment functions

*Second moment matrix*

The second moment matrix $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ in the main paper may be written as $\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it} - \boldsymbol{\Phi}_j\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)\boldsymbol{\Phi}_t^T$. The derivatives of $\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ with respect to $\boldsymbol{\Phi}_j$ and $\boldsymbol{\Phi}_t$ can be written as

$$\begin{aligned}
\frac{\partial \mathrm{vec}[\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]}{\partial \mathrm{vec}(\boldsymbol{\Phi}_j)} &= -\frac{\partial\{[\boldsymbol{\Phi}_t\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)] \otimes \boldsymbol{I}_{d_j}\}\mathrm{vec}(\boldsymbol{\Phi}_j)}{\partial \mathrm{vec}(\boldsymbol{\Phi}_j)} \\
&= -[\boldsymbol{\Phi}_t\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)] \otimes \boldsymbol{I}_{d_j}, \\
\frac{\partial \mathrm{vec}[\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]}{\partial \mathrm{vec}(\boldsymbol{\Phi}_t)} &= -\boldsymbol{T}\frac{\partial \mathrm{vec}[\boldsymbol{\Phi}_t\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)\boldsymbol{\Phi}_j^T]}{\partial \mathrm{vec}(\boldsymbol{\Phi}_t)} \\
&= -\boldsymbol{T}\frac{\partial\{[\boldsymbol{\Phi}_j\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)] \otimes \boldsymbol{I}_{d_t}\mathrm{vec}(\boldsymbol{\Phi}_t)\}}{\partial \mathrm{vec}(\boldsymbol{\Phi}_t)} \\
&= -\boldsymbol{T}\{[\boldsymbol{\Phi}_j\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)] \otimes \boldsymbol{I}_{d_t}\},
\end{aligned}$$

where $\otimes$ indicates a Kronecker product and $\boldsymbol{T}$ is a $d_tk \times d_tk$ 0/1 matrix that satisfies

$$\mathrm{vec}[\boldsymbol{\Phi}_j\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)\boldsymbol{\Phi}_t^T] = \boldsymbol{T}\mathrm{vec}[\boldsymbol{\Phi}_t\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i)\boldsymbol{\Phi}_j^T].$$

Therefore $\mathbb{E}[\partial\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})/\partial\boldsymbol{\Phi}]$ is a block matrix with block of $-\boldsymbol{\Phi}_t^T\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i) \otimes \boldsymbol{I}_{d_j}$ on columns corresponding to $\mathrm{vec}(\boldsymbol{\Phi}_t)$ and rows corresponding to $\mathrm{vec}[\boldsymbol{F}_{jt}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]$.

*Third moment tensor*

We next consider the third moment tensor. Write $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ as $\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{ij} - \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i) \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t$. Then only the second term involves $\boldsymbol{\Phi}$.

The derivatives of $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ with respect to $\boldsymbol{\Phi}_j$ can be written as

$$\begin{aligned}
\frac{\partial \text{vec}[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]}{\partial \text{vec}(\boldsymbol{\Phi}_j)} &= -\frac{\partial \text{vec}\{[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i) \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t]_{(1)}\}}{\partial \text{vec}(\boldsymbol{\Phi}_j)} \\
&= -\frac{\partial \text{vec}\{[\boldsymbol{\Phi}_j \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(1)} (\boldsymbol{\Phi}_t \otimes \boldsymbol{\Phi}_s)^T\}}{\partial \text{vec}(\boldsymbol{\Phi}_j)} \\
&= -(\boldsymbol{\Phi}_t \otimes \boldsymbol{\Phi}_s) \text{vec}[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(1)}]^T \otimes \boldsymbol{I}_{d_j},
\end{aligned}$$

where subscript (1) indicates model-1 unfolding of a three way tensor.

The derivatives of $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ with respect to $\boldsymbol{\Phi}_s$ and $\boldsymbol{\Phi}_t$ can be calculated accordingly by introducing 0/1 transformation matrices $\boldsymbol{T}_{(2)(1)}$ and $\boldsymbol{T}_{(3)(1)}$ both with size $d_j d_s d_t \times d_j d_s d_t$ that satisfy

$$\begin{aligned}
\text{vec}\{[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]_{(1)}\} &= \boldsymbol{T}_{(2)(1)} \text{vec}\{[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]_{(2)}\} \\
&= \boldsymbol{T}_{(3)(1)} \text{vec}\{[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]_{(3)}\}.
\end{aligned}$$

Then

$$\begin{aligned}
\frac{\partial \text{vec}[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]}{\partial \text{vec}(\boldsymbol{\Phi}_s)} &= -\boldsymbol{T}_{(2)(1)} \frac{\partial \text{vec}\{[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i) \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t]_{(2)}\}}{\partial \text{vec}(\boldsymbol{\Phi}_s)} \\
&= -\boldsymbol{T}_{(2)(1)} \frac{\partial \text{vec}\{[\boldsymbol{\Phi}_s \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(2)} (\boldsymbol{\Phi}_t \otimes \boldsymbol{\Phi}_j)^T\}}{\partial \text{vec}(\boldsymbol{\Phi}_s)} \\
&= -\boldsymbol{T}_{(2)(1)} \{(\boldsymbol{\Phi}_t \otimes \boldsymbol{\Phi}_j)[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(2)}]^T \otimes \boldsymbol{I}_{d_s}\}, \\
\frac{\partial \text{vec}[\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})]}{\partial \text{vec}(\boldsymbol{\Phi}_t)} &= -\boldsymbol{T}_{(3)(1)} \frac{\partial \text{vec}\{[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i) \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t]_{(3)}\}}{\partial \text{vec}(\boldsymbol{\Phi}_t)} \\
&= -\boldsymbol{T}_{(3)(1)} \frac{\partial \text{vec}\{[\boldsymbol{\Phi}_t \mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(3)} (\boldsymbol{\Phi}_s \otimes \boldsymbol{\Phi}_j)^T\}}{\partial \text{vec}(\boldsymbol{\Phi}_t)} \\
&= -\boldsymbol{T}_{(3)(1)} \{(\boldsymbol{\Phi}_s \otimes \boldsymbol{\Phi}_j)[\mathbb{E}(\boldsymbol{x}_i \circ \boldsymbol{x}_i \circ \boldsymbol{x}_i)_{(3)}]^T \otimes \boldsymbol{I}_{d_t}\}.
\end{aligned}$$

The conditions 1), 3) and 4) in Assumption 2 in main text follow after we calculating the derivatives of moment functions.

## B.5 Derivation of Newton-Raphson update

We denote

$$\boldsymbol{E}_{n,jt}^{(2)} = \boldsymbol{F}_{n,jt}^{(2)}(\boldsymbol{\Phi}) + \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T,$$

$$\boldsymbol{E}_{n,jst}^{(3)} = \boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi}) + \boldsymbol{\Lambda}^{(3)} \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t.$$

Then the two objective functions can be written as

$$Q^{(2)}(\boldsymbol{\Phi}, \boldsymbol{I}) = \sum_{j=1}^{p-1} \sum_{t=j+1}^{p} ||\boldsymbol{E}_{n,jt}^{(2)} - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T||_F^2,$$

$$Q^{(3)}(\boldsymbol{\Phi}, \boldsymbol{I}) = \sum_{j=1}^{p-1} \sum_{t=j+1}^{p} ||\boldsymbol{E}_{n,jt}^{(2)} - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T||_F^2 + \sum_{j=1}^{p-2} \sum_{s=j+1}^{p-1} \sum_{t=s+1}^{p} ||\boldsymbol{E}_{n,jst}^{(3)} - \boldsymbol{\Lambda}^{(3)} \times_1 \boldsymbol{\Phi}_j \times_2 \boldsymbol{\Phi}_s \times_3 \boldsymbol{\Phi}_t||_F^2.$$

We first consider $Q^{(2)}(\boldsymbol{\Phi}, \boldsymbol{I})$. The terms involve $\boldsymbol{\phi}_{jh}$ are

$$\sum_{t=1, t\neq j}^{p} \Big[ -2\lambda_h^{(2)} (\overline{\boldsymbol{E}}_{n,jt}^{(2)} \boldsymbol{\phi}_{th})^T \boldsymbol{\phi}_{jh} + (\lambda_h^{(2)})^2 (\boldsymbol{\phi}_{th}^T \boldsymbol{\phi}_{th}) \boldsymbol{\phi}_{jh}^T \boldsymbol{\phi}_{jh} \Big],$$

where $\overline{\boldsymbol{E}}_{n,jt}^{(2)} = \boldsymbol{E}_{n,jt}^{(2)} - \sum_{h' \neq h} \lambda_{h'}^{(2)} \boldsymbol{\phi}_{jh'} \circ \boldsymbol{\phi}_{th'}$ and $\lambda_h^{(2)}$ is the $h$th diagonal element of $\boldsymbol{\Lambda}^{(2)}$. Here we use the fact that $||\boldsymbol{E}_{n,jt}^{(2)} - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T||_F^2 = ||\boldsymbol{E}_{n,tj}^{(2)} - \boldsymbol{\Phi}_t \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_j^T||_F^2$. By letting

$$\boldsymbol{\xi}^{(2)} = -2\lambda_h^{(2)} \sum_{t=1, t\neq j}^{p} (\overline{\boldsymbol{E}}_{jt}^{(2)} \boldsymbol{\phi}_{th}),$$

$$\gamma^{(2)} = (\lambda_h^{(2)})^2 \sum_{t=1, t\neq j}^{p} \boldsymbol{\phi}_{th}^T \boldsymbol{\phi}_{th},$$

the gradient $\nabla Q^{(2)}(\boldsymbol{\phi}_{jh})$ and Hessian $\nabla^2 Q^{(2)}(\boldsymbol{\phi}_{jh})$ can be written as

$$\nabla Q^{(2)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I}) = \boldsymbol{\xi}^{(2)} + 2\gamma^{(2)}\boldsymbol{\phi}_{jh},$$

$$\nabla^2 Q^{(2)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I}) = 2\gamma^{(2)}\boldsymbol{I}.$$

The update rule in (3.18) can be derived accordingly.

Then we consider $Q^{(3)}(\boldsymbol{\Phi}, \boldsymbol{I})$. The terms involve $\boldsymbol{\phi}_{jh}$ are

$$\sum_{t=1,t\neq j}^{p}\Big[ -2\lambda_h^{(2)}(\overline{\boldsymbol{E}}_{n,jt}^{(2)}\boldsymbol{\phi}_{th})^T\boldsymbol{\phi}_{jh} + (\lambda_h^{(2)})^2(\boldsymbol{\phi}_{th}^T\boldsymbol{\phi}_{th})\boldsymbol{\phi}_{jh}^T\boldsymbol{\phi}_{jh}\Big]$$

$$+ \sum_{s=1,s\neq j}^{p}\sum_{t=1,t\neq s,t\neq j}^{p}\Big[ -2\langle\overline{\boldsymbol{E}}_{n,jst}^{(3)}, \lambda_h^{(3)}\boldsymbol{\phi}_{jh}\circ\boldsymbol{\phi}_{sh}\circ\boldsymbol{\phi}_{th}\rangle + ||\lambda_h^{(3)}\boldsymbol{\phi}_{jh}\circ\boldsymbol{\phi}_{sh}\circ\boldsymbol{\phi}_{th}||_F^2\Big],$$

where $\overline{\boldsymbol{E}}_{n,jt}^{(2)} = \boldsymbol{E}_{n,jt}^{(2)} - \sum_{h'\neq h}\lambda_{h'}^{(2)}\boldsymbol{\phi}_{jh'}\circ\boldsymbol{\phi}_{th'}$ and $\overline{\boldsymbol{E}}_{n,jst}^{(3)} = \boldsymbol{E}_{n,jst}^{(3)} - \sum_{h'\neq h}\lambda_{h'}^{(3)}\boldsymbol{\phi}_{jh'}\circ\boldsymbol{\phi}_{sh'}\circ\boldsymbol{\phi}_{th'}$. Again we use the symmetric property of $||\boldsymbol{E}_{n,jt}^{(2)} - \boldsymbol{\Phi}_j\boldsymbol{\Lambda}^{(2)}\boldsymbol{\Phi}_t^T||_F^2$ and the super-symmetric property of $||\boldsymbol{E}_{n,jst}^{(3)} - \boldsymbol{\Lambda}^{(3)}\times_1\boldsymbol{\Phi}_j\times_2\boldsymbol{\Phi}_s\times_3\boldsymbol{\Phi}_t||_F^2$. By organizing the terms, we get

$$\sum_{t=1,t\neq j}^{p}\Big[ -2\lambda_h^{(2)}(\overline{\boldsymbol{E}}_{n,jt}^{(2)}\boldsymbol{\phi}_{th})^T\boldsymbol{\phi}_{jh} + (\lambda_h^{(2)})^2(\boldsymbol{\phi}_{th}^T\boldsymbol{\phi}_{th})\boldsymbol{\phi}_{jh}^T\boldsymbol{\phi}_{jh}\Big]$$

$$+ \sum_{s=1,s\neq j}^{p}\sum_{t=1,t\neq s,t\neq j}^{p}\Big[ -2\lambda_h^{(3)}(\overline{\boldsymbol{E}}_{n,jst}^{(3)}\times_2\boldsymbol{\phi}_{sh}\times_3\boldsymbol{\phi}_{th})^T\boldsymbol{\phi}_{jh} + (\lambda_h^{(3)})^2(\boldsymbol{\phi}_{sh}^T\boldsymbol{\phi}_{sh})(\boldsymbol{\phi}_{th}^T\boldsymbol{\phi}_{th})(\boldsymbol{\phi}_{jh}^T\boldsymbol{\phi}_{jh})\Big].$$

