NEW TOOLS FOR BAYESIAN CLUSTERING AND FACTOR ANALYSIS

by

Hanyu Song

Department of Statistical Science
Duke University

Date: __________________________

Approved:

______________________________

David B. Dunson, Supervisor

______________________________

David B. Dunson

______________________________

Amy Herring

______________________________

Li Ma

______________________________

Cliburn Chan

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University

2022
ABSTRACT

NEW TOOLS FOR BAYESIAN CLUSTERING AND FACTOR ANALYSIS

by

Hanyu Song

Department of Statistical Science
Duke University

Date: __________________________

Approved: ______________________

David B. Dunson, Supervisor

David B. Dunson

Amy Herring

Li Ma

Cliburn Chan

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University

2022
Abstract

Traditional model-based clustering faces challenges when applied to mixed scale multivariate data, consisting of both categorical and continuous variables. In such cases, there is a tendency for certain variables to overly influence clustering. In addition, as dimensionality increases, clustering can become more sensitive to kernel misspecification and less reliable. In Chapter 1, we propose a simple local-global Bayesian clustering framework designed to address both of these problems. The model assigns a separate cluster ID to each variable from each subject to define the local component of the model. These local clustering IDs are dependent on a global clustering ID for each subject through a simple hierarchical model. The proposed framework builds on previous related ideas including consensus clustering, the enriched Dirichlet process, and mixed membership models. We show its property of local-global borrowing of information and ease of handling missing data. As a canonical special case, we focus on a simple Dirichlet over-fitted local-global mixture, for which we show that the extra global components of the posterior can be emptied asymptotically. This is the first such result applicable to a broad class of over-fitted finite mixture of mixtures models. We also propose kernel and prior specification for the canonical case and show it leads to a simple Gibbs sampler for posterior computation. We illustrate the approach using simulation studies and applications, through which we see the model is able to identify relevant variables for clustering. Large data have become the norm in many modern applications; they often cannot be easily moved across computers or loaded into memory on a single computer. In such cases, model-based clustering, which typically uses the inherently serial Markov chain Monte Carlo for computation, faces challenges. Existing distributed algorithms have emphasized nonparametric Bayesian mixture models and typically require moving raw data across workers. In Chapter 2, we introduce a nearly embarrassingly parallel algorithm for clustering under a Bayesian overfitted finite
mixture of Gaussian mixtures, which we term distributed Bayesian clustering (DIB-C). DIB-C can flexibly accommodate data sets with various shapes (e.g. skewed or multi-modal). With data randomly partitioned and distributed, we first run Markov chain Monte Carlo in an embarrassingly parallel manner to obtain local clustering draws and then refine across workers for a final clustering estimate based on any loss function on the space of partitions. DIB-C can also estimate cluster densities, quickly classify new subjects and provide a posterior predictive distribution. Both simulation studies and real data applications show superior performance of DIB-C in terms of robustness and computational efficiency.

Chapter 3 develops a simple factor analysis model in light of the need for new models for characterizing dependence in multivariate data. The multivariate Gaussian distribution is routinely used, but cannot characterize nonlinear relationships in the data. Most non-linear extensions tend to be highly complex; for example, involving estimation of a non-linear regression model in latent variables. We propose a relatively simple class of Ellipsoid-Gaussian multivariate distributions, which are derived by using a Gaussian linear factor model involving latent variables having a von Mises-Fisher distribution on a unit hyper-sphere. We show that the Ellipsoid-Gaussian distribution can flexibly model curved relationships among variables with lower-dimensional structures. Taking a Bayesian approach, we propose a hybrid of gradient-based geodesic Monte Carlo and adaptive Metropolis for posterior sampling. We derive basic properties and illustrate the utility of the Ellipsoid-Gaussian distribution on a variety of simulated and real data applications.
To Azeem and our soon-to-be-born daughter
Contents

Abstract

List of Tables

List of Figures

Acknowledgements

1 Local-global Bayesian clustering for mixed-scale multivariate data 1

1.1 Introduction .......................................................... 1

1.2 Methodology .......................................................... 5

1.2.1 Notation ............................................................ 5

1.2.2 Limitations of latent modeling of ordinal variables ............ 5

1.2.3 Limitations of distance-based methods for mixed-scale data ... 8

1.2.4 CLOG: clustering via local-global partitions .................. 12

1.3 Properties of CLOG .................................................. 14

1.3.1 Local-global borrowing of information ......................... 14

1.3.2 Handling missing data .......................................... 17

1.4 Properties for a canonical case .................................... 18

1.4.1 Asymptotic behaviour of posterior ............................ 18

1.5 Posterior computation ............................................. 20

1.5.1 Kernel specification ............................................ 21

1.5.2 Prior specification ............................................. 21

1.5.3 Posterior sampling ............................................ 23

1.6 Simulation study .................................................. 25
2.5 Conclusion ................................................. 114

Appendices ............................................. 115

.1 MCMC procedures for sampling of local clustering (step 2) .......... 115
.2 Collapsed Gibbs sampler for global cluster refinement ............. 117
.3 Sampling model parameters (step 5) .......................... 118
.4 Definition of clustering validation metrics .......................... 120

3 Curved factor analysis with the Ellipsoid-Gaussian distribution 122

3.1 Introduction .............................................. 122
3.2 The von Mises-Fisher linear factor model ....................... 126
3.3 Ellipsoid-Gaussian distribution ................................ 129
  3.3.1 Identifiability of the model parameters ..................... 130
  3.3.2 Limiting behavior ...................................... 131
3.4 Posterior computation ...................................... 132
3.5 Simulation studies ........................................ 134
  3.5.1 Experiment setups ..................................... 134
  3.5.2 Simulated data sets .................................... 135
  3.5.3 Simulation results .................................... 136
3.6 Real data applications ..................................... 138
  3.6.1 Horse mussel data ..................................... 138
  3.6.2 Air quality data ....................................... 138
3.7 Discussion ................................................ 141

Appendices ............................................. 142

.1 Proofs of propositions and lemmas ............................... 142
List of Tables

1.1 The clustering performance of the Byar prostate cancer data* .......................... 36

2.1 Selected notation and descriptions ................................................................. 84

2.2 An example dictionary of item indices, mixture components and cluster labels ................................................................. 91

2.3 A bijective map between the item indices and groups .................................. 93

2.4 The clustering performance of the flow cytometry data across various methods for $N = 12k, 120k$ and $1m$. In particular, the performance of DBSCAN and K-means are invariant to the number of workers. For DIB-C, the median of each statistic obtained from using varying number of workers is reported. Yellow and green indicate the best and the second best performance under each scenario respectively. ......................................................... 111

.1 Reference bounds for $\tau$ under different latent dimensions $k$ ....................... 150

.2 The parameters involved in the sampler ............................................................. 153
List of Figures

1.1 Visualization of the serum prostatic acid phosphatase levels from the byar prostate cancer data on the original scale (left) and after log transformation (right) ................................................................. 3

1.2 Bivariate data are simulated under a two-cluster model, with the first variable having separation $\mu$ in the two clusters and the second variable having separation 0. Clustering $C_1$ thresholds the first (signal) variable and clustering $C_2$ thresholds the second (noise) variable. The figure shows violin plots of the difference in log posterior probabilities of $C_1$ and $C_2$ based on observing the continuous $X_{i2}$ (left) or the thresholded $Y_i = I(X_{i2} \geq 0)$ . . . 8

1.3 Fitting the PAM algorithm to data generated from (1.3) with $\mu = 100$. Jitter is added to $Y$ to help visualize the points. The algorithm prefers to split based on $Y$ rather than $X_1$, despite the massive separation in $X_1$. . . 10

1.4 The left panel shows $C_1$ and the right panel shows $C_2$, with $\mu = 5$ and the clusters distinguished by color. The $K$-medoids objective will prefer the clustering on the left. The discrete variable has a small amount of noise added to aid visualization. The black points are the medoids of the clusters. 11

1.5 Illustration of the proposed local-global clustering model: scatter plots of a two-dimensional data set, with points colored by local cluster IDs $s_{i1}, s_{i2}$ and global cluster IDs $C_i$’s from left to right, respectively. . . . . . . . . . . . 13

1.6 Graphical model representation of CLOG. The boxes are “plates” representing replicates. The outer plate represents observations, while the inner plate represents variables. . . . . . . . . . . . . . . . . . . . . . . . . 13

1.7 A generalized pairs plot of a data set generated under configuration (d). The labels quant., ord. and nom. refer to quantitative, ordinal and nominal variables respectively. . . . . . . . . . . . . . . . . . . . . . . . . 28

1.8 Simulation results: distributions of the adjusted Rand indices in 20 replicated experiments. Note that box plots associated with Kprototype is missing in the first row because the method is only applicable to mixed-scale data. . . . . . . . . . . . . . . . . . . . . . . . . 30
1.9 Simulation results of CLOG for a randomly selected Quant noise data set: histograms of posterior draws of the KL statistic for a signal and noise variable.

1.10 Clustering results of the diamond data of a subset of the methods: scatter plots of price vs size, with shapes and colors of the points indicating certification and cluster estimates.

1.11 Clustering results of the diamond data with added noise variables: (left) histograms of posterior draws of the KL statistic for size and a noise variable and (right) scatter plots of price vs a noise variable for CLOG and Kprototype.

1.12 Results of the Byar prostatic acid phosphatase data: (left) the relationship between logged serum prostatic acid phosphatase levels (in King-Armstrong units) and primary tumour sizes, with the shape of the points indicating the presence of bone metastasis (circle: absence; triangle: presence); (right) the relationship between systolic blood pressure and serum haemoglobin levels, with the shape of the points indicating cardiovascular disease history status (circle: no; triangle: yes). All points are colored by prostate cancer stage or clustering estimates. The clustering estimates are recoded into 3 and 4 to match the cancer stages.

1.13 Results of CLOG for the Byar prostatic acid phosphatase data: the histogram of the posterior draws of the KL statistic associated with serum haemoglobin levels and serum prostatic acid phosphatase levels. The cluster labels are recoded into 3 and 4 to match the cancer stages.

1.14 Supplementary clustering results of the diamond data in the dimensions of size and certification colored by the clustering estimates.

1.15 Clustering results of the diamond data in the dimensions of size and clarity colored by the clustering estimates.

2.1 Algorithm flowchart of DIB-C. In step 2, the copies of frames refer to samples of local clustering, with the red ones representing those that are adjusted in step 3. Based on the adjusted samples of local clustering, a global clustering is estimated in step 4. Step 5 Sampling model parameters is excluded due to space constraint.
2.2 A sample of local cluster (top) and subcomponent (bottom) allocations before refinement for subset 1, 2 and 3 when the data set is partitioned and distributed to 4 workers. 6% randomly selected data points are plotted.

2.3 Representation of items via the two indexing systems. Blue and white represent non-empty and empty subcomponents respectively.

2.4 The sample of sub-component allocations in the reference subset.

2.5 Left: the scatter plot of a synthetic data set simulated with \( N = 12,000 \) observations. Right: the scatter plot of 10,000 observations randomly drawn from the 1 million training set of the flow cytometry data, where chl small, pe and fsc perp represent chlorophyll level, phycoerythrin level, and forward scatter respectively.

2.6 Increase in classification accuracy after refinement in local subsets of the training sets from the synthetic (top row) and the flow cytometry data (bottom row). 12k, 120k and 1m indicate the data size of the training sets. For each data and node setting, we randomly select one of ten replicates and include a violin plot of R statistics, where R is the number workers (or subsets).

2.7 Scatter plots of clustering results associated with the training set of one of the synthetic data sets with \( N = 10^6 \). The plot on the left represents the final clustering estimates when the data are randomly distributed to 1, 5, 20 and 120 workers respectively. The plot on the right shows subsets results associated with the 20 workers on the left; specifically, it represents a sample of cluster allocations of subset 5, 8, 15 and 20 before global cluster refinement.

2.8 Classification performance of the training and test sets associated with the synthetic (left) and flow cytometry (right) data, as measured by accuracy, ARI, and F-measure.

2.9 Scatter plots of clustering results of the test set associated with \( N = 1 \) million data size setting. The top left plot corresponds to the manually-gated labels. The remaining four represent the clustering results from using 1, 60 and 100 workers respectively. In each scatter plot, only a random subset of the data points are included; and the x, y and z coordinates are log(side scatter), log(phycoerythrin level) and log(chlorophyll level), respectively. These three explain the most variation in the data among the four variables (according to principle component analysis results).
2.10 Scatter plots of 10,000 data points simulated from the posterior predictive distributions resulting from an analysis conducted with data distributed across 40 workers for the synthetic data (left) and the flow cytometry data (right). In the plot on the right, chl, small, pe and fsc perp represent chlorophyll level, phycoerythrin level, and forward scatter respectively.

2.11 The computation time associated with the training sets of the synthetic (top) and flow cytometry data (bottom). For each data setting, we include sampling local clustering (the first column), global cluster refinement (the second column), global clustering estimation time (the third column) and total time of the above steps (the fourth column).

3.1 Demonstration of the Gaussian linear factor model with $p = 3, k = 2$. The yellow plane represents the column space of $\Lambda$ and contains all of the blue points sampled from $N_3(0, \Lambda\Lambda^T)$. The red vectors represent eigenvectors of $\Lambda\Lambda^T$; equivalently the left singular vectors of $\Lambda$. The figure on the right shows the view perpendicular to the plane, illustrating that the data have a Gaussian distribution centered around the plane.

3.2 Scatter plots of two curved data sets. In particular, H, L, S, W, and M represent shell height (mm), shell length (mm), shell mass (g), shell width (mm) and muscle mass (g) respectively.

3.3 Demonstration of the von Mises-Fisher linear factor model with $p = 3, k = 2$. The green and red lines represent the mean direction $\Lambda\mu$ and principal axes of the ellipse, respectively.