We let

$$\boldsymbol{\xi}^{(3)} = -2\lambda_h^{(2)}\sum_{t=1,t\neq j}^{p}(\overline{\boldsymbol{E}}_{n,jt}^{(2)}\boldsymbol{\phi}_{th}) - 2\lambda_h^{(3)}\sum_{s=1,s\neq j}^{p}\Big[\sum_{t=1,t\neq s,t\neq j}^{p}(\overline{\boldsymbol{E}}_{n,jst}^{(3)}\times_2\boldsymbol{\phi}_{sh}\times_3\boldsymbol{\phi}_{th})\Big],$$

$$\gamma^{(3)} = (\lambda_h^{(2)})^2\sum_{t=1,t\neq j}^{p}\boldsymbol{\phi}_{th}^T\boldsymbol{\phi}_{th} + (\lambda_h^{(3)})^2\sum_{s=1,s\neq j}^{p}\Big[\sum_{t=1,t\neq s,t\neq j}^{p}(\boldsymbol{\phi}_{sh}^T\boldsymbol{\phi}_{sh})(\boldsymbol{\phi}_{th}^T\boldsymbol{\phi}_{th})\Big],$$

and then the gradient $\nabla Q^{(3)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I})$ and Hessian $\nabla^2 Q^{(3)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I})$ can be written as

$$\nabla Q^{(3)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I}) = \boldsymbol{\xi}^{(3)} + 2\gamma^{(3)}\boldsymbol{\phi}_{jh},$$

$$\nabla^2 Q^{(3)}(\boldsymbol{\phi}_{jh}, \boldsymbol{I}) = 2\gamma^{(3)}\boldsymbol{I}.$$

The update rule in (3.19) follows directly.

## B.6 Optimal weight matrices

### *B.6.1 Derivation of weight matrix for moment vector using second moment matrices*

We encode the observations $y_{ij}$ as $\boldsymbol{b}_{ij}$ with $\boldsymbol{F}^{(2)}_{jt}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ defined as

$$\begin{aligned}\boldsymbol{F}^{(2)}_{jt}(\boldsymbol{y}_i, \boldsymbol{\Phi}) &= \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it} - \frac{\alpha_0}{\alpha_0 + 1}\mathbb{E}(\boldsymbol{b}_{ij}) \circ \mathbb{E}(\boldsymbol{b}_{it}) - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T \\ &= \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it} - \frac{\alpha_0}{\alpha_0 + 1}\boldsymbol{\mu}_j \circ \boldsymbol{\mu}_t - \boldsymbol{\Phi}_j \boldsymbol{\Lambda}^{(2)} \boldsymbol{\Phi}_t^T.\end{aligned}$$

In addition, $\mathbb{E}[\boldsymbol{F}^{(2)}_{jt}(\boldsymbol{y}_i, \boldsymbol{\Phi})] = \boldsymbol{0}$.

Estimation of the parameters $\boldsymbol{\Phi}$ requires the calculation of $\widehat{\boldsymbol{\mu}}_j$ and $\widehat{\boldsymbol{\mu}}_t$ and then plugging these values into the equation for $\boldsymbol{F}^{(2)}_{jt}(\boldsymbol{y}_i, \boldsymbol{\Phi})$. The expectations of the moment functions are then calculated by sample averages, denoted as $\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})$. The first stage of estimation involves the minimization of a quadratic function defined as the Frobenius norm of $\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})$

$$Q^{(2)}_n(\boldsymbol{\Phi}, \boldsymbol{I}) = \sum_{j=1}^{p-1} \sum_{t=j+1}^{p} \|\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})\|_F,$$

where we use $\widehat{\boldsymbol{\mu}}_j$ and $\widehat{\boldsymbol{\mu}}_t$ in the above computation rather than considering $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_t$ as unknown parameters. Minimizing the above functional provides an estimate $\widehat{\boldsymbol{\Phi}}$.

Given the estimate $\widehat{\boldsymbol{\Phi}}$ we re-compute $\widehat{\boldsymbol{\mu}}_j$ and $\widehat{\boldsymbol{\mu}}_t$ based on the first moment equations. We define following moment vector and its estimated expectation

$$\begin{aligned}\boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) &= \left(\text{vec}[\boldsymbol{F}^{(2)}_{12}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^T, \ldots, \text{vec}[\boldsymbol{F}^{(2)}_{p-1,p}(\boldsymbol{y}_i, \boldsymbol{\Phi})]^T\right)^T, \\ \boldsymbol{f}^{(2)}_n(\boldsymbol{\Phi}) &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}^{(2)}(\boldsymbol{y}_i, \boldsymbol{\Phi}).\end{aligned}$$

We then compute a variance-covariance matrix of $n^{1/2}\boldsymbol{f}^{(2)}_n(\boldsymbol{\Phi})$ to compute an optimal weight matrix. The covariance matrix is composed by the matrix blocks of

the variance-covariance of $n^{1/2}\text{vec}[\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})]$ and $n^{1/2}\text{vec}[\boldsymbol{F}^{(2)}_{n,su}(\boldsymbol{\Phi})]$ for every possible $j < t$ and $s < u$ combinations in $n^{1/2}\boldsymbol{f}^{(2)}_n(\boldsymbol{\Phi})$. We denote such a block as ${}_{jt}\boldsymbol{\Sigma}_{su}$. Equation (B.8) illustrates the covariance matrix. In the following, we are going to compute the covariance matrix by dividing the matrix into four parts: diagonal elements, off diagonal elements in ${}_{jt}\boldsymbol{\Sigma}_{jt}$, elements in ${}_{jt}\boldsymbol{\Sigma}_{su}$ with one matching variable (three possible cases $j = s$ or $t = s$ or $t = u$) and elements in ${}_{jt}\boldsymbol{\Sigma}_{su}$ with $(j,t) \neq (s,u)$.

$$
\begin{array}{c}
\text{index} \\
\text{vec}[\boldsymbol{F}^{(2)}_{n,12}(\boldsymbol{\Phi})] \\
\vdots \\
\text{vec}[\boldsymbol{F}^{(2)}_{n,1p}(\boldsymbol{\Phi})] \\
\text{vec}[\boldsymbol{F}^{(2)}_{n,23}(\boldsymbol{\Phi})] \\
\vdots \\
\text{vec}[\boldsymbol{F}^{(2)}_{n,(p-1,p)}(\boldsymbol{\Phi})]
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
\text{vec}[\boldsymbol{F}^{(2)}_{n,12}(\boldsymbol{\Phi})] & \text{vec}[\boldsymbol{F}^{(2)}_{n,13}(\boldsymbol{\Phi})] & \dots & \text{vec}[\boldsymbol{F}^{(2)}_{n,1p}(\boldsymbol{\Phi})] & \text{vec}[\boldsymbol{F}^{(2)}_{n,23}(\boldsymbol{\Phi})] & \dots & \text{vec}[\boldsymbol{F}^{(2)}_{n,(p-1,p)}(\boldsymbol{\Phi})]
\end{array} \\
\left(\begin{array}{cccccccc}
{}_{12}\boldsymbol{\Sigma}_{12} & {}_{12}\boldsymbol{\Sigma}_{13} & \dots & {}_{12}\boldsymbol{\Sigma}_{1p} & {}_{12}\boldsymbol{\Sigma}_{23} & \dots & {}_{12}\boldsymbol{\Sigma}_{p-1,p} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
{}_{1p}\boldsymbol{\Sigma}_{12} & {}_{1p}\boldsymbol{\Sigma}_{13} & \dots & {}_{1p}\boldsymbol{\Sigma}_{1p} & {}_{1p}\boldsymbol{\Sigma}_{23} & \dots & {}_{1p}\boldsymbol{\Sigma}_{p-1,p} \\
{}_{23}\boldsymbol{\Sigma}_{12} & {}_{23}\boldsymbol{\Sigma}_{13} & \dots & {}_{23}\boldsymbol{\Sigma}_{1p} & {}_{23}\boldsymbol{\Sigma}_{23} & \dots & {}_{23}\boldsymbol{\Sigma}_{p-1,p} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
{}_{p-1,p}\boldsymbol{\Sigma}_{12} & {}_{p-1,p}\boldsymbol{\Sigma}_{13} & \dots & {}_{p-1,p}\boldsymbol{\Sigma}_{1p} & {}_{p-1,p}\boldsymbol{\Sigma}_{23} & \dots & {}_{p-1,p}\boldsymbol{\Sigma}_{p-1,p}
\end{array}\right)
\end{array}
\tag{B.8}
$$

*Part one, diagonal elements.* The variance of the $(c_j, c_t)$ element of $n^{1/2}\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})$ for $c_j = 1, \dots, d_j$, $c_t = 1, \dots, d_t$, $j = 1, \dots, p-1$ and $t = j+1, \dots, p$ composes the diagonal elements of the covariance matrix. Its value can be calculated as

$$
\text{Var}\left(n^{1/2}\boldsymbol{F}_{n,jt}(\boldsymbol{\Phi})_{c_j,c_t}\right) = \text{Var}\left((\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it})_{c_j,c_t}\right) = \mathbb{E}\left([(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it})_{c_j,c_t}]^2\right) - \mathbb{E}^2\left((\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it})_{c_j,c_t}\right).
$$

We have already derived the second term. The first term is computed as follows:

$$
\begin{aligned}
\mathbb{E}\left([(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{it})_{c_j,c_t}]^2\right) &= \mathbb{E}\left((\boldsymbol{\phi}^T_{j\cdot c_j}\boldsymbol{m}_{ij})^2(\boldsymbol{\phi}^T_{t\cdot c_t}\boldsymbol{m}_{it})^2\right) \\
&= \mathbb{E}\left(\boldsymbol{\phi}^T_{j\cdot c_j}(\boldsymbol{m}_{ij} \circ \boldsymbol{m}_{ij})\boldsymbol{\phi}_{j\cdot c_j}\boldsymbol{\phi}^T_{t\cdot c_t}(\boldsymbol{m}_{it} \circ \boldsymbol{m}_{it})\boldsymbol{\phi}_{t\cdot c_t}\right) \\
&= \mathbb{E}\left(\boldsymbol{\phi}^T_{j\cdot c_j}[\text{diag}(\boldsymbol{x}_i)]\boldsymbol{\phi}_{j\cdot c_j}\boldsymbol{\phi}^T_{t\cdot c_t}[\text{diag}(\boldsymbol{x}_i)]\boldsymbol{\phi}_{t\cdot c_t}\right) \\
&= \mathbb{E}\left(\sum_{h_1}\sum_{h_2}\phi^2_{jh_1c_j}\phi^2_{th_2c_t}x_{ih_1}x_{ih_2}\right) \\
&= \sum_{h_1}\sum_{h_2}\phi^2_{jh_1c_j}\phi^2_{th_2c_t}\mathbb{E}(x_{ih_1}x_{ih_2}).
\end{aligned}
\tag{B.9}
$$

To calculate $\mathbb{E}(x_{ih_1}x_{ih_2})$ we use the fact that for a Dirichlet distributed $\boldsymbol{x}_i$

$$\mathbb{E}\bigg(\prod_h x_{ih}^{r_h}\bigg) = \frac{\Gamma(a_0)}{\Gamma(a_0+r_0)} \times \prod_h \frac{\Gamma(a_h+r_h)}{\Gamma(a_h)}, \tag{B.10}$$

where $\Gamma(\cdot)$ is a gamma function and $r_0 = \sum_h r_h$.

*Part two, off diagonal elements in ${}_{jt}\boldsymbol{\Sigma}_{jt}$.* We next consider the covariance between $n^{1/2}\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})_{c_j,c_t}$ and $n^{1/2}\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})_{g_j,g_t}$ for $c_j = 1,\ldots,d_j$ and $c_t = 1,\ldots,d_t$ with $(c_j,c_t) \neq (g_j,g_t)$. Elements of this kind constitute the off diagonal elements in the block of ${}_{jt}\boldsymbol{\Sigma}_{jt}$.

The computation requires the calculation of $\mathbb{E}[(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{g_j,g_t}]$

$$\begin{aligned}
\mathbb{E}\bigg((\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{g_j,g_t}\bigg) &= \mathbb{E}\bigg(\boldsymbol{\phi}^T_{j\cdot c_j}(\boldsymbol{m}_{ij}\circ\boldsymbol{m}_{ij})\boldsymbol{\phi}_{j\cdot g_j}\boldsymbol{\phi}^T_{t\cdot c_t}(\boldsymbol{m}_{it}\circ\boldsymbol{m}_{it})\boldsymbol{\phi}_{t\cdot g_t}\bigg)\\
&= \mathbb{E}\bigg(\boldsymbol{\phi}^T_{j\cdot c_j}[\mathrm{diag}(\boldsymbol{x}_i)]\boldsymbol{\phi}_{j\cdot g_j}\boldsymbol{\phi}^T_{t\cdot c_t}[\mathrm{diag}(\boldsymbol{x}_i)]\boldsymbol{\phi}_{t\cdot g_t}\bigg)\\
&= \mathbb{E}\bigg(\sum_{h_1}\sum_{h_2}\phi_{jh_1c_j}\phi_{jh_1g_j}\phi_{th_2c_t}\phi_{th_2g_t}x_{ih_1}x_{ih_2}\bigg)\\
&= \sum_{h_1}\sum_{h_2}\phi_{jh_1c_j}\phi_{jh_1g_j}\phi_{th_2c_t}\phi_{th_2g_t}\mathbb{E}(x_{ih_1}x_{ih_2}).
\end{aligned} \tag{B.11}$$

*Part three, elements in ${}_{jt}\boldsymbol{\Sigma}_{su}$ with one matching variable.* We next calculate the elements in ${}_{jt}\boldsymbol{\Sigma}_{su}$ with one matching variable, the case either $j = s$ or $t = s$ or $t = u$ . The calculation of $\mathbb{E}[(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{is}\circ\boldsymbol{b}_{iu})_{g_s,g_u}]$ is required

$$\mathbb{E}\bigg((\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{is}\circ\boldsymbol{b}_{iu})_{g_s,g_u}\bigg) = \mathbb{E}\bigg((\boldsymbol{\phi}^T_{j\cdot c_j}\boldsymbol{m}_{ij})(\boldsymbol{\phi}^T_{t\cdot c_t}\boldsymbol{m}_{it})(\boldsymbol{\phi}^T_{s\cdot g_s}\boldsymbol{m}_{is})(\boldsymbol{\phi}^T_{u\cdot g_u}\boldsymbol{m}_{iu})\bigg).$$

Without loss of generality, consider the case where $j = s$ and $t \neq u$

$$\begin{aligned}
\mathbb{E}\bigg((\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{is}\circ\boldsymbol{b}_{iu})_{g_s,g_u}\bigg) &= \mathbb{E}\bigg(\boldsymbol{\phi}^T_{j\cdot c_j}[\mathrm{diag}(\boldsymbol{x}_i)]\boldsymbol{\phi}_{j\cdot g_j}\boldsymbol{\phi}^T_{t\cdot c_t}[\boldsymbol{x}_i\circ\boldsymbol{x}_i]\boldsymbol{\phi}_{u\cdot g_u}\bigg)\\
&= \mathbb{E}\bigg(\sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}\phi_{jh_1g_j}\phi_{th_2c_t}\phi_{uh_3g_u}x_{ih_1}x_{ih_2}x_{ih_3}\bigg)
\end{aligned}$$

$$= \sum_{h_1}\sum_{h_2}\sum_{h_3} \phi_{jh_1c_j}\phi_{jh_1g_j}\phi_{th_2c_t}\phi_{uh_3g_u}\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}). \tag{B.12}$$

Other elements in the block can be calculated similarly.

*Part four, elements in ${}_{jt}\boldsymbol{\Sigma}_{su}$ with $(j,t) \neq (s,u)$.* We finally consider the covariance between $n^{1/2}\boldsymbol{F}^{(2)}_{n,jt}(\boldsymbol{\Phi})_{c_j,c_t}$ and $n^{1/2}\boldsymbol{F}^{(2)}_{n,su}(\boldsymbol{\Phi})_{g_s,g_u}$ with $c_j = 1,\ldots,d_j$, $c_t = 1,\ldots,d_t$, $g_s = 1,\ldots d_s$, $g_u = 1,\ldots,d_u$ for $(j,t) \neq (s,u)$. The unknown term $\mathbb{E}[(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{is}\circ \boldsymbol{b}_{iu})_{g_s,g_u}]$ can be written as

$$\mathbb{E}\bigg((\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{it})_{c_j,c_t}(\boldsymbol{b}_{is}\circ\boldsymbol{b}_{iu})_{g_s,g_u}\bigg) = \mathbb{E}\bigg((\boldsymbol{\phi}^T_{j\cdot c_j}\boldsymbol{m}_{ij})(\boldsymbol{\phi}^T_{t\cdot c_t}\boldsymbol{m}_{it})(\boldsymbol{\phi}^T_{s\cdot g_s}\boldsymbol{m}_{is})(\boldsymbol{\phi}^T_{u\cdot g_u}\boldsymbol{m}_{iu})\bigg)$$

$$= \sum_{h_1}\sum_{h_2}\sum_{h_3}\sum_{h_4} \phi_{jh_1c_j}\phi_{th_2c_t}\phi_{sh_3g_s}\phi_{uh_4g_u}\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}x_{ih_4}). \tag{B.13}$$

The optimal weight matrix for $\boldsymbol{f}^{(2)}_n(\boldsymbol{\Phi})$ is computed by combining the four parts in (B.9), (B.11), (B.12) and (B.13) and the first moment conditions derived in the main text. The size of the weight matrix is of the order $O(p^2d^2)$ and it is dense with full rank. Inverting this matrix is computationally intensive so in practice we invert the diagonal of the matrix to provide a near optimal weight in the second stage of optimization.

### *B.6.2 Derivation of weight matrix for moment vector using both second moment matrices and third moment tensors*

In this section we derive the weight matrix for $\boldsymbol{f}^{(3)}_n(\boldsymbol{\Phi})$. Define $\boldsymbol{F}^{(3)}_{jst}(\boldsymbol{y}_i,\boldsymbol{\Phi})$ as

$$\begin{aligned}\boldsymbol{F}^{(3)}_{jst}(\boldsymbol{y}_i,\boldsymbol{\Phi}) &= \boldsymbol{b}_{ij}\circ\boldsymbol{b}_{is}\circ\boldsymbol{b}_{it}\\ &- \frac{\alpha_0}{\alpha_0+2}\bigg(\boldsymbol{b}_{ij}\circ\boldsymbol{b}_{is}\circ\mathbb{E}(\boldsymbol{b}_{it}) + \mathbb{E}(\boldsymbol{b}_{ij})\circ\boldsymbol{b}_{is}\circ\boldsymbol{b}_{it} + \boldsymbol{b}_{ij}\circ\mathbb{E}(\boldsymbol{b}_{is})\circ\boldsymbol{b}_{it}\bigg)\\ &+ \frac{2\alpha_0^2}{(\alpha_0+1)(\alpha_0+2)}\mathbb{E}(\boldsymbol{b}_{ij})\circ\mathbb{E}(\boldsymbol{b}_{is})\circ\mathbb{E}(\boldsymbol{b}_{it}) - \boldsymbol{\Lambda}^{(3)}\times_1\boldsymbol{\Phi}_j\times_2\boldsymbol{\Phi}_s\times_3\boldsymbol{\Phi}_t,\end{aligned}$$

with $\mathbb{E}[\boldsymbol{F}^{(3)}_{jst}(\boldsymbol{y}_i,\boldsymbol{\Phi})] = \boldsymbol{0}$.

The optimal weight matrix for $\boldsymbol{f}_n^{(3)}(\boldsymbol{\Phi})$ can be derived following a similar way as the weight matrix derivation for $\boldsymbol{f}_n^{(2)}(\boldsymbol{\Phi})$. However the size of the matrix scales as $O(p^3d^3)$ which is prohibitive. For this reason we consider a near optimal weight matrix by only considering the diagonal elements of the matrix. We now derive the variance (diagonal) elements of $n^{1/2}\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})$. In the computations we will use $\widehat{\boldsymbol{\mu}}_j$ for $\mathbb{E}(\boldsymbol{b}_{ij})$. Computing the initial estimator $\widehat{\boldsymbol{\Phi}}$ involves minimizing $Q_n^{(3)}(\boldsymbol{\Phi}, \boldsymbol{A}_n^{(3)})$ with $\boldsymbol{A}_n^{(3)} = \boldsymbol{I}$.

We define the first two terms in $\boldsymbol{F}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$ as $\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})$

$$\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi}) = \boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it} - \frac{\alpha_0}{\alpha_0 + 2}\bigg(\boldsymbol{b}_{ij} \circ \boldsymbol{b}_{is} \circ \boldsymbol{\mu}_t + \boldsymbol{\mu}_j \circ \boldsymbol{b}_{is} \circ \boldsymbol{b}_{it} + \boldsymbol{b}_{ij} \circ \boldsymbol{\mu}_s \circ \boldsymbol{b}_{it}\bigg).$$

The variance of $n^{1/2}\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})_{c_jc_sc_t}$ can be written as

$$\begin{aligned}\mathrm{Var}\bigg(n^{1/2}\boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})_{c_jc_sc_t}\bigg) &= \mathrm{Var}\bigg(\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_jc_sc_t}\bigg) \\ &= \mathbb{E}\bigg([\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_jc_sc_t}]^2\bigg) - \mathbb{E}^2\bigg(\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_jc_sc_t}\bigg).\end{aligned}$$

The expectation of $\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_jc_sc_t}$ has been derived in the main text. We only need to consider the first term on the right hand side of the equation. After some algebra we get

$$\begin{aligned}\mathbb{E}\bigg([\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_jc_sc_t}]^2\bigg) &= \sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}^2\phi_{sh_2c_s}^2\phi_{th_3c_t}^2\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}) \\ &+ \frac{2\alpha_0\mu_{tc_t}}{\alpha_0+2}\sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}^2\phi_{sh_2c_s}^2\phi_{th_3c_t}\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}) \\ &+ \frac{2\alpha_0\mu_{sc_s}}{\alpha_0+2}\sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}^2\phi_{sh_2c_s}\phi_{th_3c_t}^2\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}) \\ &+ \frac{2\alpha_0\mu_{jc_j}}{\alpha_0+2}\sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}\phi_{sh_2c_s}^2\phi_{th_3c_t}^2\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3}) \\ &+ \frac{\alpha_0^2\mu_{jc_j}\mu_{sc_s}}{(\alpha_0+2)^2}\sum_{h_1}\sum_{h_2}\sum_{h_3}\phi_{jh_1c_j}\phi_{sh_2c_s}\phi_{th_3c_t}^2\mathbb{E}(x_{ih_1}x_{ih_2}x_{ih_3})\end{aligned}$$

$$+ \frac{\alpha_0^2 \mu_{jc_j} \mu_{tc_t}}{(\alpha_0+2)^2} \sum_{h_1} \sum_{h_2} \sum_{h_3} \phi_{jh_1c_j} \phi_{sh_2c_s}^2 \phi_{th_3c_t} \mathbb{E}(x_{ih_1} x_{ih_2} x_{ih_3})$$

$$+ \frac{\alpha_0^2 \mu_{sc_s} \mu_{tc_t}}{(\alpha_0+2)^2} \sum_{h_1} \sum_{h_2} \sum_{h_3} \phi_{jh_1c_j}^2 \phi_{sh_2c_s} \phi_{th_3c_t} \mathbb{E}(x_{ih_1} x_{ih_2} x_{ih_3})$$

$$+ \frac{\alpha_0^2 \mu_{tc_t}^2}{(\alpha_0+2)^2} \sum_{h_1} \sum_{h_2} \phi_{jh_1c_j}^2 \phi_{sh_2c_s}^2 \mathbb{E}(x_{ih_1} x_{ih_2})$$

$$+ \frac{\alpha_0^2 \mu_{sc_s}^2}{(\alpha_0+2)^2} \sum_{h_1} \sum_{h_2} \phi_{jh_1c_j}^2 \phi_{th_2c_t}^2 \mathbb{E}(x_{ih_1} x_{ih_2})$$

$$+ \frac{\alpha_0^2 \mu_{jc_j}^2}{(\alpha_0+2)^2} \sum_{h_1} \sum_{h_2} \phi_{sh_1c_s}^2 \phi_{th_2c_t}^2 \mathbb{E}(x_{ih_1} x_{ih_2}).$$

Together with $\mathbb{E}[\widetilde{\boldsymbol{F}}_{jst}^{(3)}(\boldsymbol{y}_i, \boldsymbol{\Phi})_{c_j c_s c_t}]$ we finish calculating the variance of $n^{1/2} \boldsymbol{F}_{n,jst}^{(3)}(\boldsymbol{\Phi})_{c_j c_s c_t}$.