3.4 Scatter plots of data sampled from the Ellipsoid-Gaussian in $\mathbb{R}^3$ with varying parameters.

3.5 Rotating the samples from a vMF($\mu, \tau$) (black) by $\Gamma$ results in samples from a vMF($\Gamma\mu, \tau$) (red). Mapping the black points by $\Lambda$ and the red points by $\Lambda\Gamma^T$, respectively, results in the same set of points (blue).

3.6 The log posterior predictive density of the test sets evaluated under each model as a function of the test set sizes, with red, blue and green corresponding to Ellipsoid-Gaussian, Gaussian linear factor models and mixtures of factor analyzers, respectively.
3.7 The scatter plot matrices of the posterior predictive (green) juxtaposed with
the original data (red) for each method, where the models were fitted to
the very curved data set in $\mathbb{R}^{10}$ with 1000 observations and 20% random
missingness (top row) and the hybrid Rosenbrock data in $\mathbb{R}^{3}$ with 300 ob-
servations (bottom row) respectively. Letters p and o stand for posterior
predictive and original data respectively. ........................................ 137

3.8 Results associated with Ellipsoid-Gaussian. Left: the posterior mean of the
post-processed factor loadings of $\Lambda$. Right: the posterior mean (black) and
the 95% credible band (blue ribbon) of muscle mass (top) and the benzene
level (bottom) as a function of shell length and the CO level respectively,
holding other covariates at their sample mean level. .......................... 139

3.9 The scatter plot matrices of the posterior predictive (green) juxtaposed with
the original data (red); the model was fitted to data sets with 20% in the
test sets. Letters p and o stand for posterior predictive and original data
respectively. .......................................................... 140

.10 As the concentration of a von-Mises Fisher distribution increases, the dis-
tribution approaches a degenerate Gaussian, supported on the hyperplane
perpendicular to $\mu$. .................................................. 146

.11 The log posterior predictive density of the test sets evaluated under each
model as a function of the test set sizes, with red, blue and green corre-
sponding to Ellipsoid-Gaussian, Gaussian linear factor models and mix-
tures of factor analyzers, respectively. In Fig. .11b the three curves are
exactly overlapping, indicating the performance of the three methods is
comparable. .................................................. 161

.12 The trace plot (top row) and the auto correlation function plot (bottom row)
of some precision parameters for the simulated and real data sets. ....... 162

.13 The scatter plot matrices of the posterior predictive (green) juxtaposed with
the original data (red) for each method, where the models were fitted to the
shell shape in $\mathbb{R}^{3}$ with 400 observations (top row), approx.Gaussian data in
$\mathbb{R}^{6}$ with 1000 observations and 30% random missingness (middle row) and
data from Gaussian linear factor model in $\mathbb{R}^{6}$ with 1000 observations and
20% random missingness (bottom row). Letters p and o stand for posterior
predictive and original data respectively. ........................................ 163
The posterior mean (black) and the 95\% credible band (blue ribbon) of muscle mass (top) and the benzene level (bottom) as a function of some covariates while holding other covariates at their sample mean level based on the Ellipsoid-Gaussian fit.
Acknowledgements

First I want to express my sincere gratitude to my advisor and role model, David Dunson, for his constant support, encouragement, guidance throughout my doctoral journey. As an advisor, he provided prompt, detailed and insightful feedback, from which I learned a lot. I was not an independent researcher, but he was patient and gave me ample room to explore and develop my skills. He also trusted me to work remotely so that I could be with my partner. As a highly accomplished researcher, he still constantly strives to solve challenging and impactful problems; I was and will continue to be inspired by that.

I am grateful to my committee members Amy Herring, Li Ma and Cliburn Chan. Li and Cliburn were committee members for both my Master’s and PhD defenses and provided valuable feedback on both. They also wrote numerous recommendation letters for me when I was applying for graduate schools and the 2021 Facebook fellowship. Amy invited me to work on an exciting applied project in medicine, through which I came to know many respectable researchers in a variety of fields. I saw the power of statistics and felt more motivated to study and apply statistics to real world problems for the benefit of humanity.

I was also fortunate to work as an intern at SAS Institute Inc., where I was assigned the problem of distributed clustering. Without their trust and financial support I would not have completed that chapter. I would also like to acknowledge support from the United States National Science Foundation grant 1546130, National Institute of Environmental Health Sciences of the United States National Institutes of Health under Grants R01ES027498 and R01ES028804, and multiple Summer Research Fellowships from the Duke Graduate School.

I am thankful to have met many great friends during my Master’s and doctoral years, including Xi Chen, Xu Chen, Federico Farrari, Ghassen Jerfel, Bai Li, Jiurui Tang, Xiangyu Wang and Shuxi Zeng. In particular, Xiangyu was a patient mentor to me when I was new to research during my Master’s program. Without his guidance, I would not have been admitted to the prestigious PhD program at Duke.

Finally, I would like to thank my parents for their unconditional love. They have provided me
with financial assistance and supported whatever decisions I make. After my mother retired, she moved here and cooked for me as I pushed to finish this work. I also feel incredibly lucky to have met my husband and best friend Azeem while at Duke. Pursing a PhD in statistics together made the difficult times enjoyable. There have been a lot of struggles for me during my doctoral journey, and he was always there when I needed help and support. I learned a lot from him through our discussions and was inspired by his constant passion for research. I could not have asked for a better partner. This work is dedicated to him and our daughter due in two months.
Chapter 1

Local-global Bayesian clustering for mixed-scale multivariate data

Key words: Consensus clustering; Hierarchical model; Mixed membership model; Model-based clustering

1.1 Introduction

Data sets with mixed-scale variables, consisting of quantitative, ordinal, nominal or other types of variables, have become the norm. Medical data collected from patients with a certain disease routinely contain quantitative variables such as age, weight, systolic blood pressure, nominal variables such as sex, and ordinal variables such as symptom severity score. Socioeconomic data often contain quantitative variables such as household income, ordinal variables such as education level, and nominal variables such as occupation. One common task of interest is constructing meaningful clusters. For example, doctors are interested in identifying subgroups having clinically different biomarkers and clinical covariate profiles but there are no labels available.

There is a rich literature on frequentist clustering methods that handle mixed-scaled data. They typically fall into three categories: i) specifying a pairwise distance metric and identifying the partition that minimizes the sum of the pairwise distances (Huang, 1998; Ahmad and Dey, 2007), ii) performing dimension reduction of the mixed-scale data and then clustering in the reduced space (Kiers, 1991; Pagès, 2004) or iii) performing simultaneous dimension reduction and clustering (Vichi et al., 2019; Yamamoto and Hwang, 2014). One widely used distance for methods in category is i) Gower’s dissimilarity measure, which is calculated as the simple average of the distances between each pair of variables. For quantitative and ordinal variables, the distance is the range-normalized Manhattan distance; for nominal variables, the distance is 1 if two records are different and 0 otherwise (Dice distance). Because of this, the distance between nominal variables
tends to be overly dominant in the averaged distance, i.e. Gower’s distance, meaning that nominal variables tend to be the main driver of partitioning structure, which is a strong hidden assumption.

There is also a rich literature on Bayesian model-based methods for clustering. In this case a Dirichlet process (DP) mixture or its extensions are often used (Denti et al., 2021; Teh et al., 2006; Rodríguez et al., 2008; Blei et al., 2003). A realization from a DP is an almost surely discrete probability measure, which implies that any sample from the probability measure presents ties with positive probability. This naturally induces a partition where observations from the same atom are clustered together. Other methods use finite mixture models or their variations (Malsiner-Walli et al., 2016; Miller and Harrison, 2018), with each component commonly corresponding to one cluster in the data.

One common issue of these model-based methods is substantial sensitivity to kernel misspecification. Take the Byar prostate cancer data (Byar and Greene, 1980) as an example. The data was collected from a randomized clinical trial of patients with prostatic cancer in either stage 3 or 4 of the disease (Byar and Greene, 1980). Twelve pre-trial covariates were measured, among which the serum prostatic acid phosphatase level is an important indicator for determining the disease progression (Byar and Greene, 1980). Its distribution is highly right-skewed and a simple log transformation makes it closer to a normal, as visualized in Fig. 1.1. Consider clustering the data using a finite mixture of Gaussian with the ordinal variables assuming a latent continuous structure (Agresti, 2015). First cluster with the serum prostatic acid phosphatase level on the original scale and then on the logged scale. The accuracy and adjusted Rand index (Hubert and Arabie, 1985) increased from 0.02 and 0.60 to 0.21 and 0.73 respectively. This drastic increase is a vivid proof that clustering results are sensitive to kernel misspecification.

In addition to issues of substantial sensitivity to kernel misspecification (Miller and Dunson, 2018), the model-based methods are mostly not designed for mixed-scale data. One exception (Carmona et al., 2019) jointly models latent variables underlying the observed mixed-scale variables using a multivariate Gaussian kernel. In particular, each quantitative or ordinal variable is associated with one latent variable, and a nominal variable with \(d\) levels corresponds to \(d - 1\) latent variables. As the number of variables—particularly the nominal ones—increases, the number of
latent variables can increase drastically, which has been known to lead to poor mixing of Markov chains targeting the posterior. In addition, as the dimension increases, the atoms generated from the base distribution are farther apart, with the distance between two atoms increasing at least linearly in the number of latent variables, meaning that observations are less likely to be grouped into a small number of clusters; this phenomenon was explored in Chandra et al. (2020).

The mixed-scale clustering method (Carmona et al., 2019) also suffers from the problem where data from certain measurement scales tend to unduly influence the clustering results. Consider clustering a data set with quantitative and ordered categorical variables. An example in Section 1.2.2 shows how the introduction of discrete variables can introduce inconsistency into the posterior and produce meaningless clusters.

In order to address the problems of huge sensitivity to kernel misspecification and certain variables overly dominating clustering, we propose a simple clustering via local-global (CLOG) Bayesian framework. The CLOG model assigns a separate cluster ID to each variable from each subject to define the local component of the model. These local clustering IDs are dependent on a global cluster ID for each subject through a simple hierarchical model. We note that the components associated with each local cluster are shared across different global clusters, enabling borrowing of information.
A related local-global clustering idea was explored in Dunson (2009) for the analysis of functional and longitudinal data. The primary goal was not clustering, but to induce a sparse representation for moderate to high-dimensional random effects allowing a combination of local and global borrowing of information. Similar ideas were explored in Dunson (2010); Soriano and Ma (2019); Stephenson et al. (2020).

CLOG is also related to Bayesian consensus clustering (BCC) (Lock and Dunson, 2013). BCC is motivated by obtaining a consensus among clusterings produced from multiple types of data measured on the same study subjects. For example, one may measure three different types of omics data and clinical data for each patient. BCC can identify clusters of patients with different omics and clinical profiles, while also producing one clustering of patients synthesizing the data type-specific clusterings. In contrast, CLOG incorporates variable-specific instead of type-specific local cluster memberships. The specific modeling structure is also different, with CLOG imposing a conditional independence structure on the observed data given the local cluster IDs to aid in applications to mixed scale data. In particular, the proposed structure bypasses the problem of certain variables dominating the clustering simply due to their measurement scale.

Both CLOG and BCC incorporate a nested partition structure; there is a rich literature on alternative nested partitioning models. Examples include the common atom model (CAM) (Denti et al., 2021), the nested Dirichlet process (nDP) (Rodríguez et al., 2008) and enriched Dirichlet process (EDP) (Wade et al., 2011). EDP was proposed for flexible modelling of an unknown random joint probability measure by factoring it in terms of the random marginal and conditionals, which made it very useful for density estimation and prediction problems; see Wade et al. (2014) for more details. Both CAM and nDP were designed for nested or partially exchangeable data. In contrast, our model is designed for fully exchangeable data.

Our model is also reminiscent of mixed-membership models (Blei et al., 2003; Airoldi et al., 2008; Erosheva, 2003; Pritchard et al., 2000) since the components are also shared across groups. However, these models are designed for grouped data, where each subject is itself a collection of data and each collection can belong to multiple groups. For example, a document is a collection of words and each document can belong to multiple topics. In contrast, our model is designed for
exchangeable data with multiple types and each subject can only belong to one group.

This paper is organized as follows. In Section 1.2, we propose the local-global Bayesian clustering framework. Section 1.3 establishes the desirable properties that CLOG enjoys, such as local-global borrowing of information and the ease of handling missing data in the case of missing at random. As a canonical case, we focus on a simple Dirichlet over-fitted local-global mixture. In Section 1.4, we show its posterior consistency for density estimation and that the empty global components can be emptied asymptotically. Section 1.5 proposes kernel and prior specification for the canonical case and shows it leads to a simple Gibbs sampler. Section 1.6 provide extensive simulation studies comparing our method with some popular clustering methods. Section 1.7 considers applications to a diamond and a prostate cancer data set.

1.2 Methodology

1.2.1 Notation

Let $y_{ij}$ be the data for the $j$th variable of the $i$th observation. Let $y_i = (y_{i1}, \ldots, y_{ip})^T$. Denote the complete data matrix by $Y$, with $Y = (y_1, \ldots, y_n)^T$. Let $C_i \in \{1, \ldots, K\}$ denote the global cluster ID for subject $i$ ($i = 1, \ldots, n$), and let $s_{ij} \in \{1, \ldots, K_j\}$ be the local clustering indicator for subject $i$ and variable $j$ ($j = 1, \ldots, p$). Let $I_c, I_o, I_q$ be the indices of the unordered categorical or nominal variables, ordered categorical including ordinal and count variables, and quantitative variables, respectively, with $I_c \cup I_o \cup I_q = \{1, \ldots, p\}$, $I_c \cap I_o = \emptyset$, $I_c \cap I_q = \emptyset$ and $I_o \cap I_q = \emptyset$. Let $d_j$ be the number of levels in variables $j \in I_c \cup I_o$. Let $\lambda := (\lambda_1, \ldots, \lambda_K)$ be the vector of cluster weights, and $\psi^{(j)}_\ell := (\psi^{(j)}_{\ell,1}, \ldots, \psi^{(j)}_{\ell,K_j})$ be a probability vector of length $K_j$ associated with local variable $j$ and global cluster $\ell$.