Table B.1: Comparison of mean squared error (MSE) of parameter estimation for different methods in low dimensional categorical simulation. The MSE's are calculated using ten simulated data sets for each value of $n$. For SFM and LDA their MSE's are calculated based on posterior mean estimates from 100 posterior thinned samples using their MCMC algorithms. The standard deviations of the MSE's are provided in parenthesis.

| $n$ | $k$ | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage | $Q^{(3)}(\mathbf{\Phi})$ 1st stage | $Q^{(3)}(\mathbf{\Phi})$ 2nd stage | SFM | LDA |
|---|---|---|---|---|---|---|---|
| 50 | 1 | 0.043(0.002) | **0.042(0.002)** | 0.044(0.002) | 0.084(0.040) | 0.042(0.002) | 0.062(<0.000) |
| | 2 | **0.035(0.001)** | 0.046(0.010) | 0.035(0.002) | 0.075(0.035) | 0.039(0.002) | 0.062(<0.000) |
| | 3 | **0.036(0.002)** | 0.038(0.002) | 0.037(0.002) | 0.074(0.034) | 0.038(0.002) | 0.063(<0.000) |
| | 4 | 0.041(0.002) | 0.044(0.002) | 0.042(0.003) | 0.044(0.002) | **0.038(0.004)** | 0.063(0.001) |
| | 5 | 0.045(0.002) | 0.046(0.002) | 0.044(0.002) | 0.048(0.002) | **0.040(0.003)** | 0.063(0.001) |
| 100 | 1 | **0.041(0.001)** | 0.041(0.001) | 0.042(0.001) | 0.068(0.045) | 0.041(0.001) | 0.062(<0.000) |
| | 2 | 0.031(0.002) | 0.033(0.005) | **0.031(0.001)** | 0.044(0.029) | 0.034(0.001) | 0.062(<0.000) |
| | 3 | **0.033(0.002)** | 0.034(0.002) | 0.033(0.002) | 0.050(0.030) | 0.035(0.001) | 0.062(<0.000) |
| | 4 | 0.037(0.002) | 0.038(0.002) | 0.038(0.001) | 0.039(0.002) | **0.036(0.003)** | 0.062(<0.000) |
| | 5 | 0.041(0.003) | 0.041(0.003) | 0.039(0.002) | 0.043(0.002) | **0.036(0.002)** | 0.062(0.001) |
| 200 | 1 | **0.041(<0.000)** | 0.041(<0.000) | 0.042(<0.000) | 0.041(<0.000) | 0.041(<0.000) | 0.062(<0.000) |
| | 2 | 0.030(0.001) | 0.030(0.001) | **0.029(0.001)** | 0.029(0.001) | 0.032(0.001) | 0.062(<0.000) |
| | 3 | 0.033(0.001) | 0.033(0.001) | **0.032(0.001)** | 0.033(0.001) | 0.034(0.001) | 0.062(<0.000) |
| | 4 | 0.036(0.001) | 0.037(0.001) | 0.036(0.002) | 0.037(0.001) | **0.035(0.002)** | 0.062(<0.000) |
| | 5 | 0.038(<0.000) | 0.036(0.001) | 0.038(0.001) | 0.039(0.001) | **0.034(0.002)** | 0.062(<0.000) |
| 500 | 1 | **0.041(<0.000)** | 0.041(<0.000) | 0.041(<0.000) | 0.041(<0.000) | 0.041(0.001) | 0.062(<0.000) |
| | 2 | **0.029(0.001)** | 0.030(0.001) | 0.029(0.001) | 0.029(0.001) | 0.033(0.001) | 0.062(<0.000) |
| | 3 | **0.032(0.001)** | 0.032(0.001) | 0.032(0.001) | 0.032(0.001) | 0.033(0.001) | 0.062(<0.000) |
| | 4 | **0.035(0.001)** | 0.036(0.001) | 0.035(0.001) | 0.036(0.001) | 0.035(0.002) | 0.062(<0.000) |
| | 5 | 0.036(0.001) | **0.035(0.001)** | 0.036(0.001) | 0.038(0.001) | 0.036(0.002) | 0.062(<0.000) |
| 1,000 | 1 | **0.041(0.001)** | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) | 0.041(0.001) | 0.062(<0.000) |
| | 2 | 0.029(0.001) | 0.029(0.001) | **0.028(0.001)** | 0.028(0.001) | 0.032(0.001) | 0.062(<0.000) |
| | 3 | **0.031(0.001)** | 0.031(0.005) | 0.031(0.001) | 0.031(0.001) | 0.032(0.001) | 0.062(<0.000) |
| | 4 | **0.034(0.001)** | 0.034(0.001) | 0.034(0.001) | 0.034(0.001) | 0.034(0.001) | 0.062(<0.000) |
| | 5 | 0.036(0.001) | **0.034(0.001)** | 0.036(0.001) | 0.037(0.001) | 0.038(0.002) | 0.062(<0.000) |

Table B.2: Comparison of MSE of parameter estimation for different methods in low dimensional categorical simulation. The simulated ten data sets are contaminated by setting 4% of the samples to outliers. The MSE's are calculated with the same methods in Table B.1.

| $n$ | $k$ | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage | $Q^{(3)}(\mathbf{\Phi})$ 1st stage | $Q^{(3)}(\mathbf{\Phi})$ 2nd stage | SFM | LDA |
|---|---|---|---|---|---|---|---|
| | 1 | 0.043(0.002) | 0.043(0.002) | 0.044(0.002) | 0.061(0.024) | **0.042(0.002)** | 0.063(<0.000) |
| | 2 | **0.036(0.001)** | 0.039(0.005) | 0.036(0.003) | 0.058(0.038) | 0.042(0.002) | 0.063(0.001) |
| 50 | 3 | **0.037(0.002)** | 0.038(0.002) | 0.037(0.002) | 0.052(0.021) | 0.044(0.006) | 0.063(0.001) |
| | 4 | **0.041(0.001)** | 0.044(0.002) | 0.043(0.002) | 0.050(0.007) | 0.050(0.002) | 0.064(0.001) |
| | 5 | 0.046(0.001) | 0.048(0.002) | **0.044(0.002)** | 0.050(0.003) | 0.050(0.002) | 0.064(0.001) |
| | 1 | **0.041(0.001)** | 0.041(0.001) | 0.042(0.001) | 0.041(0.002) | 0.041(0.001) | 0.062(<0.000) |
| | 2 | **0.032(0.001)** | 0.032(0.002) | 0.032(0.002) | 0.032(0.002) | 0.043(0.010) | 0.062(<0.000) |
| 100 | 3 | **0.033(0.002)** | 0.034(0.002) | 0.033(0.002) | 0.036(0.008) | 0.041(0.008) | 0.063(<0.000) |
| | 4 | **0.038(0.001)** | 0.040(0.001) | 0.038(0.002) | 0.040(0.003) | 0.055(0.002) | 0.063(<0.000) |
| | 5 | **0.043(0.002)** | 0.044(0.002) | 0.043(0.003) | 0.046(0.003) | 0.053(0.007) | 0.063(0.001) |
| | 1 | **0.041(<0.000)** | 0.041(<0.000) | 0.042(<0.000) | 0.048(0.020) | 0.041(<0.000) | 0.063(<0.000) |
| | 2 | 0.031(0.001) | 0.031(0.001) | **0.030(0.001)** | 0.032(0.005) | 0.039(0.010) | 0.063(<0.000) |
| 200 | 3 | 0.033(0.001) | 0.033(0.001) | **0.032(0.001)** | 0.033(0.001) | 0.044(0.011) | 0.063(<0.000) |
| | 4 | **0.038(0.002)** | 0.039(0.002) | 0.038(0.002) | 0.039(0.002) | 0.054(0.011) | 0.063(<0.000) |
| | 5 | **0.041(0.001)** | 0.041(0.002) | 0.042(0.003) | 0.044(0.003) | 0.059(0.007) | 0.063(<0.000) |
| | 1 | 0.041(0.001) | 0.041(0.001) | **0.041(<0.000)** | 0.041(0.001) | 0.041(0.001) | 0.063(<0.000) |
| | 2 | 0.030(0.001) | 0.031(0.001) | **0.029(0.001)** | 0.030(0.001) | 0.036(0.001) | 0.063(<0.000) |
| 500 | 3 | 0.032(0.001) | 0.033(0.001) | **0.032(0.001)** | 0.032(0.001) | 0.038(0.009) | 0.063(<0.000) |
| | 4 | **0.037(0.001)** | 0.038(0.001) | 0.038(0.001) | 0.039(0.001) | 0.058(0.011) | 0.063(<0.000) |
| | 5 | **0.040(0.001)** | 0.040(0.001) | 0.043(0.003) | 0.045(0.002) | 0.062(0.006) | 0.063(<0.000) |
| | 1 | **0.041(<0.000)** | 0.041(<0.000) | 0.041(<0.000) | 0.041(<0.000) | 0.041(<0.000) | 0.063(<0.000) |
| | 2 | 0.030(0.001) | 0.030(0.001) | **0.029(<0.000)** | 0.030(<0.000) | 0.035(0.001) | 0.063(<0.000) |
| 1,000 | 3 | **0.031(<0.000)** | 0.032(<0.000) | 0.031(<0.000) | 0.031(<0.000) | 0.037(0.010) | 0.063(<0.000) |
| | 4 | **0.037(0.001)** | 0.038(0.001) | 0.038(0.001) | 0.039(0.001) | 0.060(0.012) | 0.063(<0.000) |
| | 5 | 0.040(<0.000) | **0.039(0.001)** | 0.043(0.002) | 0.044(0.002) | 0.058(0.013) | 0.063(<0.000) |

Table B.3: Comparison of MSE of parameter estimation for different methods in low dimensional categorical simulation. The simulated ten data sets are contaminated by setting 10% of the samples to outliers. The MSE's are calculated with the same methods in Table B.1.

| $n$ | $k$ | $Q^{(2)}(\mathbf{\Phi})$ 1st stage | $Q^{(2)}(\mathbf{\Phi})$ 2nd stage | $Q^{(3)}(\mathbf{\Phi})$ 1st stage | $Q^{(3)}(\mathbf{\Phi})$ 2nd stage | SFM | LDA |
|---|---|---|---|---|---|---|---|
| | 1 | 0.044(0.002) | **0.044(0.001)** | 0.044(0.002) | 0.057(0.017) | 0.044(0.002) | 0.064(0.001) |
| | 2 | **0.041(0.003)** | 0.044(0.006) | 0.045(0.006) | 0.060(0.025) | 0.054(0.008) | 0.064(0.001) |
| 50 | 3 | **0.045(0.003)** | 0.046(0.003) | 0.052(0.006) | 0.068(0.031) | 0.064(0.007) | 0.067(0.002) |
| | 4 | **0.048(0.003)** | 0.051(0.003) | 0.056(0.003) | 0.061(0.008) | 0.067(0.002) | 0.069(0.002) |
| | 5 | **0.053(0.002)** | 0.059(0.006) | 0.057(0.003) | 0.062(0.004) | 0.067(0.003) | 0.070(0.003) |
| | 1 | **0.042(0.002)** | 0.043(0.003) | 0.042(0.002) | 0.044(0.006) | 0.043(0.001) | 0.064(0.001) |
| | 2 | **0.040(0.002)** | 0.040(0.002) | 0.050(0.005) | 0.047(0.005) | 0.049(0.011) | 0.065(0.001) |
| 100 | 3 | **0.044(0.002)** | 0.044(0.002) | 0.054(0.003) | 0.054(0.004) | 0.069(0.009) | 0.065(0.001) |
| | 4 | **0.047(0.002)** | 0.049(0.002) | 0.056(0.002) | 0.058(0.003) | 0.073(0.003) | 0.066(0.003) |
| | 5 | **0.051(0.003)** | 0.054(0.003) | 0.055(0.004) | 0.059(0.004) | 0.073(0.002) | 0.068(0.003) |
| | 1 | 0.043(0.001) | 0.043(0.001) | **0.042(0.001)** | 0.043(<0.000) | 0.043(0.001) | 0.064(<0.000) |
| | 2 | **0.039(0.001)** | 0.039(0.001) | 0.050(0.002) | 0.044(0.001) | 0.058(0.022) | 0.064(<0.000) |
| 200 | 3 | **0.041(0.002)** | 0.042(0.001) | 0.052(0.001) | 0.052(0.002) | 0.066(0.016) | 0.065(<0.000) |
| | 4 | **0.047(0.002)** | 0.048(0.002) | 0.056(0.002) | 0.057(0.002) | 0.076(0.001) | 0.065(<0.000) |
| | 5 | **0.050(0.002)** | 0.053(0.002) | 0.054(0.002) | 0.058(0.002) | 0.077(0.002) | 0.065(0.001) |
| | 1 | **0.042(0.001)** | 0.042(0.001) | 0.042(0.001) | 0.042(0.001) | 0.043(0.001) | 0.064(<0.000) |
| | 2 | **0.039(0.001)** | 0.039(0.001) | 0.050(0.001) | 0.044(0.001) | 0.057(0.020) | 0.064(<0.000) |
| 500 | 3 | **0.040(0.001)** | 0.040(0.001) | 0.051(0.001) | 0.051(0.001) | 0.072(0.014) | 0.064(<0.000) |
| | 4 | **0.046(0.001)** | 0.048(0.001) | 0.056(0.001) | 0.057(0.001) | 0.078(0.001) | 0.064(<0.000) |
| | 5 | **0.050(0.002)** | 0.053(0.001) | 0.054(0.001) | 0.058(0.001) | 0.079(0.001) | 0.065(<0.000) |
| | 1 | **0.042(<0.000)** | 0.042(<0.000) | 0.042(<0.000) | 0.042(<0.000) | 0.043(<0.000) | 0.064(<0.000) |
| | 2 | 0.040(<0.000) | **0.039(<0.000)** | 0.050(0.001) | 0.045(0.001) | 0.047(0.014) | 0.064(<0.000) |
| 1,000 | 3 | **0.039(<0.000)** | 0.040(<0.000) | 0.051(<0.000) | 0.051(0.001) | 0.076(0.015) | 0.064(<0.000) |
| | 4 | **0.047(<0.000)** | 0.048(<0.000) | 0.056(0.001) | 0.057(0.001) | 0.079(0.003) | 0.064(<0.000) |
| | 5 | **0.050(0.001)** | 0.052(0.001) | 0.054(0.001) | 0.058(0.001) | 0.077(0.011) | 0.064(<0.000) |

Table B.4: Comparison of mean squared error (MSE) of estimated genotype distribution and running time in seconds in categorical simulation to infer population structure under Hardy-Weinberg equilibrium (HWE). All methods are run on a Intel(R) Core i7-3770 CPU at 3.40GHz machine. For all the methods, the MSE's are calculated using parameter estimates on the ten simulated data sets. For SFM and LDA posterior means are calculated from 100 thinned posterior draws from their MCMC algorithms. For MELD, LFA and ADMIXTURE their averaged running times are calculated on the ten simulated data sets. For MELD the running times are for the first stage estimation. Its averaged number of iterations is showed in parentheses of time column. The second stage estimation requires 1-2 additional iterations starting from the estimated parameter in the first stage. For SFM and LDA their running times are calculated based on $10,000$ iterations of their MCMC algorithms.

| Methods | | MELD $Q^{(2)}(\mathbf{\Phi})$ | | | SFM | | LDA | | LFA | | ADM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage MSE | 2nd stage MSE | time | MSE | time | MSE | time | MSE | time | MSE | time |
| 50 | 1 | **0.043(0.001)** | 0.043(0.001) | 0.45(1) | 0.043(0.001) | 9.98 | 0.050(0.001) | 0.09 | 0.047(0.001) | 0.22 | 0.083(0.002) | 0.01 |
| | 2 | 0.037(0.001) | 0.037(0.002) | 6.91(10) | 0.048(0.002) | 30.2 | 0.050(0.001) | 0.14 | **0.035(0.002)** | 0.26 | 0.116(0.005) | 0.01 |
| | 3 | 0.044(0.002) | 0.043(0.002) | 13.7(15) | 0.054(0.001) | 54.6 | 0.050(0.001) | 0.18 | **0.027(0.002)** | 0.27 | 0.138(0.006) | 0.01 |
| | 4 | 0.054(0.001) | 0.055(0.001) | 20.6(17) | 0.060(0.002) | 73.4 | 0.050(0.001) | 0.23 | **0.023(0.002)** | 0.29 | 0.155(0.007) | 0.01 |
| | 5 | 0.064(0.002) | 0.067(0.002) | 28.2(18) | 0.060(0.002) | 83.3 | 0.050(0.001) | 0.27 | **0.026(0.002)** | 0.36 | 0.159(0.007) | 0.02 |
| 100 | 1 | **0.041(<0.001)** | 0.041(<0.001) | 0.34(1) | 0.041(<0.001) | 10.5 | 0.050(<0.001) | 0.18 | 0.047(<0.001) | 0.28 | 0.083(0.001) | 0.01 |
| | 2 | **0.034(0.001)** | 0.034(0.001) | 7.18(11) | 0.044(0.002) | 35.7 | 0.050(<0.001) | 0.27 | 0.034(0.001) | 0.35 | 0.112(0.003) | 0.01 |
| | 3 | 0.038(0.001) | 0.038(0.001) | 13.1(14) | 0.049(0.002) | 62.5 | 0.050(<0.001) | 0.36 | **0.025(0.001)** | 0.41 | 0.134(0.003) | 0.01 |
| | 4 | 0.046(0.001) | 0.046(0.001) | 18.6(15) | 0.055(0.002) | 86.1 | 0.050(<0.001) | 0.45 | **0.019(0.002)** | 0.41 | 0.151(0.003) | 0.01 |
| | 5 | 0.053(0.001) | 0.056(0.001) | 25.1(16) | 0.058(0.004) | 100.6 | 0.050(<0.001) | 0.53 | **0.021(0.001)** | 0.47 | 0.153(0.003) | 0.03 |
| 200 | 1 | **0.040(<0.001)** | 0.040(<0.001) | 0.34(1) | 0.040(0.001) | 15.8 | 0.050(<0.001) | 0.35 | 0.047(0.001) | 0.39 | 0.082(0.001) | 0.01 |
| | 2 | **0.033(0.001)** | 0.033(<0.001) | 6.28(10) | 0.038(0.001) | 45.9 | 0.050(<0.001) | 0.54 | 0.033(0.001) | 0.47 | 0.111(0.003) | 0.01 |
| | 3 | 0.035(0.002) | 0.035(0.001) | 11.8(13) | 0.041(0.002) | 70.2 | 0.050(<0.001) | 0.72 | **0.024(0.001)** | 0.50 | 0.132(0.003) | 0.01 |
| | 4 | 0.042(0.001) | 0.041(0.001) | 17.05(14) | 0.046(0.002) | 105.2 | 0.050(<0.001) | 0.90 | **0.017(0.001)** | 0.58 | 0.150(0.002) | 0.01 |
| | 5 | 0.047(0.001) | 0.048(0.001) | 21.1(14) | 0.050(0.006) | 129.5 | 0.050(<0.001) | 1.06 | **0.019(0.001)** | 0.72 | 0.151(0.002) | 0.03 |
| 500 | 1 | **0.040(<0.001)** | 0.040(<0.001) | 0.34(1) | 0.040(<0.001) | 33.7 | 0.050(<0.001) | 0.91 | 0.047(<0.001) | 0.67 | 0.082(0.001) | 0.02 |
| | 2 | **0.032(0.001)** | 0.032(0.001) | 5.32(8) | 0.035(0.001) | 74.8 | 0.050(<0.001) | 1.37 | 0.033(0.001) | 0.77 | 0.108(0.002) | 0.02 |
| | 3 | 0.034(0.001) | 0.034(0.001) | 12.1(13) | 0.036(0.001) | 126.8 | 0.050(<0.001) | 1.82 | **0.024(0.001)** | 1.00 | 0.129(0.002) | 0.02 |
| | 4 | 0.039(0.001) | 0.037(0.001) | 17.1(14) | 0.039(0.001) | 171.5 | 0.050(<0.001) | 2.24 | **0.016(0.001)** | 1.14 | 0.147(0.002) | 0.02 |
| | 5 | 0.043(0.001) | 0.043(0.001) | 21.0(14) | 0.040(0.002) | 199.4 | 0.050(<0.001) | 2.66 | **0.017(0.001)** | 1.19 | 0.148(0.002) | 0.05 |
| 1,000 | 1 | **0.040(<0.001)** | 0.040(<0.001) | 0.34(1) | 0.040(<0.001) | 62.3 | 0.050(<0.001) | 1.84 | 0.047(<0.001) | 1.33 | 0.082(0.001) | 0.02 |
| | 2 | **0.031(<0.001)** | 0.032(0.001) | 1.81(3) | 0.034(0.001) | 140.4 | 0.050(<0.001) | 2.77 | 0.033(<0.001) | 1.43 | 0.108(0.002) | 0.02 |
| | 3 | 0.033(0.002) | 0.033(0.002) | 11.37(12) | 0.034(0.001) | 208.6 | 0.050(<0.001) | 3.66 | **0.023(<0.001)** | 1.75 | 0.130(0.002) | 0.02 |
| | 4 | 0.038(0.001) | 0.036(<0.001) | 16.77(14) | 0.037(<0.001) | 270.8 | 0.050(<0.001) | 4.51 | **0.015(<0.001)** | 1.61 | 0.148(0.001) | 0.02 |
| | 5 | 0.042(0.001) | 0.043(0.001) | 19.96(13) | 0.040(0.002) | 325.2 | 0.050(<0.001) | 5.35 | **0.016(<0.001)** | 1.97 | 0.148(0.001) | 0.07 |

Table B.5: Comparison of mean squared error (MSE) of estimated genotype distribution and running time in seconds in categorical simulation to infer population structure under non-HWE. All methods are run on a Intel(R) Core i7-3770 CPU at 3.40GHz machine. For all the methods, the MSE's are calculated using parameter estimates on the ten simulated data sets. For SFM and LDA posterior means are calculated from 100 thinned posterior draws from their MCMC algorithms. For MELD, LFA and ADMIXTURE their averaged running times are calculated on the ten simulated data sets. For MELD the running times are for the first stage estimation. Its averaged number of iterations is showed in parentheses of time column. The second stage estimation requires 1-2 additional iterations starting from the estimated parameter in the first stage. For SFM and LDA their running times are calculated based on $10,000$ iterations of their MCMC algorithms.

| Methods | | MELD $Q^{(2)}(\mathbf{\Phi})$ | | | SFM | | LDA | | LFA | | ADM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage MSE | 2nd stage MSE | time | MSE | time | MSE | time | MSE | time | MSE | time |
| | 1 | **0.067(0.001)** | 0.067(0.001) | 0.45(1) | 0.067(0.001) | 9.98 | 0.088(0.001) | 0.09 | 0.088(0.001) | 0.22 | 0.131(0.002) | 0.01 |
| | 2 | **0.053(0.002)** | 0.054(0.004) | 6.91(10) | 0.056(0.003) | 30.2 | 0.088(0.001) | 0.14 | 0.071(0.001) | 0.26 | 0.160(0.003) | 0.01 |
| 50 | 3 | 0.052(0.002) | **0.052(0.001)** | 13.7(15) | 0.054(0.002) | 54.6 | 0.088(0.001) | 0.18 | 0.058(0.003) | 0.27 | 0.182(0.004) | 0.01 |
| | 4 | 0.059(0.001) | 0.060(0.002) | 20.6(17) | 0.056(0.001) | 73.4 | 0.088(0.001) | 0.23 | **0.050(0.002)** | 0.29 | 0.198(0.003) | 0.01 |
| | 5 | 0.063(0.001) | 0.067(0.001) | 28.2(18) | 0.057(0.002) | 83.3 | 0.088(0.001) | 0.27 | **0.053(0.002)** | 0.36 | 0.200(0.003) | 0.02 |
| | 1 | 0.067(0.001) | **0.066(0.001)** | 0.34(1) | 0.066(0.001) | 10.5 | 0.088(<0.001) | 0.18 | 0.088(0.001) | 0.28 | 0.129(0.002) | 0.01 |
| | 2 | **0.051(0.001)** | 0.051(0.001) | 7.18(11) | 0.055(0.002) | 35.7 | 0.088(<0.001) | 0.27 | 0.071(0.001) | 0.35 | 0.155(0.002) | 0.01 |
| 100 | 3 | 0.050(0.002) | **0.050(0.001)** | 13.1(14) | 0.053(0.002) | 62.5 | 0.088(<0.001) | 0.36 | 0.058(0.001) | 0.41 | 0.176(0.002) | 0.01 |
| | 4 | 0.055(0.001) | 0.057(0.001) | 18.6(15) | 0.056(0.001) | 86.1 | 0.088(<0.001) | 0.45 | **0.048(0.001)** | 0.41 | 0.193(0.002) | 0.01 |
| | 5 | 0.060(0.001) | 0.063(0.001) | 25.1(16) | 0.056(0.002) | 100.6 | 0.088(<0.001) | 0.53 | **0.050(0.001)** | 0.47 | 0.194(0.002) | 0.03 |
| | 1 | **0.065(<0.001)** | 0.065(<0.001) | 0.34(1) | 0.066(<0.001) | 15.8 | 0.088(<0.001) | 0.35 | 0.088(<0.001) | 0.39 | 0.129(0.001) | 0.01 |
| | 2 | 0.051(0.001) | **0.050(0.001)** | 6.28(10) | 0.055(0.002) | 45.9 | 0.088(<0.001) | 0.54 | 0.071(0.001) | 0.47 | 0.154(0.001) | 0.01 |
| 200 | 3 | 0.050(0.002) | **0.048(0.001)** | 11.8(13) | 0.052(0.001) | 70.2 | 0.088(<0.001) | 0.72 | 0.057(0.001) | 0.50 | 0.175(0.002) | 0.01 |
| | 4 | 0.053(0.001) | 0.054(0.001) | 17.05(14) | 0.055(0.001) | 105.2 | 0.088(<0.001) | 0.90 | **0.047(0.001)** | 0.58 | 0.192(0.001) | 0.01 |
| | 5 | 0.059(0.001) | 0.060(0.001) | 21.1(14) | 0.055(0.002) | 129.5 | 0.088(<0.001) | 1.06 | **0.048(0.001)** | 0.72 | 0.193(0.001) | 0.03 |
| | 1 | **0.065(<0.001)** | 0.065(<0.001) | 0.34(1) | 0.065(<0.001) | 33.7 | 0.088(<0.001) | 0.91 | 0.088(<0.001) | 0.67 | 0.128(<0.001) | 0.02 |
| | 2 | **0.050(0.001)** | 0.050(0.001) | 5.32(8) | 0.053(0.001) | 74.8 | 0.088(<0.001) | 1.37 | 0.072(0.001) | 0.77 | 0.152(0.001) | 0.02 |
| 500 | 3 | **0.047(0.001)** | 0.047(0.001) | 12.1(13) | 0.051(0.001) | 126.8 | 0.088(<0.001) | 1.82 | 0.057(0.001) | 1.00 | 0.172(0.001) | 0.02 |
| | 4 | 0.053(0.001) | 0.053(0.001) | 17.1(14) | 0.054(0.001) | 171.5 | 0.088(<0.001) | 2.24 | **0.046(0.001)** | 1.14 | 0.190(0.001) | 0.02 |
| | 5 | 0.057(0.001) | 0.058(0.001) | 21.0(14) | 0.055(0.001) | 199.4 | 0.088(<0.001) | 2.66 | **0.047(0.001)** | 1.19 | 0.190(0.001) | 0.05 |
| | 1 | **0.065(<0.001)** | 0.065(<0.001) | 0.34(1) | 0.065(<0.001) | 62.3 | 0.088(<0.001) | 1.84 | 0.088(<0.001) | 1.33 | 0.128(<0.001) | 0.02 |
| | 2 | **0.050(0.001)** | 0.050(0.001) | 1.81(3) | 0.053(0.001) | 140.4 | 0.088(<0.001) | 2.77 | 0.072(<0.001) | 1.43 | 0.151(0.001) | 0.02 |
| 1,000 | 3 | 0.048(0.001) | **0.047(0.001)** | 11.37(12) | 0.050(0.001) | 208.6 | 0.088(<0.001) | 3.66 | 0.057(0.001) | 1.75 | 0.172(0.001) | 0.02 |
| | 4 | 0.052(<0.001) | 0.052(<0.001) | 16.77(14) | 0.053(<0.001) | 270.8 | 0.088(<0.001) | 4.51 | **0.046(0.001)** | 1.61 | 0.189(0.001) | 0.02 |
| | 5 | 0.057(<0.001) | 0.058(0.001) | 19.96(13) | 0.054(0.001) | 325.2 | 0.088(<0.001) | 5.35 | **0.047(0.001)** | 1.97 | 0.190(0.001) | 0.07 |