1.2.2 Limitations of latent modeling of ordinal variables

A textbook (Agresti, 2015) approach for modeling ordinal variables is the use of a latent score variable that maps through a monotonic function to the discrete ordinal variable. The use of these latent
score models has been applied to clustering (Carmona et al., 2019) as a natural way to jointly model ordinal variables with continuous variables. This model is simple and can have a nice interpretation when there is a natural latent variable, but has undesirable behavior since the dimension of the posterior (which includes $n$ latent score variables) must grow linearly with the sample size. In this section we will show how in a simple setting the posterior distribution of a latent score model has undesirable behavior.

We will consider two clustering models. For Model 1, we have iid observations $X_i = (X_{i1}, X_{i2})$ for $i = 1, \ldots, n$ from a mixture of two bivariate normals. We wish to recover the mixture component from which the observation was sampled as the unknown cluster assignment. For Model 2, we observe $\tilde{X}_i = (X_{i1}, Y_i)$, where $Y_i = I(X_i \geq 0)$ is considered a mapping of the latent score $X_{i2}$ to an observed ordinal variable. We will show that the posterior for Model 1 can behave very differently from the posterior of Model 2. The generative model is

$$
C_i \overset{iid}{\sim} \text{Bern}(1/2) \quad \mu_{jk} \overset{iid}{\sim} \text{N}(0, 1) \quad j = 1, 2, \quad k = 0, 1
$$

$$
X_{i1} \mid C_i \overset{ind}{\sim} \text{N}(\mu_{1C_i}, 1) \quad X_{i2} \mid C_i \overset{ind}{\sim} \text{N}(\mu_{2C_i}, 1) \quad Y_i = I(X_{i2} \geq 0) \quad i = 1, \ldots, n.
$$

We want to compare the marginal posteriors $p(C \mid X)$ and $p(C \mid \tilde{X})$. For $j = 1, 2$ and $k = 0, 1$ let $X_{jk}$ be the subvector containing $\{X_{ij} : C_i = k\}$ and let $n_{jk}$ be the size of this subvector. For $k = 1, 2$ we let $n_k$ be the size of $\{i : C_i = k\}$ and $m_{kh}$ be the size of $\{i : C_i = k, Y_i = h\}$. Using marginalization results for Gaussians, we have

$$
p(C \mid X) = \int p(C, \mu_{10}, \mu_{11}, \mu_{20}, \mu_{21} \mid X) \, d\mu_{10} \, d\mu_{11} \, d\mu_{20} \, d\mu_{21}
$$

$$
\propto \prod_{j=1}^{2} \prod_{k=0}^{1} N_{n_{jk}}(X_{jk} \mid 0, I + 11^T).
$$

From this we can find

$$
p(C \mid \tilde{X}) \propto \int p(C, X_1, X_2, Y) \, dX_2 \propto \int p(C \mid X) \prod_{i : Y_i = 0} I(X_{i2} < 0) \prod_{i : Y_i = 1} I(X_{i2} \geq 0) \, dX_2.
$$

(1.1)

We are able to evaluate this integral using Lemma 2. The Lemma gives

$$
p(C \mid \tilde{X}) \propto \frac{N_{n_{10}}(X_{10} \mid 0, I + 11^T)N_{n_{11}}(X_{11} \mid 0, I + 11^T)}{(n_0 + 1)(n_1 + 1)} \frac{n_{01}^n}{m_{01}^n} \frac{n_{11}^m}{m_{11}^m}.
$$

(1.2)
The posteriors in (1.1) and (1.2) behave very differently when one of the variables has the same mean in both clusters. Assume that the data are generated with $\mu_{10} = -\mu_{11} = \mu$ for some $\mu > 0$ and that $\mu_{20} = \mu_{21} = 0$. This means that $X_2$ and $Y$ are independent of $C$ in both Model 1 and Model 2. We consider two candidate clusterings and compare their value in the posterior. The first clustering $C_1$ is given by $C_{1i} = I(X_{i1} \geq 0)$, which only uses the signal variable and ignores the noise variable; this will have an expected accuracy of $\Phi(\mu) > 0.5$ when used to recover the $C_i$ used to generate the data. The second clustering $C_2$ is given by $\tilde{C}_i = I(X_{i2} \geq 0) = Y_i$, which splits based on a variable that is independent of the generative clustering. The clustering is a random guess and has an expected accuracy of $1/2$.

To see how the clusterings $C_1$ and $C_2$ compare in the posterior, we calculate $\log p(C_1 \mid X) - \log p(C_2 \mid X)$ and $\log p(C_1 \mid \tilde{X}) - \log p(C_2 \mid \tilde{X})$ for 100 simulated data sets. When the difference is positive, the posterior prefers the reasonable clustering given by the true signal. If the difference is negative, then the posterior gives higher probability to a clustering that contains no information about the generative clustering. We will simulate 100 data sets of size $n = 1000$ for each $\mu$ in a sequence from $\mu = 0.5$ to $\mu = 1.5$. In all simulations, the model is correctly specified. The results are given in Figure 1.2

Figure 1.2 shows very different behaviors for the two models. When we observe the continuous variable $X_{i2}$, we see that clustering on the signal is always superior to clustering on the noise variable across all signal strengths $\mu$. When we observe the ordinal variable $Y$, the behavior is very different. For values of $\mu < 1$, the model tends to give larger posterior probability to the noise clustering $C_2$. As the signal strength increases, the posterior does prefer the clustering derived from the signal variable. This behavior is striking because when $\mu = .94$, the clustering $C_0$ will have an expected accuracy as high as 0.82, but in 98/100 simulations the posterior gives more probability to $C_2$.

The only difference between the data sets $X$ and $\tilde{X}$ is replacing a continuous variable with a derived ordinal variable using the commonly used latent score model. The undesirable behavior of the posterior in these simple examples provides a clear motivation for the need of other approaches when clustering mixed data.
Figure 1.2: Bivariate data are simulated under a two-cluster model, with the first variable having separation $\mu$ in the two clusters and the second variable having separation 0. Clustering $C_1$ thresholds the first (signal) variable and clustering $C_2$ thresholds the second (noise) variable. The figure shows violin plots of the difference in log posterior probabilities of $C_1$ and $C_2$ based on observing the continuous $X_{i2}$ (left) or the thresholded $Y_i = I(X_{i2} \geq 0)$.

1.2.3 Limitations of distance-based methods for mixed-scale data

The example given above also exhibits pathological behavior in distance based algorithms. Recall that Gower’s distance uses the scaled (by the range) Manhattan distance on quantitative variables and a zero-one distance on categorical variables, then averages them to obtain a value between zero and one. In a simple case with one quantitative and one categorical variable, the distance exhibits undesirable behavior.

Proposition 1 (Shortcoming of Gower’s distance). Suppose we have observations $(x_i, y_i)$ for $i = 1, \ldots, n$ where $x_i$ is continuous and $y_i$ is a binary-coded categorical variable. If $d_G$ is Gower’s distance, then for any values of $x_i, x_j$, and $x_k$ we have

$$d_G ((x_i, 0), (x_j, 0)) \leq d_G ((x_i, 0), (x_k, 1)).$$

Proof. Let $R$ be the range of $x_i$. From the definition of Gower’s distance we have

$$d_G ((x_i, 0), (x_j, 0)) = \frac{|x_i - x_j|/R + 0}{2} \leq \frac{1}{2} + \frac{|x_i - x_k|}{2R} = d_G ((x_i, 0), (x_k, 1)).$$
The proposition says that no matter how far we perturb the first dimension, the change in distance is smaller than the perturbation from flipping the second dimension. This simple result has significant consequences for the $K$-medoids (see Section 14.3.10 of Hastie et al. (2001) for an overview). Let $\{i_k\}_{k=1}^K$ be the indices of the $K$-medoids and $C : \{1, \ldots, n\} \rightarrow \{i_k\}_{k=1}^K$ be a mapping from the data points to the medoid indices. The loss function for a given $C$ is

$$L(C) = \sum_{i=1}^n d_G((x_i, y_i), (x_{C(i)}, y_{C(i)}))$$

We define two clusterings that we will compare. Let $\text{median}^*$ be a function that returns the index of largest data point less than or equal to the median. We define the medoids

$$i_1 = \text{median}^*(\{x_i : y_i = 1\}) \quad i_2 = \text{median}^*(\{x_i : y_i = 0\})$$

$$\tilde{i}_1 = \text{median}^*(\{x_i : x_i \geq 0, y_i = 1\}) \quad \tilde{i}_2 = \text{median}^*(\{x_i : x_i < 0, y_i = 0\})$$

and the assignments

$$C_1(i) = \begin{cases} 
  i_1 & y_i = 1 \\
  i_2 & y_i = 0
\end{cases} \quad C_2(i) = \begin{cases} 
  \tilde{i}_1 & x_i \geq 0 \\
  \tilde{i}_2 & x_i < 0
\end{cases}$$

The clustering $C_1$ splits on the discrete variable and the clustering $C_2$ splits on based on the median of the continuous variable. Suppose our data is generated from the same true model as the previous section:

$$X_i \sim \frac{1}{2} N(\mu, 1) + \frac{1}{2} N(-\mu, 1) \quad Y_i \sim \text{Bern}(1/2) \quad \text{for } i = 1, \ldots, n. \quad (1.3)$$

The clusters $C_1$ and $C_2$ are visualized in Figure 1.4 for a sample from (1.3) when $\mu = 5$.

It would be reasonable to expect that as $\mu \rightarrow \infty$, the loss function would favor splitting the continuous variable at the zero, but this is not the case.

**Proposition 2.** Assume we have data generated from (1.3). In a sample of $n \geq 2$ data points we have $\mathbb{P}(L(C_1) < L(C_2)) \geq (1 - 2^{-n+1})^2$ for samples from (1.3) for any value of $\mu$. 

9
Figure 1.3: Fitting the PAM algorithm to data generated from (1.3) with $\mu = 100$. Jitter is added to $Y$ to help visualize the points. The algorithm prefers to split based on $Y$ rather than $X_1$, despite the massive separation in $X_1$.

The implication of this result is that $C_2$ is highly unlikely to be the optimal clustering. The small probability where the result fails is the probability that the sets needed to define the medoids are empty. This corresponds to low probability events, such as when $y_i = 1$ for all $i = 1, \ldots, n$.

Proposition 2 implies that algorithms fitting $K$-medoids will prefer clustering $C_2$. This is confirmed by a simulation study in Figure 1.3. In the study, we generated data from (1.3) with $\mu = 100$, making the clusters extremely separated in the first component. Regardless, the PAM algorithm prefers splitting on $Y_i$ instead of $X_i$. 
Figure 1.4: The left panel shows $\mathcal{C}_1$ and the right panel shows $\mathcal{C}_2$, with $\mu = 5$ and the clusters distinguished by color. The $K$-medoids objective will prefer the clustering on the left. The discrete variable has a small amount of noise added to aid visualization. The black points are the medoids of the clusters.
1.2.4 CLOG: clustering via local-global partitions

To overcome the problems discussed in Section 1.1 and demonstrated in Section 1.2.2, we propose a general class of local-global clustering models defined as:

\[ P(C_i = \ell \mid \lambda) = \lambda_\ell, \quad \ell = 1, \ldots, \infty, \quad \text{(global partition)} \]

\[ P(s_{ij} = h \mid C_i = \ell, \psi^{(j)}_1, \psi^{(j)}_2, \ldots) = \psi^{(j)}_{\ell,h}, \quad h = 1, \ldots, \infty, \quad \text{(local partition)} \]

\[ y_{ij} \mid s_{ij} = h, \theta^{(j)}_h \sim f^{u}_{\theta^{(j)}_h}, \quad j \in I_u, \quad (1.4) \]

where \( f^{u}_{\theta^{(j)}_h} \) represents the kernel for variable \( j \) that is of type \( u \in \{q, o, c\} \). In (1.4), we define a local-global partition process, where there is a separate clustering of each variable at the local level and the probabilities \( \psi^{(j)}_{\ell} \) of such cluster assignment depend on the global cluster IDs. This induces a nested partition structure. Conditional on the local cluster IDs, the observations associated with each variable are assumed to be independent with distribution \( f^{u}_{\theta^{(j)}_h} \). We refer to this clustering via a local-global Bayesian framework as CLOG. Figure 1.5 provides an illustration of how the model works in a toy example, where there are two variables and three clusters. The local cluster IDs for \( y_j \) reflect the clustering structure when data are projected to the \( j \)th dimension.

Figure 1.6 shows the graphical model representation of CLOG, from which we see that the observation \( y_{ij} \) is assumed to be independent of the global cluster ID \( C_i \) conditional on the local cluster ID \( s_{ij} \); this breaks the direct dependence of \( C_i \) on data and allows the variable \( j \) to influence the global clustering only through the probabilities \( \psi^{(j)}_{C_i} \), mitigating the problem of undue influence of certain measurement scales. In addition, this conditional independence structure means that the kernel parameters \( \theta^{(j)}_h \)'s do not depend on the global cluster IDs. This induces a common atoms structure, in which subjects assigned to different global clusters can be assigned to the same local (variable-specific) clusters, inducing borrowing of information. With different motivation, an alternative common atom model (CAM) is proposed in Denti et al. (2021).