Table B.6: Quantitative trait association simulation with 50 nucleotides and one Gaussian trait. For MELD the averaged Kullback-Leibler (KL) distance between estimated component distributions and marginal frequency for each nucleotide are calculated. The first eight nucleotides with largest averaged KL distance are selected. For the Bayesian copula factor model, partial correlation coefficients are calculated. Nucleotides with 95% credible interval of the partial correlation excluding zero are selected. Nucleotides not in $J = \{2, 4, 12, 14, 32, 34, 42, 44\}$ are labeled by an underline and missing nucleotides are crossed out.

| Contamination | Data set | MELD $Q^{(2)}(\mathbf{\Phi})$ 1st stage | Bayesian copula factor model |
|---|---|---|---|
| 0% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, \underline{33}, 34, \underline{35}, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{18}, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{22}, \underline{27}, \cancel{32}, 34, \underline{35}, 42, 44, \underline{45}\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{31}, \cancel{32}, 34, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{31}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, 12, 14, \underline{20}, \cancel{32}, 34, \underline{40}, 42, 44, \underline{46}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{26}, \cancel{32}, 34, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44, \underline{45}\}$ |
| 4% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, \underline{33}, 34, \underline{35}, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{18}, \underline{27}, \cancel{32}, 34, 42, 44, \underline{45}\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{22}, \underline{27}, \cancel{32}, 34, \underline{35}, \underline{41}, 42, 44, \underline{45}\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, 32, \underline{33}, 34, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{31}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{5}, \underline{7}, 12, 14, \underline{18}, \underline{26}, \cancel{32}, 34, 42, 44, \underline{46}\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{25}, \underline{26}, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{19}, \underline{21}, \cancel{32}, 34, \underline{41}, 42, \underline{43}, 44, \underline{46}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{26}, \cancel{32}, 34, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, 12, 14, \cancel{32}, 34, 42, 44, \underline{45}\}$ |
| 10% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, 12, 14, \cancel{32}, 34, \underline{35}, \underline{39}, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, 27, \cancel{32}, 34, 42, 44, \underline{49}, \underline{50}\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \underline{22}, \underline{26}, \cancel{32}, 34, \underline{35}, 42, 44, \underline{45}\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{17}, 32, \underline{33}, 34, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{31}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \underline{26}, \cancel{32}, 34, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{26}, \cancel{32}, 34, \underline{36}, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, 12, 14, \underline{22}, 32, 34, 42, 44, \underline{45}, \underline{50}\}$ |
| 20% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{15}, \underline{16}, \cancel{32}, \underline{33}, 34, \underline{35}, \underline{40}, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{9}, \underline{11}, 12, \cancel{14}, \underline{20}, \cancel{32}, 34, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{6}, 12, 14, \underline{16}, \underline{20}, \underline{22}, \underline{27}, \cancel{32}, 34, \underline{35}, \underline{41}, 42, 44, \underline{45}, \underline{50}\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \underline{13}, 14, \underline{30}, \underline{31}, 32, \underline{33}, 34, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{15}, \underline{18}, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{5}, \underline{7}, 12, 14, \underline{18}, \cancel{32}, 34, \underline{38}, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{15}, \underline{16}, \cancel{32}, \underline{33}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{\underline{1}, 2, 4, 12, 14, \underline{19}, \underline{20}, \underline{21}, \underline{25}, \cancel{32}, \underline{33}, 34, \underline{37}, 42, \underline{43}, 44\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{18}, \underline{31}, \cancel{32}, 34, \underline{36}, \underline{39}, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, 12, 14, \underline{19}, \cancel{32}, 34, 42, 44, \underline{45}\}$ |

Table B.7: Quantitative trait association simulation with 50 nucleotides and one Poisson trait. For MELD the averaged Kullback-Leibler (KL) distance between estimated component distributions and marginal frequency for each nucleotide are calculated. The first eight nucleotides with largest averaged KL distance are selected. For the Bayesian copula factor model, partial correlation coefficients are calculated. Nucleotides with 95% posterior interval of the partial correlation excluding zero are selected. Nucleotides not in $J = \{2, 4, 12, 14, 32, 34, 42, 44\}$ are labeled by an underline and missing nucleotides are crossed out.

| Contamination | Data set | MELD $Q^{(2)}(\mathbf{\Phi})$ 1st stage | Bayesian copula factor model |
|---|---|---|---|
| 0% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{15}, \underline{26}, \underline{29}, \cancel{32}, 34, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \underline{20}, \underline{28}, \cancel{32}, 34, \underline{37}, 42, 44, \underline{45}\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{20}, \cancel{32}, 34, \underline{37}, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{10}, 12, 14, \underline{18}, \cancel{32}, \underline{33}, 3442, 44, \underline{46}, \underline{49}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, \underline{10}, 12, 14, \underline{21}, \underline{25}, \cancel{32}, 34, \underline{39}, \underline{41}, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| 4% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \underline{15}, \underline{26}, \underline{29}, \cancel{32}, \underline{33}, 34, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \cancel{32}, 34, \underline{37}, 42, 44\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{20}, \cancel{32}, 34, \underline{37}, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{19}, \cancel{32}, 34, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{10}, 12, \underline{13}, 14, \underline{18}, \cancel{32}, \underline{33}, 34, 42, 44, \underline{46}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{\underline{1}, 2, \underline{3}, 4, \underline{6}, \underline{10}, 12, 14, \underline{21}, \underline{25}, \cancel{32}, 34, \underline{39}, \underline{41}, 42, 44, \underline{45}\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{23}, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| 10% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \underline{15}, \underline{26}, \cancel{32}, 34, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \underline{16}, 32, 34, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{20}, \cancel{32}, 34, \underline{37}, 42, 44\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, \underline{33}, 34, \underline{37}, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{10}, 12, 14, \underline{16}, \underline{21}, \cancel{32}, 34, 42, 44\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{6}, 12, 14, \underline{22}, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, 14, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{10}, 12, 14, \underline{18}, \cancel{32}, 34, 42, 44, \underline{46}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, 4, \underline{6}, \underline{10}, 12, 14, \underline{17}, \underline{21}, \cancel{32}, 34, 42, 44, \underline{48}\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{23}, \cancel{32}, 34, 42, 44, \underline{49}\}$ |
| 20% | 1 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{15}, \underline{29}, \cancel{32}, 34, 42, 44\}$ |
| | 2 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{7}, 12, \cancel{14}, 32, 34, \underline{35}, 42, 44\}$ |
| | 3 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, \cancel{14}, \cancel{32}, 34, \underline{37}, 42, 44\}$ |
| | 4 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{30}, \cancel{32}, \underline{33}, 34, \underline{37}, 42, 44\}$ |
| | 5 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{16}, \cancel{32}, 34, \underline{40}, 42, 44, \underline{47}\}$ |
| | 6 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \cancel{32}, 34, \underline{36}, 42, 44\}$ |
| | 7 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, 12, 14, \underline{17}, \underline{27}, \cancel{32}, 34, 42, 44\}$ |
| | 8 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{10}, 12, 14, \underline{18}, \underline{23}, \cancel{32}, 34, \underline{40}, 42, 44, \underline{46}, \underline{49}\}$ |
| | 9 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, \underline{3}, \underline{4}, \underline{10}, 12, 14, \underline{21}, \cancel{32}, 34, \underline{41}, 42, 44\}$ |
| | 10 | $\{2, 4, 12, 14, 32, 34, 42, 44\}$ | $\{2, 4, \underline{6}, 12, 14, \underline{23}, \cancel{32}, 34, 42, 44, \underline{49}\}$ |

Table B.8: Mean squared error (MSE) of parameter estimation and running time in seconds in simulation with categorical, Gaussian, Poisson mixed variables. MELD is run on a Intel(R) Core i7-3770 CPU at 3.40GHz machine. Standard deviations of MSE's calculated on ten simulated data sets are provided in parentheses of MSE column. The averaged number of iterations is showed in parentheses of time column. For non-categorical data squared Euclidean distance is used to recover membership variable.

| MELD $Q^{(2)}(\mathbf{\Phi})$ | | Categorical | | Normal | | Poisson | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | $k$ | 1st stage MSE | 2nd stage MSE | 1st stage MSE | 2nd stage MSE | 1st stage MSE | 2nd stage MSE | time |
| 50 | 1 | 0.006(<0.001) | 0.006(0.001) | 9.57(0.32) | 8.94(0.16) | 6.42(0.20) | 7.23(2.08) | 0.95(3) |
| | 2 | 0.011(0.001) | 0.010(0.001) | 0.33(0.21) | 0.30(0.20) | 4.73(0.40) | 4.76(0.47) | 17.4(28) |
| | 3 | 0.020(0.001) | 0.018(0.001) | 1.63(1.01) | 1.67(1.00) | 6.41(0.74) | 6.44(0.75) | 33.2(36) |
| | 4 | 0.029(0.002) | 0.027(0.001) | 2.81(1.07) | 2.99(1.07) | 8.27(1.23) | 8.20(1.33) | 42.4(35) |
| 100 | 1 | 0.004(<0.001) | 0.004(0.001) | 9.53(0.20) | 9.51(1.77) | 6.36(0.13) | 7.16(2.78) | 0.91(3) |
| | 2 | 0.006(<0.001) | 0.006(<0.001) | 0.25(0.15) | 0.25(0.15) | 4.61(0.46) | 4.61(0.44) | 16.5(26) |
| | 3 | 0.013(0.001) | 0.011(0.001) | 1.97(0.88) | 1.95(0.88) | 5.28(0.56) | 5.23(0.52) | 31.5(34) |
| | 4 | 0.020(0.002) | 0.017(0.001) | 2.49(0.87) | 2.57(0.91) | 6.59(0.72) | 6.52(0.72) | 37.1(30) |
| 200 | 1 | 0.003(<0.001) | 0.003(<0.001) | 9.53(0.11) | 8.94(0.10) | 6.42(0.13) | 6.26(0.06) | 0.81(2) |
| | 2 | 0.004(<0.001) | 0.004(<0.001) | 0.20(0.13) | 0.21(0.14) | 4.60(0.39) | 4.57(0.41) | 16.7(27) |
| | 3 | 0.008(0.001) | 0.007(0.001) | 1.80(0.62) | 1.81(0.65) | 5.14(0.46) | 5.13(0.49) | 25.7(28) |
| | 4 | 0.014(0.002) | 0.011(0.001) | 2.93(0.38) | 2.99(0.38) | 6.32(0.82) | 6.30(0.82) | 30.6(25) |
| 500 | 1 | 0.002(<0.001) | 0.002(<0.001) | 9.66(0.05) | 9.01(0.01) | 6.39(0.05) | 6.26(0.02) | 0.92(3) |
| | 2 | 0.002(<0.001) | 0.002(<0.001) | 0.06(0.05) | 0.07(0.05) | 4.45(0.27) | 4.45(0.28) | 17.1(27) |
| | 3 | 0.004(<0.001) | 0.004(<0.001) | 1.83(0.36) | 1.87(0.37) | 5.52(0.31) | 5.51(0.25) | 25.6(28) |
| | 4 | 0.007(0.001) | 0.006(0.001) | 3.30(0.23) | 3.35(0.24) | 5.97(0.43) | 6.03(0.46) | 25.6(21) |
| 1,000 | 1 | 0.002(<0.001) | 0.002(<0.001) | 9.66(0.05) | 9.02(0.01) | 6.36(0.01) | 6.26(0.00) | 0.97(3) |
| | 2 | 0.002(<0.001) | 0.002(<0.001) | 0.06(0.04) | 0.06(0.04) | 4.44(0.12) | 4.45(0.12) | 15.0(24) |
| | 3 | 0.003(<0.001) | 0.003(<0.001) | 1.80(0.19) | 1.87(0.19) | 5.51(0.19) | 5.53(0.18) | 24.4(27) |
| | 4 | 0.005(<0.001) | 0.004(<0.001) | 3.32(0.17) | 3.34(0.17) | 5.46(0.19) | 5.47(0.19) | 24.4(20) |

# Bibliography


Absil, P.-A., Mahony, R., and Sepulchre, R. (2009), *Optimization Algorithms on Matrix Manifolds*, Princeton University Press.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008), "Mixed Membership Stochastic Blockmodels," *the Journal of Machine Learning Research*, 9, 1981–2014.

Airoldi, E. M., Blei, D., Erosheva, E. A., and Fienberg, S. E. (2014), *Handbook of Mixed Membership Models and Their Applications*, CRC Press.

Alexander, D. H., Novembre, J., and Lange, K. (2009), "Fast Model-Based Estimation of Ancestry in Unrelated Individuals," *Genome Research*, 19, 1655–1664.

Amari, S.-I. (1998), "Natural Gradient Works Efficiently in Learning," *Neural computation*, 10, 251–276.

Anandkumar, A., Hsu, D., and Kakade, S. M. (2012a), "A Method of Moments for Mixture Models and Hidden Markov Models," *JMLR W&CP 23: COLT*.

Anandkumar, A., Liu, Y., Hsu, D. J., Foster, D. P., and Kakade, S. M. (2012b), "A Spectral Algorithm for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems 25*, pp. 917–925.

Anandkumar, A., Ge, R., Hsu, D., and Kakade, S. M. (2014a), "A Tensor Approach to Learning Mixed Membership Community Models," *the Journal of Machine Learning Research*, 15, 2239–2312.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014b), "Tensor Decompositions for Learning Latent Variable Models," *the Journal of Machine Learning Research*, 15, 2773–2832.

Anderson, J. C. and Gerbing, D. W. (1988), "Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach." *Psychological bulletin*, 103, 411.

Archambeau, C. and Bach, F. R. (2009), "Sparse Probabilistic Projections," in *Advances in Neural Information Processing Systems 21*, pp. 73–80.

Armagan, A., Clyde, M., and Dunson, D. B. (2011), "Generalized Beta Mixtures of Gaussians," in *Advances in Neural Information Processing Systems 24*, pp. 523–531.

Armagan, A., Dunson, D. B., and Lee, J. (2013), "Generalized Double Pareto Shrinkage," *Statistica Sinica*, 23, 119.

Arminger, G. and Küsters, U. (1988), "Latent Trait Models with Indicators of Mixed Measurement Level," in *Latent Trait and Latent Class Models*, pp. 51–73, Springer, New York.

Arora, S., Ge, R., and Moitra, A. (2012), "Learning Topic Models - Going beyond SVD," in *Fifty-Third IEEE Annual Symposium on Foundations of Computer Science*, pp. 1–10.

Bach, F. R. and Jordan, M. I. (2005), "A Probabilistic Interpretation of Canonical Correlation Analysis," *Technical Report 688, Department of Statistics, University of California, Berkeley*.

Bentler, P. M. (1983), "Some Contributions to Efficient Statistics in Structural Models: Specification and Estimation of Moment Structures," *Psychometrika*, 48, 493–517.

Bhattacharya, A. and Dunson, D. B. (2011), "Sparse Bayesian Infinite Factor Models," *Biometrika*, 98, 291–306.

Bhattacharya, A. and Dunson, D. B. (2012), "Simplex Factor Models for Multivariate Unordered Categorical Data," *Journal of the American Statistical Association*, 107, 362–377.

Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015), "Dirichlet-Laplace Priors for Optimal Shrinkage," *Journal of the American Statistical Association*, 110, 1479–1490.

Biernacki, C., Celeux, G., and Govaert, G. (2000), "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22, 719–725.

Bingham, C. (1974), "An Antipodally Symmetric Distribution on the Sphere," *The Annals of Statistics*, pp. 1201–1225.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003), "Latent Dirichlet Allocation," *the Journal of Machine Learning Research*, 3, 993–1022.

Bollen, K. A., Kolenikov, S., and Bauldry, S. (2014), "Model-Implied Instrumental Variable-Generalized Method of Moments (MIIV-GMM) Estimators for Latent Variable Models," *Psychometrika*, 79, 20–50.

Brown, C. D., Mangravite, L. M., and Engelhardt, B. E. (2013), "Integrative Modeling of eQTLs and Cis-Regulatory Elements Suggests Mechanisms Underlying Cell Type Specificity of eQTLs," *PLoS Genetics*, 9, e1003649.

Browne, M. W. (1973), "Generalized Least Squares Estimators in the Analysis of Covariance Structures," *ETS Research Bulletin Series*, 1973, i–36.

Browne, M. W. (1979), "The Maximum-Likelihood Solution in Inter-Battery Factor Analysis," *British Journal of Mathematical and Statistical Psychology*, 32, 75–86.

Browne, M. W. (1980), "Factor Analysis of Multiple Batteries by Maximum Likelihood," *British Journal of Mathematical and Statistical Psychology*, 33, 184–199.

Browning, S. R. and Browning, B. L. (2011), "Haplotype Phasing: Existing Methods and New Developments," *Nature Reviews. Genetics*, 12, 703–714.

Byrne, S. and Girolami, M. (2013), "Geodesic Monte Carlo on Embedded Manifolds," *Scandinavian Journal of Statistics*, 40, 825–845.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011), "Robust Principal Component Analysis?" *Journal of the ACM*, 58, 11.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2008), "High-Dimensional Sparse Factor Modeling: Applications in Gene Expression Genomics," *Journal of the American Statistical Association*, 103, 1438–1456.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The Horseshoe Estimator for Sparse Signals," *Biometrika*, 97, 465–480.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2009), "Sparse and Low-Rank Matrix Decompositions," in *47th Annual Allerton Conference on Communication, Control, and Computing*, pp. 962–967.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2011), "Rank-Sparsity Incoherence for Matrix Decomposition," *SIAM Journal on Optimization*, 21, 572–596.

Chang, J. (2012), "lda: Collapsed Gibbs Sampling Methods for Topic Models," .

Colombo, N. and Vlassis, N. (2015), "FastMotif: Spectral Sequence Motif Discovery," *Bioinformatics, to appear*.

Comon, P. (1994), "Independent Component Analysis, a New Concept?" *Signal Processing*, 36, 287–314.

Cover, T. M. and Thomas, J. A. (2006), *Elements of Information Theory*, Wiley.

Cunningham, J. P. and Ghahramani, Z. (2014), "Linear Dimensionality Reduction: Survey, Insights, and Generalizations," *the Journal of Machine Learning Research, to appear*.

Damianou, A., Ek, C., Titsias, M., and Lawrence, N. (2012), "Manifold Relevance Determination," in *29th International Conference on Machine Learning*, pp. 145–152.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B*, 39, 1–38.

Diaconis, P., Holmes, S., and Shahshahani, M. (2012), "Sampling From a Manifold," *arXiv:1206.6913 [math, stat]*.

Dunson, D. B. (2000), "Bayesian Latent Variable Models for Clustered Mixed Outcomes," *Journal of the Royal Statistical Society. Series B*, pp. 355–366.

Dunson, D. B. (2003), "Dynamic Latent Trait Models for Multidimensional Longitudinal Data," *Journal of the American Statistical Association*, 98, 555–563.

Dunson, D. B. and Xing, C. (2009), "Nonparametric Bayes Modeling of Multivariate Categorical Data," *Journal of the American Statistical Association*, 104, 1042–1051.

Edelman, A., Arias, T. A., and Smith, S. T. (1998), "The Geometry of Algorithms with Orthogonality Constraints," *SIAM Journal on Matrix Analysis and Applications*, 20, 303–353.

Edwards, D. (2000), *Introduction to Graphical Modelling*, Springer, New York, 2nd edn.

Efron, B. (1975), "Defining the Curvature of a Statistical Problem (With Applications to Second Order Efficiency)," *The Annals of Statistics*, pp. 1189–1242.

Federer, H. (1969), *Geometric Measure Theory*, Springer, New York.

Flury, B. N. (1984), "Common Principal Components in k Groups," *Journal of the American Statistical Association*, 79, 892–898.

Gallant, A. R., Giacomini, R., and Ragusa, G. (2013), *Generalized Method of Moments with Latent Variables*, Centre for Economic Policy Research.

Gao, C., Brown, C. D., and Engelhardt, B. E. (2013), "A Latent Factor Model with a Mixture of Sparse and Dense Factors to Model Gene Expression Data with Confounding Effects," *arXiv:1310.4792*.

George, E. I. and McCulloch, R. E. (1993), "Variable Selection via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881–889.

Girolami, M. and Calderhead, B. (2011), "Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods," *Journal of the Royal Statistical Society: Series B*, 73, 123–214.

González, I., Déjean, S., Martin, P. G., and Baccini, A. (2008), "CCA: an R package to Extend Canonical Correlation Analysis," *Journal of Statistical Software*, 23, 1–14.

Griffiths, T. L. and Ghahramani, Z. (2011), "The Indian Buffet Process: An Introduction and Review," *the Journal of Machine Learning Research*, 12, 1185–1224.

Gruhl, J., Erosheva, E. A., Crane, P. K., and others (2013), "A Semiparametric Approach to Mixed Outcome Latent Variable Models: Estimating the Association Between Cognition and Regional Brain Volumes," *The Annals of Applied Statistics*, 7, 2361–2383.

Hall, A. R. (2005), *Generalized Method of Moments*, Oxford University Press.

Hansen, L. P. (1982), "Large Sample Properties of Generalized Method of Moments Estimators," *Econometrica: Journal of the Econometric Society*, 50, 1029.

Hao, W., Song, M., and Storey, J. D. (2013), "Probabilistic Models of Genetic Variation in Structured Populations Applied to Global Human Studies," *arXiv:1312.2041 [q-bio, stat]*.

Harley, C. B. and Reynolds, R. P. (1987), "Analysis of E. coli Promoter Sequences." *Nucleic Acids Research*, 15, 2343–2361.

Hoff, P. D. (2007), "Extending the Rank Likelihood for Semiparametric Copula Estimation," *The Annals of Applied Statistics*, 1, 265–283.

Hoff, P. D. (2009a), "A Hierarchical Eigenmodel for Pooled Covariance Estimation," *Journal of the Royal Statistical Society. Series B*, 71, 971–992.

Hoff, P. D. (2009b), "Simulation of the Matrix Binghamvon MisesFisher Distribution, with Applications to Multivariate and Relational Data," *Journal of Computational and Graphical Statistics*, 18, 438–456.

Hofmann, T. (1999), "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, ACM.

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, 24, 417–441.

Hotelling, H. (1936), "Relations Between Two Sets of Variates," *Biometrika*, 28, 321–377.

Hsu, D. and Kakade, S. M. (2013), "Learning Mixtures of Spherical Gaussians: Moment Methods and Spectral Decompositions," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 11–20, ACM.

Hsu, D., Kakade, S. M., and Zhang, T. (2012), "A Spectral Algorithm for Learning Hidden Markov Models," *Journal of Computer and System Sciences*, 78, 1460–1480.

Huang, J., Zhang, T., and Metaxas, D. (2011), "Learning with Structured Sparsity," *the Journal of Machine Learning Research*, 12, 3371–3412.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2011), "Structured Variable Selection with Sparsity-Inducing Norms," *the Journal of Machine Learning Research*, 12, 2777–2824.

Jia, Y., Salzmann, M., and Darrell, T. (2010), "Factorized Latent Spaces with Structured Sparsity," in *Advances in Neural Information Processing Systems 23*, pp. 982–990.

Joachims, T. (1997), "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 143–151.

Jöreskog, K. G. and Sörbom, D. (1987), "New Developments in LISREL," *Paper presented at the National Symposium on Methodological Issues in Causal Modeling, University of Alabama, Tuscaloosa.*

Khan, S. A., Virtanen, S., Kallioniemi, O. P., Wennerberg, K., Poso, A., and Kaski, S. (2014), "Identification of Structural Features in Chemicals Associated with Cancer Drug Response: A Systematic Data-Driven Analysis," *Bioinformatics*, 30, i497–i504.