CLOG implies a product kernel structure, \( \prod_{j=1}^p p(y_{ij} \mid s_{ij}, \theta^{(j)}_h) \), which does not account for dependence between variables at the level of the kernel. In classical (global only) mixture models, product kernels can lead to over-clustering, and it is natural to wonder whether similar problems can arise for CLOG models? To assess this, we first consider a commonly used (global only) mixture
Figure 1.5: Illustration of the proposed local-global clustering model: scatter plots of a two-dimensional data set, with points colored by local cluster IDs $s_{i1}$, $s_{i2}$ and global cluster IDs $C_i$'s from left to right, respectively.

Figure 1.6: Graphical model representation of CLOG. The boxes are “plates” representing replicates. The outer plate represents observations, while the inner plate represents variables.
model for clustering

\[ f(y_i) = \sum_{\ell=1}^{\infty} \lambda_\ell p_\ell(y_i), \quad (1.5) \]

where each component represents a cluster. Consider, for now, a product kernel

\[ p_\ell(y_i) = \prod_{j=1}^{p} p(y_{ij}), \]

which assumes that the variables are independent conditional on the cluster or component label. Because variables are typically dependent within each “true” cluster, more than one component is necessary to approximate the within-cluster data distribution, leading to over-clustering. This is different in CLOG, as conditional independence of the variables is at the level of local clusters. If we integrate out the local cluster IDs in the joint likelihood of \((y_i, s_i \mid C_i = \ell)\), we obtain

\[ \sum_{h_1} \cdots \sum_{h_p} \left\{ \prod_{j=1}^{p} p(y_{ij} \mid s_{ij} = h_j, \theta_{h_j}) P(s_{ij} = h_j \mid C_i = \ell, \psi_{\ell, h_j}) \right\}, \]

which allows for dependence between variables within global clusters. In the global-only mixture model (1.5), inference on \(C_i\) depends on \(\lambda_\ell p_\ell(y_i)\), inducing large sensitivity to misspecification of the kernel \(p_\ell(y_i)\), especially when dimensionality is moderately high. In contrast, for CLOG, inference on the cluster assignment for observation \(i\) depends on the data only through \(\lambda_\ell\) and \(\psi_{\ell, s_{ij}}, j = 1, \ldots, p\), and is therefore less sensitive to model mis-specification. In addition, by building the model through univariate kernels, we bypass the difficult task of choosing and fitting multivariate kernels that can capture a variety of within-cluster data distributions. We will illustrate these advantages over global-only mixtures through examples later in the paper.

1.3 Properties of CLOG

1.3.1 Local-global borrowing of information

The nature of the local-global borrowing of information in CLOG is somewhat opaque based only on examining the form of (1.4). In this section, we carefully study how local clustering probabilities depend on global clustering and vice versa. In the important special case of choosing Dirichlet priors, we obtain closed form expressions that provide insights into the role of the hyperparameters
and factors impacting the marginal and conditional clustering behavior at both the local and global level. Relevant proofs are provided in Appendix 2.1.

**Lemma 1.** Under (1.4), for any \( i \) and \( i' \) with \( i \neq i' \), we have

\[
P(s_{ij} = s_{i'j} \mid C_i = C_{i'}) = \sum_{h=1}^{\infty} \mathbb{E} \left( \psi_{1,h}^{(j)} \right)^2 \quad \text{and} \quad P(s_{ij} = s_{i'j} \mid C_i \neq C_{i'}) = \sum_{h=1}^{\infty} \mathbb{E} \left( \psi_{1,h}^{(j)} \right) \mathbb{E} \left( \psi_{2,h}^{(j)} \right).
\]

Further, if \( \psi_{h}^{(j)} \sim \text{Dir}_{K_j}(\omega_1^{(j)}, \ldots, \omega_{K_j}^{(j)}) \) across \( j \), then for any \( i \) and \( i' \) with \( i \neq i' \),

\[
P(s_{ij} = s_{i'j} \mid C_i = C_{i'}) = \frac{\sum_{h=1}^{K_j} \omega_h^{(j)} + 1}{(\sum_{h=1}^{K_j} \omega_h^{(j)} + 1)(\sum_{h=1}^{K_j} \omega_h^{(j)})} \quad \text{and} \quad P(s_{ij} = s_{i'j} \mid C_i \neq C_{i'}) = \frac{\sum_{h=1}^{K_j} \omega_h^{(j)} + 1}{(\sum_{h=1}^{K_j} \omega_h^{(j)})^2}.
\]

**Proposition 3.** Under (1.4), for any variable \( j \), observations \( i \) and \( i' \) with \( i \neq i' \),

\[
P(s_{ij} = s_{i'j} \mid C_i = C_{i'}) = P(s_{ij} = s_{i'j} \mid C_i \neq C_{i'}) + \sum_{h=1}^{\infty} \text{Var}(\psi_{1,h}^{(j)}).
\]

Further, if \( \psi_{h}^{(j)} \sim \text{Dir}(\omega_1^{(j)}, \ldots, \omega_{K_j}^{(j)}) \), then we have

\[
P(s_{ij} = s_{i'j} \mid C_i = C_{i'}) = P(s_{ij} = s_{i'j} \mid C_i \neq C_{i'}) + \frac{\sum_{h=1}^{K_j} \omega_h^{(j)} \sum_{m \neq h} \omega_m^{(j)}}{(\sum_{h=1}^{K_j} \omega_h^{(j)})^2(\sum_{h=1}^{K_j} \omega_h^{(j)} + 1)}.
\]

Proposition 3 shows that clustering together two observation vectors globally increases the chance of the observations being clustered together locally for the \( j \)th variable by \( \sum_{h=1}^{\infty} \text{Var}(\psi_{1,h}^{(j)}) \).

If we put a Dirichlet prior, \( \text{Dir}(\omega_1^{(j)}, \ldots, \omega_{K_j}^{(j)}) \), on \( \psi_{h}^{(j)} \) and set \( \sum_{h=1}^{K_j} \omega_h^{(j)} = c \) to a constant \( c \), the increase of probability is maximized at \( \omega_1^{(j)} = \cdots = \omega_{K_j}^{(j)} = c/K_j \) for a fixed \( K_j \) and is minimized when \( \omega_1^{(j)} = \cdots = \omega_{K_j}^{(j)} \) is proportional to a standard basis vector, meaning that when the prior is highly concentrated on a component, the gain is relatively small. This is intuitive as such a probability vector suggests strong confidence that \( y_{ij} \) belongs to a certain local cluster; thus the knowledge of the two observation vectors being clustered together globally does not contribute much to the local clustering. If we let \( \omega_h^{(j)} = \beta_j \pi_h^{(j)} \), with \( \beta_j, \pi_h^{(j)} > 0 \), then

\[
\text{Var}(\psi_{1,h}^{(j)}) = \frac{(\pi_h^{(j)}/\sum_h \pi_h^{(j)})(1 - \pi_h^{(j)}/\sum_h \pi_h^{(j)})}{\beta_j \sum_h \pi_h^{(j)} + 1}.
\]

15
For fixed $K_j$, as $\beta_j$ increases, the gain decreases and vice versa. The property of this prior is further discussed after the next proposition, which shows a similar result in the opposite direction.

**Proposition 4.** For any variable $j$ and observations $i$ and $i'$ ($i \neq i'$), we have

$$P(C_i = C_{i'} < P(C_i = C_{i'} | s_{ij} = s_{i'j}) < P(C_i = C_{i'} | s_{ij} = s_{i'j}, s_{ij'} = s_{i'j'})$$

Proposition 4 shows that local clustering is also informative of global clustering. Further, consider the prior $\psi^{(j)}_i \sim \text{Dir}(\beta_j \pi^{(j)}_1, \ldots, \beta_j \pi^{(j)}_{K_j})$ discussed after Proposition 3. We can derive the role of $\beta_j$ in enabling the information flow from the local to the global clustering layer. By Bayes’ theorem,

$$P(C_i = C_{i'} | s_{ij} = s_{i'j}) = \frac{\sum_{h=1}^{K_j} \mathbb{E}(\psi^{(j)}_{1,h}^2)P(C_i = C_{i'})}{\sum_{h=1}^{K_j} \mathbb{E}(\psi^{(j)}_{1,h}^2)P(C_i = C_{i'}) + \sum_{h=1}^{K_j} \mathbb{E}(\psi^{(j)}_{1,h})\mathbb{E}(\psi^{(j)}_{2,h})P(C_i \neq C_{i'})}$$

where the only quantity that depends on $\beta_j$ is $\mathbb{E}(\psi^{(j)}_{1,h}) = \frac{(\pi^{(j)}_h / \sum_h \pi^{(j)}_h) (1 - \pi^{(j)}_h / \sum_h \pi^{(j)}_h) + \pi^{(j)}_h}{\beta_j \sum_h \pi^{(j)}_h + 1}$, meaning $P(C_i = C_{i'} | s_{ij} = s_{i'j})$ is a monotonically decreasing function in $\beta_j$. This suggests that $\beta_j$ is an effective parameter for controlling the information flow between the local and global clustering layers.

We can generalize Proposition 4 to the following result:

**Corollary 1.** For any variable $j$ and observation $i$ and $i'$ ($i \neq i'$), we have

$$P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in \mathcal{S}) < P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in \mathcal{S}')$$

where $\mathcal{S} \subseteq \mathcal{S}' \subseteq \{p\}$.

Corollary 1 shows that for any two observations, the more dimensions that are clustered together, the more likely they will be assigned to the same global cluster. This is an intuitively appealing property.
1.3.2 Handling missing data

Missing data are common in applications. In this section, we discuss how CLOG is able to easily handle such cases in a coherent way. We assume that data are missing at random (MAR). Formally, let the inclusion indicator \( I \) be a matrix of the same size as \( Y \), with each element of \( I \) equal to 1 if the corresponding component of \( Y \) is observed and 0 if it is missing. Write \( Y \) as a block matrix \((Y_{\text{obs}}, Y_{\text{mis}})\), where \( Y_{\text{obs}} \) denotes the observed values and \( Y_{\text{mis}} \) denotes the missing values and decompose \( y_i \) as \((y_{i,\text{obs}}, y_{i,\text{mis}})\), where \( y_{i,\text{obs}} \) and \( y_{i,\text{mis}} \) denote the observed and missing values of the \( i \)th observation. Denote the set of all model parameters by \( \xi \), and the parameters governing the missingness \( I \) by \( \eta \). Then the joint distribution of \((Y, I)\), given parameters \((\xi, \eta)\), can be written as

\[
p(Y, I \mid \xi, \eta) = p(Y \mid \xi)p(I \mid Y, \eta)
\]

The assumption of MAR allows us to have

\[
p(Y_{\text{obs}}, I \mid \xi, \eta) = p(I \mid Y_{\text{obs}}, \eta) \int p(Y_{\text{obs}}, Y_{\text{mis}} \mid \xi) dY_{\text{mis}},
\]

where

\[
\int p(Y_{\text{obs}}, Y_{\text{mis}} \mid \xi) dY_{\text{mis}} = \prod_{i=1}^{n} p(y_{i,\text{obs}}, y_{i,\text{mis}} \mid \xi) dy_{1,\text{mis}} \cdots dy_{n,\text{mis}} = \prod_{i=1}^{n} p(y_{i,\text{obs}} \mid \xi)
\]

when the data are iid conditional on \( \xi \). Therefore, Bayesian inference on \( \xi \) can be obtained by only considering the observed data likelihood \( \prod_{i=1}^{n} p(y_{i,\text{obs}} \mid \xi) \), as long as \( \xi \) and \( \eta \) are \textit{a priori} independent. Unfortunately, \( \prod_{i=1}^{n} p(y_{i,\text{obs}} \mid \xi) \) does not admit a closed form in many Bayesian models, requiring approximations or missing data imputation. This either introduces approximation errors or makes computation much more expensive. CLOG, however, admits a closed form due to the independent modeling of each variable at the kernel level, as we show in the next proposition.

**Proposition 5.** \textit{CLOG admits a closed form of}

\[
p(y_{i,\text{obs}} \mid \xi) = \sum_{\ell} \lambda_{\ell} \prod_{j \in \{m: I_{im} = 1\}} \sum_{h=1}^{\infty} \psi^{(j)}_{\ell,h} f_{\theta_{ij}}^{(j)}(y_{ij}).
\]
Proposition 5 suggests that we do not need to impute missing data for inference on $\xi$, which makes the computation dramatically more efficient in some cases.

1.4 Properties for a canonical case

From this section onward, we focus on a simple Dirichlet over-fitted local-global mixture as a canonical special case. This model is practically very useful, representing a local-global extension of the widely popular (global only) over-fitted mixtures of Rousseau and Mengersen (2011) (henceforth RM). Over-fitted mixtures choose an upper bound on the number of clusters, and then remove unnecessary clusters via a shrinkage prior (typically Dirichlet with a small precision parameter).

Starting with (1.4), we choose the following priors to define our OF-CLOG model:

$$\lambda = (\lambda_1, \ldots, \lambda_K) \sim \text{Dir}_K(\alpha, \ldots, \alpha)$$

$$\psi_\ell = (\psi_\ell, \ldots, \psi_\ell, K_j) \sim \text{Dir}_{K_j}(\omega_1, \ldots, \omega_{K_j}),$$

where $K$ is an upper bound of the number of clusters, $\ell = 1, \ldots, K$ and $j = 1, \ldots, p$. In this section, we show posterior consistency and variable selection properties.

1.4.1 Asymptotic behaviour of posterior

RM proved that, when the true data-generating model is a (global) mixture of $K_0$ components with $K_0 \leq K$, the posterior of an over-fitted mixture model asymptotically empties the extra $(K_0 + 1, \ldots, K)$ components by concentrating the corresponding $\lambda_{K_0+1}, \ldots, \lambda_K$ values close to zero.

Our OF-CLOG model implies the following form for the density of the data:

$$f_\xi(y) = \sum_{\ell=1}^{K} \lambda_\ell g_{\zeta_\ell}(y), \quad K \geq 1, \quad \zeta_\ell \in \Gamma, \quad \Gamma \in \mathbb{R}^d. \quad (1.7)$$

where $g_{\zeta_\ell}(y) = \sum_{k=1}^{H} \rho_{\ell k} h_{\varphi_k}(y)$ is the inner mixture, $H = \prod_{j=1}^{p} K_j$ is the number of kernels in the inner mixture, $\xi = (\lambda_1, \ldots, \lambda_K, \zeta_1, \ldots, \zeta_K) \in \Xi_K$ is the vector of all parameters, and $\zeta_\ell = (\rho_{\ell 1}, \ldots, \rho_{\ell H}, \varphi_1, \ldots, \varphi_H)$ are the parameters for the $\ell$-th component of the outer mixture,
where \( \vartheta_k \in \Theta, \zeta_\ell \in V_H \times \Theta^H = \Gamma \), and \( V_H \subset \mathbb{R}^H \) is the \((H-1)\)-dimensional probability simplex.

The component distribution \( h_{\vartheta_k}(y) \) of the inner mixture is a product of categorical and Gaussian kernels, with the former kernel associated with nominal and ordinal variables and the latter with quantitative variables. We assume that the model is correctly specified; data \( Y^n = (Y_1, \ldots, Y_n) \) are independently and identically distributed from a mixture

\[
f_0(y) = \sum_{\ell=1}^{K_0} \lambda^0_{\ell} g_{\zeta^0_\ell}(y),
\]

where \( K_0 \) is the true number of components, \( g_{\zeta^0_\ell}(y) = \sum_{k=1}^{H} \rho^0_{\ell k} h_{\vartheta_0_k}(y), \vartheta^0_k \in \Theta, \zeta^0_\ell \in \Gamma \). Let \( \mathcal{S}_K \) be the set of all permutations \( \sigma \) of \( \{1, \ldots, K\} \). We denote the posterior distribution conditional on \( Y^n \) by \( P^\pi(\cdot | Y^n) \). Formally, we have the following theorem.

**Theorem 1** (Excess components empty asymptotically). Suppose that

1. The kernels of the inner mixture \( g_{\zeta}(y) \) satisfy Assumptions 1-5 of RM,
2. The number of kernels in the inner mixture \( H \) is equal to the true number \( H^0_\ell \) for all \( \ell \),
3. The parameters \( \vartheta_k \) have sub-exponential priors for all \( k = 1, \ldots, H \).

Then if \( \alpha < d/2 \) and \( \rho = \{dK_0 + K_0 - 1 + \alpha(K - K_0)\} / (d/2 - \alpha) \) we have

\[
\lim_{M \to \infty} \limsup_n \left( \mathbb{P}^n \left[ \min_{\sigma \in \mathcal{S}_K} \left( \sum_{\ell=K_0+1}^{K} \lambda_{\sigma(\ell)} \right) > M n^{-1/2} \log(n)^{(1+\rho)/2} \right] \right) = 0.
\]

(1.9)

The conclusion in (1.9) is a statement about the limit of the expected posterior probability of a certain set under repeated sampling from (1.8) as the sample size goes to infinity. The set of interest is the event that the sum of the \( K - K_0 \) smallest values of \( \lambda \) exceeding a certain threshold, which is achieved by considering all permutations of the entries of \( \lambda \). The permutation is necessary because the result of RM gives no guarantee that the components that empty will be the last \( K - K_0 \) positions. The threshold is, up to logarithmic factors, the usual posterior concentration rate of \( n^{-1/2} \). The result gives us confidence that for suitable choice of the prior parameter \( \alpha \), we should see the mixture (1.7) approaches a mixture with only \( K_0 \) non-zero weights. In practice, this means

19
that we could learn the number of true clusters even if we select $K$ larger than $K_0$ by thresholding small values of $\lambda$.

The major limitation of the result is that we require $H_\ell^0 = H$, which means that while we allow the outer mixture to be overfitted ($K > K_0$) we require the inner mixture to be specified with the correct number of mixture components. We are not claiming that the result fails to hold when $H \neq H_\ell^0$, instead we are acknowledging that proof does not apply.

The proof of Theorem 1 is deferred to Appendix 3. RM established the posterior asymptotics under five assumptions, which are reproduced in Appendix 3 along with the verification that they hold for OF-CLOG. Assumption 4 does not apply directly to our situation because our kernel has a mixture structure. We therefore propose a small extension of Assumption 4 that applies to our model and show that the proof of RM’s Theorem 1 is valid with this assumption. While most of the assumptions are technical, Assumption 1 may be of independent interest. We state the result below as a theorem.

**Theorem 2** ($L_1$ consistency). There exists $\delta_n \leq \log(n)^q / \sqrt{n}$ for some $q \geq 0$ such that

$$\lim_{M \to \infty} \limsup_{n} \left[ \mathbb{E}_n \{ P^\pi (\| f_0 - f_\xi \| \geq M \delta_n \mid Y^n) \} \right] = 0.$$ 

Theorem 2 tells us that the posterior distribution on the density (in contrast with the posterior on the parameters) concentrates around the true density $f_0$ in $L_1$. The details of the proof, which is based on Theorem 4.1 of Fruhwirth-Schnatter et al. (2019), can be found in Appendix 3.

Together, Theorems 1 and 2 establish that the posterior distribution of OF-CLOG is reasonably well behaved.

### 1.5 Posterior computation

We describe a simple Gibbs sampler for posterior computation under our proposed OF-CLOG model, focusing on a specific default choice of kernel for each data type. We have found this Gibbs sampler to perform well in a variety of applications.
1.5.1 Kernel specification

As a simple default choice, we focus on Gaussian kernels for continuous variables and flexible discrete kernels for nominal categorical variables. Marginalizing out $s_{ij}$,

$$y_{ij} \mid C_i = \ell, \psi^{(j)}_{\ell,h}, \theta^{(j)}_h \sim \sum_{h=1}^{K_i} P(s_{ij} = h \mid C_i = \ell, \psi^{(j)}_{\ell,h}) f(y_{ij} \mid s_{ij} = h, \theta^{(j)}_h) = \sum_{h=1}^{K_i} \psi^{(j)}_{\ell,h} f^u_{\theta^{(j)}_h},$$

where $u \in \{q, c\}$ indexes quantitative ($q$) and categorical ($c$) kernels chosen as:

For ordered categorical and count data, it is important to preserve order information in the levels. Poisson kernels are a poor choice due to limited flexibility in having a single parameter for the mean and variance. Canale and Dunson (2011) proposed a rounded Gaussian kernel approach as an alternative. Applying their approach, for ordinal and count observed variables, we let $y_{ij} = g(y^*_{ij})$ with

$$y^*_{ij} \mid s_{ij} = h, \theta^{(j)}_h \sim \begin{cases} N(y^*_{ij}; \mu_h^{(j)}, \sigma_h^{(j)}), & j \in I_q, \\ \text{Categorical}(y^*_{ij}; \{1, \ldots, d_j\}, \gamma_1^{(j)}, \ldots, \gamma_{d_j}^{(j)}), & j \in I_c. \end{cases}$$

The hyperprior specification for the prior parameters in the kernels and specification of thresholds $\{z_j\}$ are discussed in the next section.

1.5.2 Prior specification

We now provide a description of the prior specification for OF-CLOG that we will use in our simulation experiments and data analyses in the remainder of the paper. For the probabilities on each of the global mixture components, we follow common practice for over-fitted Dirichlet mixtures and let $\lambda \sim \text{Dir}_K(1/K, \ldots, 1/K)$, which satisfies the conditions in Theorem 1. For each of the local
component weights $\psi^{(j)}_k$ we use a symmetric Dirichlet prior, with $\omega^{(j)} = (\beta/K_j, \ldots, \beta/K_j)$. We put a hyperprior on $\beta$ in case of sensitivity issue or simply let $\beta = 1$ as a default choice. Setting $K_j$ can be challenging. For unordered categorical variables, it is not clear how to define similarity between levels and group them. We therefore set $K_j = d_j$ for $j \in I_c$. For ordered categorical variables, we let $K_j = \max(4, d_j)$ as a default choice. For quantitative variables, we run K-means with different number of clusters and choose the optimal one based on the silhouette and the elbow method. For sensitivity analysis, we repeat the analysis by setting $K_j, j \notin I_c$ at different values and calculate the WAIC.

Next we consider hyperpriors on kernel parameters $\{\mu^{(j)}_h, \sigma^{(j)2}_h, \gamma^{(j)}_h, \nu^{(j)}_h, \tau^{(j)2}_h\}$. We choose a normal-inverse-Gamma prior for $(\mu^{(j)}_h, \sigma^{(j)2}_h)$, since these parameters are scalars and a simple and computationally convenient prior suffices. A natural conjugate prior for $\gamma^{(j)}_h$ is a Dirichlet distribution. However, because we expect an unordered categorical variable from local cluster $h$ to be more likely to take value $h$, we set the prior to be

$$\gamma^{(j)}_h \sim (1 - \epsilon)e_h + \epsilon\text{Dir}(1, \ldots, 1),$$

$$\epsilon \sim \text{Beta}(a_\epsilon, b_\epsilon),$$

where $e_h$ is the $h$th standard basis vector, with hyperparameters $a_\epsilon$ and $b_\epsilon$ chosen such that $\epsilon$ is concentrated near 0 apriori. This prior also has a simple posterior update.

The choice of a hyperprior for $(\nu^{(j)}_h, \tau^{(j)2}_h)$ is less straightforward due to the non-identifiability. With the underlying continuous variable $y^{*}_{ij}$ unobserved, there can be infinitely many combinations of $(y^{*}_{ij}, \nu^{(j)}_h, \tau^{(j)2}_h)$ that induce the same kernel on $y_{ij}$ for a given set of thresholds $\{z_k\}$. In addition, the free updates of $\tau^{(j)2}_h$ often leads to highly imbalanced component variances across $h$’s and unreasonable local cluster assignment that is largely influenced by the number of observations at each level $h$. Specifically, when a level contains many observations, the observations tend to be assigned to a component with a large variance or be split into multiple components. To ameliorate these issues, we fix $\tau^{(j)}_h$ at 1 and set $z_k = 3(k - 1)$ such that the cut-offs are far enough that observations from the same level all tend to be assigned to one or two local clusters. We also note that while Canale and Dunson (2011) chose $z_k = k - 1$ and allowed free updates of other parameters, their goal was very different than ours—their goal was to flexibly model count variables
whereas we aim to produce a meaningful partitioning of observations associated with each variable.

To summarize, the prior specification is as follows:

\[ \lambda \sim \text{Dir}_K(\alpha, \ldots, \alpha), \]

\[ \psi_{\ell}^{(j)} \sim \text{Dir}_{K \ell}(\beta/K_{\ell}, \ldots, \beta/K_{\ell}) \]

\[ (\mu_{h}^{(j)}, \sigma_{h}^{(j)^2}) \sim N(\mu_{0h}^{(j)}, \sigma_{h}^{(j)^2}V_0)\text{InvGamma}(a_0, b_0), \]

\[ \nu_{h}^{(j)} \sim N(\nu_{0h}, U_0), \]

\[ \gamma_{h}^{(j)} := (\gamma_{1h}^{(j)}, \ldots, \gamma_{dh,h}^{(j)}) \sim (1 - \epsilon)e_h + \epsilon\text{Dir}(1, \ldots, 1), \]

\[ \epsilon \sim \text{Beta}(a_\epsilon, b_\epsilon). \]

In our experiments, we choose \( \alpha = 1/K, K_{\ell} = d_{\ell}, \beta = 1. \) Other parameters are set at \( \nu_{0h} = (z_h + z_{h+1})/2, \mu_0^{(j)} = \bar{y}_j, a_\epsilon = 1, b_\epsilon = 100. \) Other parameters are \( z_0 = -\infty, z_{d_{\ell}+1} = \infty, z_k = 3(k - 1) \) for \( k \notin \{0, d_{\ell} + 1\}. \)

### 1.5.3 Posterior sampling

For posterior computation we use a Gibbs sampler defined by the following steps. Let \( I_{h}^{(j)} \sim \text{Ber}(\epsilon). \) That is, \( I_{h}^{(j)} = 1 \) when \( \gamma_{h}^{(j)} \) takes the component with kernel \( \text{Dir}(1, \ldots, 1) \) and 0 otherwise.