Khatri, C. G. and Mardia, K. V. (1977), "The von Mises-Fisher Matrix Distribution in Orientation Statistics," *Journal of the Royal Statistical Society.*, 39, 95–106.

Klami, A. (2014), "Polya-Gamma Augmentations for Factor Models," in *The 6th Asian Conference on Machine Learning*, pp. 112–128.

Klami, A., Virtanen, S., and Kaski, S. (2013), "Bayesian Canonical Correlation Analysis," *the Journal of Machine Learning Research*, 14, 965–1003.

Klami, A., Virtanen, S., Leppäaho, E., and Kaski, S. (2014a), "Group Factor Analysis," *IEEE Transactions on Neural Networks and Learning Systems*, Accepted for publication.

Klami, A., Bouchard, G., and Tripathi, A. (2014b), "Group-Sparse Embeddings in Collective Matrix Factorization," in *International Conference on Learning Representations*.

Knowles, D. and Ghahramani, Z. (2011), "Nonparametric Bayesian Sparse Factor Models with Application to Gene Expression Modeling," *The Annals of Applied Statistics*, 5, 1534–1552.

Koller, D. and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press.

Kowalski, M. and Torrésani, B. (2009), "Structured Sparsity: From Mixed Norms to Structured Shrinkage," in *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*.

Lawrence, N. (2005), "Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models," *the Journal of Machine Learning Research*, 6, 1783–1816.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010), "Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data," *Nature Reviews Genetics*, 11, 733–739.

Li, F. and Zhang, N. R. (2010), "Bayesian Variable Selection in Structured High-Dimensional Covariate Spaces With Applications in Genomics," *Journal of the American Statistical Association*, 105, 1202–1214.

Li, K.-C. (2002), "Genome-Wide Coexpression Dynamics: Theory and Application," *Proceedings of the National Academy of Sciences*, 99, 16875–16880.

Liang, K.-Y. and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, pp. 13–22.

Lichman, M. (2013), "UCI Machine Learning Repository," .

Liu, C., Rubin, D. B., and Wu, Y. N. (1998), "Parameter Expansion to Accelerate EM: The PX-EM Algorithm," *Biometrika*, 85, 755–770.

Liu, J. S. and Wu, Y. N. (1999), "Parameter Expansion for Data Augmentation," *Journal of the American Statistical Association*, 94, 1264–1274.

Lucas, J., Carvalho, C., Wang, Q., Bild, A., Nevins, J. R., and West, M. (2006), "Sparse Statistical Modelling in Gene Expression Genomics," *Bayesian Inference for Gene Expression and Proteomics*, 1.

Lucas, J. E., Kung, H.-N., and Chi, J.-T. A. (2010), "Latent Factor Analysis to Discover Pathway-Associated Putative Segmental Aneuploidies in Human Cancers," *PLoS Computational Biology*, 6, e1000920.

Luo, M. F., Boettcher, P. J., Schaeffer, L. R., and Dekkers, J. C. (2001), "Bayesian Inference for Categorical Traits with an Application to Variance Component Estimation," *Journal of Dairy Science*, 84, 694–704.

Ma, H., Schadt, E. E., Kaplan, L. M., and Zhao, H. (2011), "COSINE: Condition-Specific Sub-Network Identification Using a Global Optimization Method," *Bioinformatics*, 27, 1290–1298.

Mangravite, L. M., Engelhardt, B. E., Medina, M. W., Smith, J. D., Brown, C. D., Chasman, D. I., Mecham, B. H., Howie, B., Shim, H., Naidoo, D., et al. (2013), "A Statin-Dependent QTL for *GATM* Expression is Associated with Statin-Induced Myopathy," *Nature*, 502, 377–380.

McDonald, R. P. (1970), "Three Common Factor Models for Groups of Variables," *Psychometrika*, 35, 111–128.

Mitchell, T. J. and Beauchamp, J. J. (1988), "Bayesian Variable Selection in Linear Regression," *Journal of the American Statistical Association*, 83, 1023–1032.

Moustaki, I. and Knott, M. (2000), "Generalized Latent Trait Models," *Psychometrika*, 65, 391–411.

Murray, J. S., Dunson, D. B., Carin, L., and Lucas, J. E. (2013), "Bayesian Gaussian Copula Factor Models for Mixed Data," *Journal of the American Statistical Association*, 108, 656–665.

Muthén, B. (1983), "Latent Variable Structural Equation Modeling with Categorical Data," *Journal of Econometrics*, 22, 43–65.

Muthén, B. (1984), "A General Structural Equation Model with Dichotomous, Ordered Categorical, and Continuous Latent Variable Indicators," *Psychometrika*, 49, 115–132.

Neal, R. M. (2011), "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, eds. S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, Boca Raton: Chapman and Hall-CRC Press.

Newey, W. K. and West, K. D. (1987), "Hypothesis Testing with Efficient Method of Moments Estimation," *International Economic Review*, pp. 777–787.

Newman, M. E. J. (2012), "Communities, Modules and Large-scale Structure in Networks," *Nature Physics*, 8, 25–31.

Park, T. and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686.

Parkhomenko, E., Tritchler, D., and Beyene, J. (2009), "Sparse Canonical Correlation Analysis with Application to Genomic Data Integration," *Statistical Applications in Genetics and Molecular Biology*, 8, 1–34.

Pearson, K. (1894), "Contributions to the Mathematical Theory of Evolution," *Philosophical Transactions of the Royal Society A*, 185, 71–110.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011), "Scikit-learn: Machine Learning in Python," *the Journal of Machine Learning Research*, 12, 2825–2830.

Polson, N. G. and Scott, J. G. (2011), "Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction," in *Bayesian Statistics*, eds. J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, no. 9, pp. 501–538, Oxford University Press.

Pournara, I. and Wernisch, L. (2007), "Factor Analysis for Gene Regulatory Networks and Transcription Factor Activity Profiles," *BMC Bioinformatics*, 8, 61.

Pritchard, J. K., Stephens, M., Rosenberg, N. A., and Donnelly, P. (2000a), "Association Mapping in Structured Populations," *The American Journal of Human Genetics*, 67, 170–181.

Pritchard, J. K., Stephens, M., and Donnelly, P. (2000b), "Inference of Population Structure Using Multilocus Genotype Data," *Genetics*, 155, 945–959.

Qu, X. and Chen, X. (2011), "Sparse Structured Probabilistic Projections for Factorized Latent Spaces," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 1389–1394.

Quinn, K. M. (2004), "Bayesian Factor Analysis for Mixed Ordinal and Continuous Responses," *Political Analysis*, 12, 338–353.

Rao, V., Lin, L., and Dunson, D. (2014), "Data Augmentation for Models Based on Rejection Sampling," *arXiv:1406.6652 [stat]*.

Raskutti, G. and Mukherjee, S. (2015), "The Information Geometry of Mirror Descent," *IEEE Transactions on Information Theory*, 61, 1451–1457.

Ray, P., Zheng, L., Lucas, J., and Carin, L. (2014), "Bayesian Joint Analysis of Heterogeneous Genomics Data," *Bioinformatics*, 30, 1370–1376.

Ročková, V. and George, E. I. (2014), "EMVS: the EM Approach to Bayesian Variable Selection," *Journal of the American Statistical Association*, 109, 828–846.

Ročková, V. and George, E. I. (2015), "Fast Bayesian Factor Analysis via Automatic Rotations to Sparsity," *Journal of the American Statistical Association*.

Salomatin, K., Yang, Y., and Lad, A. (2009), "Multi-Field Correlated Topic Modeling," in *SIAM International Conference on Data Mining*, pp. 628–637.

Sammel, M. D., Ryan, L. M., and Legler, J. M. (1997), "Latent Variable Models for Mixed Discrete and Continuous Outcomes," *Journal of the Royal Statistical Society. Series B*, 59, 667–678.

Schäfer, J. and Strimmer, K. (2005), "An Empirical Bayes Approach to Inferring Large-Scale Gene Association Networks," *Bioinformatics*, 21, 754–764.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Shi, J.-Q. and Lee, S.-Y. (2000), "Latent Variable Models with Mixed Continuous and Polytomous Data," *Journal of the Royal Statistical Society. Series B*, 62, 77–87.

Spearman, C. (1904), "General Intelligence, Objectively Determined and Measured," *The American Journal of Psychology*, 15, 201–292.

Steele, R. J. and Raftery, A. (2010), "Performance of Bayesian Model Selection Criteria for Gaussian Mixture Models," in *Frontiers of Statistical Decision Making and Bayesian Analysis*, pp. 113–130, New York, Springer.

Stranger, B. E., Montgomery, S. B., Dimas, A. S., Parts, L., Stegle, O., Ingle, C. E., Sekowska, M., Smith, G. D., Evans, D., Gutierrez-Arcelus, M., Price, A., Raj, T., Nisbett, J., Nica, A. C., Beazley, C., Durbin, R., Deloukas, P., and Dermitzakis, E. T. (2012), "Patterns of Cis-Regulatory Variation in Diverse Human Populations," *PLoS Genetics*, 8, e1002639.

Strehl, A. and Ghosh, J. (2003), "Cluster EnsemblesA Knowledge Reuse Framework for Combining Multiple Partitions," *the Journal of Machine Learning Research*, 3, 583–617.

Suvitaival, T., Parkkinen, J. A., Virtanen, S., and Kaski, S. (2014), "Cross-Organism Toxicogenomics with Group Factor Analysis," *Systems Biomedicine*, 2, e29291.

Tipping, M. E. (2001), "Sparse Bayesian Learning and the Relevance Vector Machine," *the Journal of Machine Learning Research*, 1, 211–244.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011), "Mulan: A Java Library for Multi-Label Learning," *the Journal of Machine Learning Research*, 12, 2411–2414.

Tung, H. F. and Smola, A. J. (2014), "Spectral Methods for Indian Buffet Process Inference," in *Advances in Neural Information Processing Systems 27*, pp. 1484–1492.

van Dyk, D. A. and Meng, X.-L. (2001), "The Art of Data Augmentation," *Journal of Computational and Graphical Statistics*, 10, 1–50.

Virtanen, S., Klami, A., and Kaski, S. (2011), "Bayesian CCA via Group Sparsity," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 457–464.

Virtanen, S., Klami, A., Khan, S. A., and Kaski, S. (2012), "Bayesian Group Factor Analysis," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, vol. 22, pp. 1269–1277.

Wen, Z. and Yin, W. (2013), "A Feasible Method for Optimization with Orthogonality Constraints," *Mathematical Programming*, 142, 397–434.

West, M. (1987), "On Scale Mixtures of Normal Distributions," *Biometrika*, 74, 646–648.

West, M. (2003), "Bayesian Factor Regression Models in the "Large p, Small n" Paradigm," in *Bayesian Statistics 7, eds. J.M. Bernardo et al.*, pp. 723–732, Oxford University Press.

Witten, D., Tibshirani, R., Gross, S., and Narasimhan, B. (2013), *PMA: Penalized Multivariate Analysis*, R package version 1.0.9.

Witten, D. M. and Tibshirani, R. J. (2009), "Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data," *Statistical Applications in Genetics and Molecular Biology*, 8, 1–27.

Yin, G. (2009), "Bayesian Generalized Method of Moments," *Bayesian Analysis*, 4, 191–207.

Yuan, M. and Lin, Y. (2006), "Model Selection and Estimation in Regression with Grouped Variables," *Journal of the Royal Statistical Society. Series B*, 68, 49–67.

Zeger, S. L. and Liang, K.-Y. (1986), "Longitudinal Data Analysis for Discrete and Continuous Outcomes," *Biometrics*, 42, 121–130.

Zellner, A. (1996), "Bayesian Method of Moments (BMOM) Analysis of Mean and Regression Models," in *Modelling and Prediction Honoring Seymour Geisser*, eds. J. C. Lee, W. O. Johnson, and A. Zellner, pp. 61–72, Springer, New York.

Zhao, S. and Li, S. (2012), "A Co-Module Approach for Elucidating Drug-Disease Associations and Revealing Their Molecular Basis," *Bioinformatics*, 28, 955–961.

Zhong, M. and Girolami, M. (2012), "A Bayesian Approach to Approximate Joint Diagonalization of Square Matrices," in *Proceedings of the 29th International Conference on Machine Learning*, pp. 647–654.

Zhou, T., Tao, D., and Wu, X. (2011), "Manifold Elastic Net: A Unified Framework for Sparse Dimension Reduction," *Data Mining and Knowledge Discovery*, 22, 340–371.

Zou, H. and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society. Series B*, 67, 301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2006), "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, 15, 265–286.

# Biography

Shiwen Zhao was born in Henan, China on June 28, 1986. He received his Bachelor of Engineering and Master of Science in Tsinghua University, Beijing, China in June 2009 and June 2012 respectively. In August 2012 he enrolled as a graduate student in Computational Biology and Bioinformatics program at Duke University, Durham, North Carolina, United States. He graduated with a Doctor of Philosophy in May 2016.