Let \( n_{hj} = \sum_i 1(s_{ij} = h). \) Write \( N(\mu; m, \sigma^2V)\text{InvGamma}(\sigma^2; a, b) \) as \( \text{NIvg}(\mu, \sigma^2; m, V, a, b). \)

1. Update \( \lambda \) from the full conditional posterior distribution obtained in updating the Dirichlet prior with the likelihood:

\[ \text{Dir} \left( \alpha + \sum_i 1(C_i = 1), \ldots, \alpha + \sum_i 1(C_i = K) \right) \]

2. For \( \ell = 1, \ldots, K, \) update \( \psi_{\ell}^{(j)} \) from the full conditional posterior distribution obtained in updating the Dirichlet prior with the likelihood of the \( j \)th response for subjects in component/cluster \( \ell \) and local cluster \( h, \)

\[ \text{Dir} \left( \beta\pi_{1}^{(j)} + \sum_{i:C_i=\ell} 1(s_{ij} = 1), \ldots, \beta\pi_{K_j}^{(j)} + \sum_{i:C_i=\ell} 1(s_{ij} = K_j) \right). \]
3. Update \((\mu_h^{(j)}, \omega_h^{(j)^2})\) from the full conditional posterior distribution obtained in updating the Normal-Inverse-Gamma prior with the likelihood of the \(j\)th variable for subjects in local component/cluster \(h\), where \(j \in I_q\),

\[
\text{NInvG}(m_h^{(j)}, V_h^{(j)}, a_h^{(j)}, b_h^{(j)}),
\]

where

\[
m_h^{(j)} = \frac{V_0^{-1} \mu_0^{(j)} + n_{hj} \bar{y}_{hj}}{V_0^{-1} + n_{hj}}, \quad V_h^{(j)-1} = V_0^{-1} + n_{hj},
\]
\[
a_h^{(j)} = a_0 + n_{hj}/2, \quad b_h^{(j)} = b_0 + \sum_{i:s_{ij}=h} (y_{ij} - \bar{y}_{hj})^2/2 + \frac{V_0^{-1} n_{hj}}{2(V_0^{-1} + n_{hj})} (\mu_0^{(j)} - \bar{y}_{hj})^2.
\]

4. Update \(\gamma_h^{(j)}\) from the full conditional posterior distribution obtained in updating the Dirichlet mixture prior with the likelihood of the variable \(j \in I_c\) for subjects in local cluster \(h\),

\[
\gamma_h^{(j)} \sim \begin{cases} 
\text{Dir} \left( 1 + \sum_{i:s_{ij}=h} 1(y_{ij} = 1), \ldots, 1 + \sum_{i:s_{ij}=h} 1(y_{ij} = d_j) \right), & I_h^{(j)} = 1 \\
\epsilon_h, & \text{otherwise}.
\end{cases}
\]

5. Update \(y_{ij}^*\) from its full conditional posterior distribution \(N(\nu_{s_{ij}^*}, \tau_{s_{ij}^*}^2)\) for subjects in local cluster \(h\).

6. Update \(C_i\) from the full conditional posterior distribution obtained in updating the Dirichlet prior with the likelihood of local cluster assignment associated with subject \(i\):

\[
P(C_i = \ell \mid -) \propto \lambda_\ell \prod_{j=1}^p \psi_{\ell, s_{ij}}^{(j)}.
\]

7. Update \(s_{ij}\) from the full conditional posterior distribution obtained in updating \(P(s_{ij} = h \mid C_i = \ell, \psi_{\ell, h}^{(j)})\) with the likelihood of the \(j\)th variable for subject \(i\):

\[
P(s_{ij} = h \mid -) \propto \begin{cases} 
N(y_{ij}; \mu_h^{(j)}, \sigma_h^{(j)^2}) \psi_{\ell, h}^{(j)}, & j \in I_q \\
\psi_{\ell, h}^{(j)} \psi_{\ell, h}^{(j)} \gamma_{y_{ij}^*}^{(j)}, & j \in I_c \\
P(s_{ij} = h \mid C_i = \ell, y_{ij}^*) \propto \psi_{\ell, h}^{(j)} N(y_{ij}^*; \nu_h^{(j)}, \tau_h^{(j)^2}), & j \in I_o.
\end{cases}
\]

8. Update \(\epsilon\) from the full conditional posterior distribution obtained in updating the Beta prior with the likelihood of \(\{I_h^{(j)}\}\),

\[
\text{Beta} \left( 1 + \sum_{j \in I_o} K_j - \sum_{j \in I_c} \sum_{h=1}^{K_j} I_h^{(j)}, 100 + \sum_{j \in I_c} \sum_{h=1}^{K_j} I_h^{(j)} \right).
\]
9. Update $I_h^{(j)}$ from the full conditional posterior distribution obtained in updating the Bernoulli prior with the likelihood of the variable $j$, $j \in I_c$, for subjects in local cluster $h$,

$$
P(I_h^{(j)} = 1 \mid -) \propto \frac{\epsilon}{d_j^{(i:s_{ij}=h)}} \quad P(I_h^{(j)} = 0 \mid -) \propto (1 - \epsilon) \prod_{i:s_{ij}=h} e_{hk_i},$$

where $k_i$ indicates the level observed for subject $i$. The derivation for step 9 can be found in the Appendix.

1.6 Simulation study

We perform a simulation study to analyze the performance of CLOG for clustering mixed-scale data. The code is available on the GitHub Page of the first author. We compare with frequentist clustering methods including (a) K-means, (b) K-prototypes (Huang, 1998), which integrates K-means for quantitative and K-modes for categorical variables and is implemented in R package clustMixType (Szepannek, 2018), (c) Partitioning around Medoids (PAM) with Gower’s dissimilarity measure implemented the R package cluster (Maechler et al., 2022), (d) tandem, a two-stage approach that first performs principal component analysis for dimension reduction and then clusters using K-means, (e) reduced K-means clustering and (f) factorial K-means clustering. Methods (e) and (f) perform joint dimension reduction and cluster analysis. Methods (d), (e) and (f) are implemented in R package clustrd (Markos et al., 2019). We also compare with Bayesian methods including (g) mixture of Gaussian with ordinal variables having a latent continuous structure, (h) BNPmixclust, a multivariate Pitman-Yor process mixture model with Gaussian kernels that jointly models a set of latent variables for the mixed-scale variables (Carmona et al., 2019), implemented in the R package BNPmixcluster (Carmona, 2020) and (i) BNPclust, a multivariate Pitman-Yor process mixture model with Gaussian kernels implemented in the R package BNPmix (Corradin et al., 2021). These methods are henceforth abbreviated as (a) Kmeans, (b) Kprototype, (c) PAM, (d) tandem, (e) mixedRKM, (f) mixedFKM, (g) mixtureGaussian, (h) BNPmixclust and (i) BNPclust.

Kprototype, tandem, mixedRKM and mixedFKM are designed for data with quantitative and nominal variables only. PAM and BNPmixclust are designed for data with quantitative, nominal
and ordinal variables. While K-means and BNPclust are not designed for data with mixed-scale variables, we include them to assess whether it is necessary practically to specifically account for mixed-scale data.

To test the accuracy of the estimated clustering relative to the true clustering, we compute the adjusted Rand index (Hubert and Arabie, 1985). We also compute a KL divergence statistic for CLOG to demonstrate that CLOG is able to select the relevant variables for clustering. To determine whether variable \( j \) is important for the clustering result, we can compare \( P(C_i = \ell \mid Y, \lambda, \psi, s) \) to \( P(C_i = \ell \mid Y_{-j}, \lambda, \psi, s) \), where \( Y_{-j} \) is the data with variable \( j \) removed. If variable \( j \) does not influence the clustering, we expect these probabilities to be equal for all \( \ell \). The full conditional update for \( C_i \) follows

\[
P(C_i = \ell \mid Y, \lambda, \psi, s) \propto \lambda \prod_{j=1}^{p} \psi^{(j)}_{\ell s_{ij}}
\]

and consequently

\[
P(C_i = \ell \mid Y_{-j}, \lambda, \psi, s) \propto \lambda \prod_{j \neq j} \psi^{(j)}_{\ell s_{ij}}.
\]

Given values of \( \lambda, \psi \) and \( s \) we can then construct the probability vectors

\[
Q(i, \lambda, \psi, s) := (P(C_i = 1 \mid Y, \lambda, \psi, s), \ldots, P(C_i = K \mid Y, \lambda, \psi, s))^T,
\]

\[
Q_{-j}(i, \lambda, \psi, s) := (P(C_i = 1 \mid Y_{-j}, \lambda, \psi, s), \ldots, P(C_i = K \mid Y_{-j}, \lambda, \psi, s))^T.
\]

At each iteration of an MCMC, we can calculate the KL divergence between \( Q(i, \lambda, \psi, s) \) and \( Q_{-j}(i, \lambda, \psi, s) \); summing this statistic for all observations gives

\[
V(j, \lambda, \psi, s) = \sum_{i=1}^{N} KL(Q(i, \lambda, \psi, s), Q_{-j}(i, \lambda, \psi, s)),
\]

which has an interpretation as the KL divergence between the \((C_1, \ldots, C_n) \mid Y, \lambda, \psi, s \) and \((C_1, \ldots, C_n) \mid Y_{-j}, \lambda, \psi, s \) due to the conditional independence. By comparing the posterior distributions of \( V(j, \lambda, \psi, s) \) for different variables \( j \), we can measure their importance in determining the global clustering.

We generated the data sets using the R package `clusterGeneration` (Qiu and Joe, 2020). The package assumes each cluster follows a Gaussian distribution; we replaced the Gaussian distribution assumption with a multivariate skew normal so as to increase the difficulty of clustering.
The generating function allows seven inputs—number of clusters $K_{true}$, separation index between a cluster and its nearest neighboring cluster, number of non-noisy variables, number of noisy variables, the range of cluster sizes, whether to randomly rotate data in non-noisy dimensions, and range of scale parameters. A variable is considered “noisy” if its marginal distribution does not vary across clusters. The separation index can take any value in $[-1, 1)$; and the closer it is to 1, the more separated the clusters are. We set the separation index at 0.3, which corresponds to a separated though not necessarily well-separated cluster structure. We set $K_{true} = 3$, use $[1, 20]$ as the range of scale parameters and $[150, 300]$ as the range of cluster sizes. We also set the total number of variables to be 5. The shape parameters in the skew normal kernels are randomly generated from $\text{Unif}(-1000, 1000)$. To produce ordinal and nominal variables, we discretize quantitative variables to have the desired numbers of levels.

We produce data sets under four configurations as follows. (a) All quantitative: all five variables are non-noisy quantitative variables. (b) Ordinal noise: there are two quantitative, two nominal and one ordinal variables, among which the ordinal variable is noise. The ordinal and nominal variables have 5, 3, and 2 levels respectively. (c) Quantitative noise: there are two quantitative and three ordinal variables; both quantitative variables are noise. The ordinal variables have 2, 3, and 4 levels respectively. (d) Rotated: Same setting as (b), but data are randomly rotated in non-noisy dimensions so that we may not detect the full cluster structure from pair-wise scatter plots of the variables. For each configuration, we perform 20 independent replications. Figure 1.7 shows a visualisation of an example data set under configuration (d).

When running cluster analysis, we set the number of clusters $K$ to be 3 or 6. Setting $K = 3 = K_{true}$ is meaningful because we are interested in knowing whether CLOG performs well when the number of clusters is known. For the Bayesian methods (BNPmixclust, BNPclust and our method CLOG), we run samplers for 3000 iterations and discard the first half as burn in. Using the last 1500 iterations we identify a clustering estimate using the variation of information loss (Wade and Ghahramani, 2018). We use the default prior specification for all three methods. In particular, prior elicitation of CLOG follows the specification introduced in Section 1.5.2.

Figure 1.8 reports the distributions of the 20 replicates of the adjusted Rand index. Our pro-
Figure 1.7: A generalized pairs plot of a data set generated under configuration (d). The labels quant., ord. and nom. refer to quantitative, ordinal and nominal variables respectively.
posed CLOG is superior in each scenario obtaining high adjusted Rand indices, even when the number of clusters is misspecified. Methods that rely on the specification of the number of clusters, such as Kmeans and tandem, are sensitive to the misspecification. In addition, mixtureGaussian, which tends to give higher weights to quantitative variables in clustering, performs poorly when quantitative variables are noise. BNPmixclust consistently performs very poorly, even though it is one of the few methods specifically designed for data with quantitative, nominal and ordinal variables. It also experiences the longest run time among all methods. BNPclust fails to run for some mixed-scale data sets, because the categorical variables only take a few values and this can result in not positive definite covariance matrices.

Figure 1.9 sheds light on the robust performance of CLOG in the presence of noise variables. The posterior distribution of the KL divergence statistic shows that CLOG is able to identify the noise and signal variables for clustering. Some other methods under comparison, such as PAM and BNPclust, do not have such variable selection property, which partly explains their poor performance in the presence of noise variables.
Figure 1.8: Simulation results: distributions of the adjusted Rand indices in 20 replicated experiments. Note that box plots associated with Kprototype is missing in the first row because the method is only applicable to mixed-scale data.
Figure 1.9: Simulation results of CLOG for a randomly selected Quant noise data set: histograms of posterior draws of the KL statistic for a signal and noise variable.
1.7 Applications

In this section, we analyze two data sets to illustrate the proposed method. We compare with the same methods mentioned in Section 1.6. For the Bayesian methods, we also follow the same practice as that in Section 1.6— we use the default priors, run the samplers for 3000 iterations, discard the first half as burn in and use the variation of information loss to identify a clustering estimate.

1.7.1 Diamond data

The data set is from the R package clustrd, and contains information on 308 round diamonds sold in Singapore (Markos et al., 2019). Diamond features include price, caret size (or size), color, clarity and certification body (or certification), among which there are 3 quantitative, 2 ordinal and 1 nominal variables. We cluster the diamonds using the latter four variables and compare the results to price to see if the clustering maps onto price in a reasonable way, as there are no available true cluster labels for the data set and we expect the price to be highly dependent on the other variables (Khan, 2022). For example, the price of a diamond tends to increase with the caret size; under the same diamond condition, GIA tends to give a grading that results in a higher price than that of IGI (Diamond Screener, 2022). After clustering the original data, we add two noise variables, 1 quantitative and 1 ordinal with three levels, to see whether the methods are able to eliminate the noise information for clustering. The two variables are created by generating from a skew-normal distribution \( SN(0, 1, 10) \), followed by discretization into 3 levels if applicable.

For both the cluster analyses, we set \( K = 3 \) for all methods. For the analysis of the original data, the package functions of BNPmixclust and BNPclust experienced numerical issues under the default prior setting and they are therefore excluded from the comparison. Figure 1.10 visualizes the clustering results in the dimensions of certification and size. Due to space constraints, the figure only includes a subset of the methods (i.e. CLOG, Kmeans, mixtureGaussian and PAM) and the remaining results can be found in Appendix .5. Both Kmeans and mixtureGaussian treat
certification body as a quantitative variable and unfortunately produced undesirable results. For mixtureGaussian, certification body dominates the partition structure and there is no separation in the dimension of caret size; for Kmeans, there is no clear separation in either certification or size. While PAM generated well-separated clusters, they are purely driven by the differences in certification; this is not surprising since PAM tends to prioritize nominal variables in clustering as we note in Section 1.1. CLOG produced a reasonable result, with three clusters representing small to medium IGI certified diamonds, medium to large non-IGI certified diamonds, and small GIA certified diamonds respectively. We also include clustering results in the dimensions of clarity and size in Appendix 5.

Figure 1.11 illustrates the results of the diamond data with added noise variables. The posterior distributions of the KL statistic for size and one noise variable shows that CLOG is able to exclude the redundant information for clustering. Kprototype fails to do so, clustering on the noise dimension, as is shown in the figure.

1.7.2 Byar prostate cancer data

The data set was introduced in Section 1.1. It is available in the R package clustMD and was originally from a randomized clinical trial of patients with prostatic cancer in either stage 3 or 4 of the disease (Byar and Greene, 1980). According to Byar and Greene (1980), stage 3 represents local extension of the disease without evidence of distant metastasis, while stage 4 represents distant metastasis as evidenced by elevated acid phosphatase, X-ray evidence, or both. We evaluate the clustering performance based on the classification of clinical stages. The data set contains 12 pre-trial covariates and 1 post-trial covariate, with the post-trial one being the survival status. We exclude it as it is impossible to know its value when determining the clinical stage. The 12 pre-trial covariates are age, weight, performance rating, cardiovascular disease history, systolic blood pressure, diastolic blood pressure, electrocardiogram code, serum haemoglobin, size of primary tumour, index of tumour stage and histologic grade, serum prostatic acid phosphatase and bone metastases; the list contains 8 quantitative, 1 ordinal and 3 nominal variables. We note that the number of vari-
Figure 1.10: Clustering results of the diamond data of a subset of the methods: scatter plots of price vs size, with shapes and colors of the points indicating certification and cluster estimates.
Figure 1.11: Clustering results of the diamond data with added noise variables: (left) histograms of posterior draws of the KL statistic for size and a noise variable and (right) scatter plots of price vs a noise variable for CLOG and Kprototype.

Because the outcome is binary, we use area under curve (AUC) besides ARI to evaluate clustering performance; see Table 1.1 for the results. Based on the AUCs, estimates by K-means, K-prototype, PAM, mixed FKM and BNPclust are almost equivalent to random guesses of the labels. CLOG performs the best among all methods and in fact much better than the second best method mixed RKM. One of the advantages of CLOG is uncertainty quantification. Its 95% credible intervals of ARI and AUC are (0.43, 0.55) and (0.81, 0.86).

Figure 1.12 shows that CLOG characterizes the stage-3 patients with lower levels of serum prostatic acid phosphatase and absence of bone metastatis, which is consistent with the medical definition. While PAM is a method designed for mixed-scale data and tends to give larger weights to nominal variables in distance calculation, it is unable to identify bone metastases as an important clustering variable. Interestingly, consistent with the truth, CLOG identifies systolic blood pressure and serum haemoglobin as noise variables. In fact, Fig. 1.13 shows the posterior draws of the KL statistics associated with serum prostatic acid phosphatase
Table 1.1: The clustering performance of the Byar prostate cancer data*

<table>
<thead>
<tr>
<th>Method</th>
<th>ARI</th>
<th>AUC</th>
<th>Method</th>
<th>ARI</th>
<th>AUC</th>
<th>Method</th>
<th>ARI</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>0.01</td>
<td>0.51</td>
<td>tandem</td>
<td>0.23</td>
<td>0.71</td>
<td>mixtureGaussian</td>
<td>0.02</td>
<td>0.53</td>
</tr>
<tr>
<td>K-prototype</td>
<td>0.01</td>
<td>0.51</td>
<td>mixed RKM</td>
<td>0.24</td>
<td>0.71</td>
<td>BNPclust</td>
<td>0.00</td>
<td>0.5</td>
</tr>
<tr>
<td>PAM</td>
<td>0.00</td>
<td>0.52</td>
<td>mixed FKM</td>
<td>0.00</td>
<td>0.55</td>
<td>CLOG</td>
<td>0.53</td>
<td>0.86</td>
</tr>
</tbody>
</table>

*BNPmixclust is missing due to numerical issues of running the package function and serum haemoglobin, suggesting that the former variable has higher influence on the clustering outcome. However, methods such as PAM and Kprototype do not have such variable selection property, which partly explains their poor performance for this data set.
Figure 1.12: Results of the Byar prostatic acid phosphatase data: (left) the relationship between logged serum prostatic acid phosphatase levels (in King-Armstrong units) and primary tumour sizes, with the shape of the points indicating the presence of bone metastasis (circle: absence; triangle: presence); (right) the relationship between systolic blood pressure and serum haemoglobin levels, with the shape of the points indicating cardiovascular disease history status (circle: no; triangle: yes). All points are colored by prostate cancer stage or clustering estimates. The clustering estimates are recoded into 3 and 4 to match the cancer stages.
Figure 1.13: Results of CLOG for the Byar prostatic acid phosphatase data: the histogram of the posterior draws of the KL statistic associated with serum haemoglobin levels and serum prostatic acid phosphatase levels. The cluster labels are recoded into 3 and 4 to match the cancer stages.
1.8 Discussion

There is a lack of methods designed for mixed-scale data and many existing methods suffer from the problem of undue influence from variables of different measurement scales. Leveraging on Bayesian consensus clustering (Lock and Dunson, 2013) and Common Atoms model (Denti et al., 2021), we propose a local-global Bayesian clustering framework. We establish its desirable properties, such as local-global borrowing of information and the ease of handling missing data in the case of missing at random. As a canonical case, we focus on a simple Dirichlet over-fitted local-global mixture, abbreviated as OF-CLOG. We show that the posterior on the density concentrates on the true density asymptotically. In addition, RM proved that, when the true data-generating model is a (global) mixture of $K_0$ components with $K_0 \leq K$, the posterior of an over-fitted mixture model asymptotically empties the extra $(K_0 + 1, \ldots, K)$ components by concentrating the corresponding $\lambda_{K_0+1}, \ldots, \lambda_K$ values close to zero. We verify the assumptions of their theorem (proving that their theorem holds with a small extension for one assumption) to show that the result holds for OF-CLOG. This result allows practitioners to use the model with an upper bound on the number of clusters rather than fit the model with multiple values of $K$ and try to select a best fit. To facilitate the practical application of OF-CLOG, we propose kernels and prior distributions and show they lead to a simple Gibbs sampler. We compare OF-CLOG with a variety of methods and show its superior performance, including its ability to select relevant variables for clustering in the presence of a noise variable. This observation is pending a mathematical proof. In addition, While the model seems to perform well in simulations where $n$ grows while $n/p$ is fixed at 5, it is left for future work to explore the theory when both $n$ and $p$ grow to infinity simultaneously.
.1 Theory for motivating examples

We note that by the Woodbury inversion formula we have \((I + 11^T)^{-1} = I - 11^T/(1 + n)\) when the matrices are \(n\)-dimensional. The determinant of \(I + 11^T\) is \(n + 1\). For \(j = 1, 2\) we have

\[
\prod_{k=0}^{1} N_{njk}(X_{jk} \mid 0, I + 11^T) \propto \exp \left( -\frac{1}{2} \left[ X_{j0}^T(I - 11^T/(1 + n_{j0}))X_{j0} + X_{j1}^T(I - 11^T/(1 + n_{j1}))X_{j1} \right] \right) / \sqrt{(1 + n_{j0})(1 + n_{j1})}.
\]

**Lemma 2.** Suppose \((X_1, \ldots, X_n) \sim N(0, I_n + 11_n^T)\) is an \(n\)-dimensional Gaussian. Then \(\sum_i I(X_i \geq 0)\) has a uniform distribution of \(\{0, \ldots, n\}\). Let \(A\) be subset of \(\mathbb{R}^n\) which can be expressed as the Cartesian product of the half-spaces \(\mathbb{R}^+ = \{x : x \geq 0\}\) in some dimensions and \(\mathbb{R}^- = \{x : x < 0\}\) in the remaining dimensions. Then \(\mathbb{P}(X \in A) = (n + 1)^{-1}(\binom{n}{k})^{-1}\), where \(k\) is the number of positive half spaces.

**Proof of Proposition 2.** Consider the event that we observe at least one observation with \(x_{i2} = 1\) and one observation with \(x_{i2} = 0\), which occurs with probability \(1 - 2^{-n+1}\). The event where we have at least one \(x_{i1} \geq 0\) and one \(x_{i1} < 0\) also has probability \(1 - 2^{-n+1}\). As these events are independent, the probability that both occur is \((1 - 2^{-n+1})^2\). We will show that on this event we have \(\mathcal{L}(C_1) \leq \mathcal{L}(C_2)\) by producing another clustering \(C_3\) for which we can deduce \(\mathcal{L}(C_1) \leq \mathcal{L}(C_3)\) and \(\mathcal{L}(C_3) \leq \mathcal{L}(C_2)\). This is sufficient to ensure that \(i_1, i_2, \tilde{i}_1\) and \(\tilde{i}_2\) are well defined (the sets are non-empty). Let \(R_1\) be the range of \(\{x_{i1}\}\). Suppose that we define a new clustering \(C_3\) where all points in cluster 1 of \(C_2\) with \(x_{i2} = 0\) are reassigned to cluster 2 and all points in cluster 1 with \(x_{i2} = 1\) are reassigned to cluster 1; all other points retain the same cluster label. If \(x_i\) switched from cluster 1 to cluster 2, Proposition 1 tells us that

\[
d_G(x_i, x_{i1}) = d_G((x_{i1}, 0), (x_{i1}, 1)) \geq d_G((x_{i1}, 0), (x_{i1}, 0)) = d_G(x_i, x_{i2}).
\]

A similar results holds for the points that were switched from cluster 2 to cluster 1 in \(C_3\). This establishes that \(\mathcal{L}(C_2) \geq \mathcal{L}(C_3)\). The clustering \(C_3\) has the property that cluster 1 contains only
Hence, without loss of generality, assume \( \mathcal{C}_1 \) and \( \mathcal{C}_2 \) are independent conditional on \( \mathcal{C}_1 \) and \( \mathcal{C}_2 \) reasoning above, due to the independence of \( \mathcal{C}_1 \) and \( \mathcal{C}_2 \) we have

\[
\mathcal{L}(\mathcal{C}_3) = \frac{1}{2R_1} \left[ \sum_{i:x_{i2}=0} |x_{i1} - x_{i2}^*| + \sum_{i:x_{i2}=1} |x_{i1} - x_{i2}^*| \right]
\]

\[
\geq \frac{1}{2R_1} \left[ \sum_{i:x_{i2}=0} |x_{i1} - x_{i2}| + \sum_{i:x_{i2}=1} |x_{i1} - x_{i2}| \right] = \mathcal{L}(\mathcal{C}_1)
\]

because \( i_1 \) and \( i_2 \) are the closest data points to the relevant medians, which minimize the \( L_1 \) loss.

\[\square\]

.2 Proofs for properties of CLOG

.2.1 Proofs for local-global borrowing of information

In this section, we provide proofs to the theory on local-global borrowing of information in Section 1.3.1.

Proof for Lemma 1

Proof. Without loss of generality, assume \( C_i = C_i' = \ell \). As \( s_{ij} \) and \( s_{i'j} \) are independent conditional on \( C_i \) and \( C_i' \), we have

\[
P(s_{ij} = s_{i'j} = h \mid C_i = C_i' = \ell, \psi^{(j)}_\ell) = P(s_{ij} = h \mid C_i = \ell, \psi^{(j)}_\ell)P(s_{i'j} = h \mid C_i' = \ell, \psi^{(j)}_{i'\ell}) = \psi^{(j)}_{i'\ell}^2.
\]

Hence, \( P(s_{ij} = s_{i'j} = h \mid C_i = C_i' = \ell) = \mathbb{E}\left(\psi^{(j)}_{i'\ell}\right) = \mathbb{E}\left(\psi^{(j)}_{1h}\right) \). Summing over \( h \) gives

\[
P(s_{ij} = s_{i'j} \mid C_i = C_i' = \ell) = \sum_{h=1}^{\infty} \mathbb{E}\left(\psi^{(j)}_{1h}\right). \]

Since the result does not depend on \( \ell \),

\[
P(s_{ij} = s_{i'j} \mid C_i = C_i') = \sum_{h=1}^{\infty} \mathbb{E}\left(\psi^{(j)}_{1h}\right).
\]

To prove the second result, assume \( C_i = \ell \) and \( C_i' = \ell' \) with \( \ell \neq \ell' \). As \( s_{ij} \) and \( s_{i'j} \) are independent conditional on \( C_i \) and \( C_i' \), and \( \psi^{(j)}_\ell \) and \( \psi^{(j)}_{i'\ell} \), we have \( P(s_{ij} = s_{i'j} = h \mid C_i = \ell, C_i' = \ell', \psi^{(j)}_\ell, \psi^{(j)}_{i'\ell}) = P(s_{ij} = h \mid C_i = \ell, \psi^{(j)}_\ell)P(s_{i'j} = h \mid C_i' = \ell', \psi^{(j)}_{i'\ell}) = \psi^{(j)}_\ell \psi^{(j)}_{i'\ell}. \) By similar reasoning above, \( P(s_{ij} = s_{i'j} \mid C_i \neq C_i') = \sum_{h=1}^{\infty} \mathbb{E}(\psi^{(j)}_{1h}\psi^{(j)}_{2h}), \) which is \( \sum_{h=1}^{\infty} \mathbb{E}(\psi^{(j)}_{1h})\mathbb{E}(\psi^{(j)}_{2h}) \) due to the independence of \( \psi^{(j)}_{1h} \) and \( \psi^{(j)}_{2h} \).
Next we show the result when \( \psi^{(j)}_\ell \sim \text{Dir}(\omega_1^{(j)}, \ldots, \omega_{K_j}^{(j)}) \). With this assumption, the marginal is \( \psi^{(j)}_{\ell, h} \sim \text{Beta}(\omega_h^{(j)}, \sum_{m \neq h} \omega_m^{(j)}) \). Integrating over the distribution of \( \psi^{(j)}_{\ell, h} \) we have

\[
P(s_{ij} = s'_{i'j} = h \mid C_i = C'_{i'}) = \sum_h P(s_{ij} = s'_{i'j} = h \mid C_i = C'_{i'})
\]

\[
= \sum_h \frac{\omega_h^{(j)} + 1}{\omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}} \frac{\omega_h^{(j)} + 1}{\omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}}
\]

Thus,

\[
P(s_{ij} = s'_{i'j} \mid C_i = C'_{i'}) = \sum_h P(s_{ij} = s'_{i'j} = h \mid C_i = C'_{i'})
\]

\[
= \frac{\omega_h^{(j)} + 1}{\omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}} \frac{\omega_h^{(j)} + 1}{\omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}}
\]

For the second result. We know each of \( \psi^{(j)}_{\ell, h} \) and \( \psi^{(j)}_{\ell', h} \) follows a Beta distribution. Hence, integrating over their distributions yields:

\[
P(s_{ij} = s'_{i'j} = h \mid C_i = \ell, C'_{i'} = \ell')
\]

\[
= \int \int \psi^{(j)}_{\ell, h} \psi^{(j)}_{\ell', h} f(\psi^{(j)}_{\ell, h}) f(\psi^{(j)}_{\ell', h}) \frac{d\psi^{(j)}_{\ell, h}}{d\psi^{(j)}_{\ell', h}}
\]

\[
= \frac{B(\omega_h^{(j)} + 1, \sum_{m \neq h} \omega_m^{(j)})}{B(\omega_h^{(j)}, \sum_{m \neq h} \omega_m^{(j)})} \frac{B(\omega_{\ell'}^{(j)} + 1, \sum_{m \neq \ell'} \omega_m^{(j)})}{B(\omega_{\ell'}^{(j)}, \sum_{m \neq \ell'} \omega_m^{(j)})}
\]

\[
= \frac{\omega_h^{(j)} + 1}{\omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}} \frac{\omega_{\ell'}^{(j)} + 1}{\omega_{\ell'}^{(j)} + \sum_{i' \neq \ell'} \omega_{i'}^{(j)}}
\]

Summing over \( h \) gives

\[
P(s_{ij} = s'_{i'j} \mid C_i \neq C'_{i'}) = \frac{\sum_{h=1}^{K_j} \omega_h^{(j)} + 1}{\sum_{h=1}^{K_j} \omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}}
\]

\[
= \frac{\sum_{h=1}^{K_j} \omega_h^{(j)} + 1}{\sum_{h=1}^{K_j} \omega_h^{(j)} + \sum_{i' \neq h} \omega_{i'}^{(j)}}
\]
Proof for Proposition 3

Proof. Since \( \psi_{\ell}^{(j)} \) are independent and identically distributed across \( \ell \), \( \mathbb{E}(\psi_{\ell,h}^{(j)}) = \mathbb{E}(\psi_{\ell',h}^{(j)}) \). Hence, \( P(s_{ij} = s'_{ij} \mid C_i \neq C'_i) = \sum_h \mathbb{E}^2(\psi_{\ell,h}^{(j)}) \) according to Lemma 1. Since \( \text{Var}(\psi_{\ell,h}^{(j)}) = \mathbb{E}(\psi_{\ell,h}^{(j)2}) - \mathbb{E}^2(\psi_{\ell,h}^{(j)}) \) and \( \text{Var}(\psi_{\ell,h}^{(j)}) = \text{Var}(\psi_{1,h}^{(j)}) \) due to their identical distribution, \( P(s_{ij} = s'_{ij} \mid C_i = C'_i) = P(s_{ij} = s'_{ij} \mid C_i \neq C'_i) + \sum_h \text{Var}(\psi_{1,h}^{(j)}) \).

When \( \psi_{\ell}^{(j)} \sim \text{Dir}(\omega_{1}^{(j)}, \ldots, \omega_{K}^{(j)}) \), \( \psi_{1,h}^{(j)} \sim \text{Beta}(\omega_{h}^{(j)}, \sum_{m \neq h} \omega_{m}^{(j)}) \) and has variance
\[
\omega_{h}^{(j)} \sum_{m \neq h} \omega_{m}^{(j)}/((\sum_{h=1}^{K} \omega_{h}^{(j)})^2(\sum_{h=1}^{K} \omega_{h}^{(j)}) + 1).
\]
Hence, \( P(s_{ij} = s_{ij'} \mid C_i = C'_i) = P(s_{ij} = s_{ij'} \mid C_i \neq C'_i) + \sum_{h} \text{Var}(\psi_{1,h}^{(j)}) \).

Proof for Proposition 4

Proof. Write \( P(s_{ij} = s'_{ij} = \cdot \mid C_i = C'_i) \) and \( P(s_{ij} = s'_{ij} = \cdot \mid C_i \neq C'_i) \) as \( g_j(\cdot) \) and \( f_j(\cdot) \) respectively. Let \( P(C_i = C'_i) = q_1 \). By Bayes’ rule,
\[
P(C_i = C'_i \mid s_{ij} = s_{ij'}) = \frac{P(s_{ij} = s_{ij'} \mid C_i = C'_i)P(C_i = C'_i)}{P(s_{ij} = s_{ij'})} = \frac{\sum_h g_j(h)q_1}{\sum_h g_j(h)q_1 + \sum_h f_j(h)(1 - q_1)} = \frac{q_1}{q_1 + \sum_h f_j(h)(1 - q_1)}
\]
by Lemma 3. Next we calculate \( P(C_i = C'_i \mid s_{ij} = s_{ij'}, s_{ij'} = s_{ij''}) \). We first calculate \( P(s_{ij} = s_{ij} = h_1, s_{ij'} = s_{ij'} = h_2 \mid C_i = C'_i) \). Without loss of generality, assume \( C_i = C'_i = \ell \).

Due to conditional independence, we have \( P(s_{ij} = h_1, s_{ij'} = h_1, s_{ij'} = h_2 \mid C_i = C'_i = \ell, \psi_{\ell}^{(j)}, \psi_{\ell}^{(j')}) = P(s_{ij} = h_1 \mid C_i = \ell, \psi_{\ell}^{(j)})P(s_{ij'} = h_2 \mid C_i = \ell, \psi_{\ell}^{(j')}P(s_{ij'} = h_1 \mid C_i = \ell, \psi_{\ell}^{(j')} \psi_{\ell,h_1}^{(j')} \psi_{\ell,h_2}^{(j')2} = P(s_{ij} = s_{ij'} = h_1, s_{ij'} = s_{ij'} = h_2 \mid C_i = C'_i = g_j(h_1)g_{j'}(h_2). \) Hence, \( P(s_{ij} = s_{ij'}, s_{ij'} = s_{ij'} = C_i = C'_i) = \sum_h g_j(h_1)\sum_h g_{j'}(h_2). \) We can similarly derive that

"
\[ P(s_{ij} = s_{i'j'}, s_{ij'} = s_{i'j'} | C_i = C_{i'}) = \sum_{h_1} f_j(h_1) \sum_{h_2} f_{j'}(h_2). \] Hence,

\[
P(C_i = C_{i'} | s_{ij} = s_{i'j}, s_{ij'} = s_{i'j'}) = \frac{\sum_{h} g_j(h) \sum_{h_2} g_{j'}(h_2) q_1}{\sum_{h} g_j(h) q_1 + \sum_{h} f_j(h) \sum_{h_2} f_{j'}(h_2) (1 - q_1)}\]

\[
> \frac{\sum_{h} g_j(h) q_1 + \sum_{h} f_j(h) (1 - q_1)}{\sum_{h} g_j(h) q_1}
\]

\[= P(C_i = C_{i'} | s_{ij} = s_{i'j}). \]

where we have applied Proposition 3 for the inequality. \hfill \Box

**Proof for Corollary 1**

*Proof.* Use the notation defined in the proof of Proposition 4. First we see the result follows from Proposition 4 when \( S = \emptyset \) and \( S' = \{j\} \), where \( j \) is any fixed index in \([p]\), since \( P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in \emptyset) = P(C_i = C_{i'}) = q_1 \).

Next we show this result follows when \( S' = S \cup \{j^*\} \), where \( j^* \in [p] \setminus S \). Using similar steps for deriving \( P(C_i = C_{i'} | s_{ij} = s_{i'j}, s_{ij'} = s_{i'j'}) \), we can find that

\[
P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in A) = \frac{\prod_{j \in A} \sum_{h=1}^{K_j} g_j(h) q_1}{\prod_{j \in A} \sum_{h=1}^{K_j} g_j(h) q_1 + \prod_{j \in A} \sum_{h=1}^{K_j} f_j(h) (1 - q_1)},
\]

for \( A \in \{S, S'\} \). Dividing the numerator and denominator by \( \sum_{h=1}^{K_j} g_j(h) \) and by Proposition 3, the relationship \( P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in S) < P(C_i = C_{i'} | s_{ij} = s_{i'j}, \forall j \in S') \) follows.

Further, by mathematical induction, one can obtain the desired result for any \( S \subset S' \subset [p] \). \hfill \Box

### 2.2 Proofs for missing data handling property

Below is the proof for Proposition 5.
Proof. To see this, note that the term \( p(y_{i,\text{obs}}, y_{i,\text{mis}} | \xi) \) in (1.6) can be expressed as

\[
\int \sum_{\ell=1}^{\infty} \lambda_{\ell} \prod_{j=1}^{p} \sum_{h=1}^{\infty} \psi_{\ell,h}^{(j)} f_{\theta^{(j)}}^{(j)}(y_{ij}) d y_{i,\text{mis}}
\]

By the monotone convergence theorem, we obtain

\[
= \sum_{\ell=1}^{\infty} \lambda_{\ell} \left( \prod_{j \in \{m : I_{im} = 1\}} \sum_{h=1}^{\infty} \psi_{\ell,h}^{(j)} f_{\theta^{(j)}}^{(j)}(y_{ij}) \prod_{j \in \{m : I_{im} = 0\}} \int \sum_{h=1}^{\infty} \psi_{\ell,h}^{(j)} f_{\theta^{(j)}}^{(j)}(y_{ij}) d y_{ij} \right)
\]

\[
= \sum_{\ell=1}^{\infty} \lambda_{\ell} \left( \prod_{j \in \{m : I_{im} = 1\}} \sum_{h=1}^{\infty} \psi_{\ell,h}^{(j)} f_{\theta^{(j)}}^{(j)}(y_{ij}) \prod_{j \in \{m : I_{im} = 0\}} \int f_{\theta^{(j)}}^{(j)}(y_{ij}) d y_{ij} \right)
\]

\[
= \sum_{\ell=1}^{\infty} \lambda_{\ell} \prod_{j \in \{m : I_{im} = 1\}} \sum_{h=1}^{\infty} \psi_{\ell,h}^{(j)} f_{\theta^{(j)}}^{(j)}(y_{ij})
\]

where \( t_j \in \{c, o, q\} \) denotes the type of variable \( j \). \( \square \)

.3 Proofs for posterior asymptotics

This property is obtained under five assumptions. In this section, we show that OF-CLOG satisfies Assumptions 1-3 and 5, while extending the RM theory to an extended version of Assumption 4 that OF-CLOG satisfies.

(Assume for now that \( g_\xi(y) \) satisfies all five assumptions in RM.) We denote \( \Xi_k^0 = \{ \xi \in \Xi_k : f_\xi = f_0 \} \).

We now introduce some notation that is useful to characterize \( \Xi_k^0 \), following RM’s definition. Assume that \( \vartheta_k \)’s that are associated with \( \rho_{\ell_k} > 0 \) are unique in the sense that if at least one element between \( \vartheta_k \) and \( \vartheta_{k'} \) are different then they are different. We map \( \vartheta_k \)’s that are associated with \( \rho_{\ell_k} = 0 \) to a fixed shared parameter in the parameter space \( \Theta \); note that this does not affect the functional value of \( g_\xi(y) \) since the associated weights \( \rho_{\ell_k} = 0 \). Without loss of generality, we also assume \( \rho_{\ell 1}, \ldots, \rho_{\ell H} \) are ordered and are strictly monotonically decreasing except for the last few elements that are identically 0. We say \( \zeta_\ell = \zeta_{\ell'} \) if and only if

\[
H^0_\ell = H^0_{\ell'},
\]

\[
\forall k = 1, \ldots, H, \rho_{\ell_k} = \rho_{\ell'_{k}}, \vartheta_{\ell_k} = \vartheta_{\ell'_{k}}.
\]
We for now assume that $H^0_\ell = H$ for all $\ell$. We make this assumption because RM requires that $\zeta^0_\ell \in \text{int}(\Gamma)$, which means $\rho_{\ell k}$ (as a parameter in $\zeta^0_\ell$ cannot take the value of 0. Following RM, let $t = (t_i)_{i=0}^{K_0}$ with $0 = t_0 < t_1 < \ldots < t_{K_0} \leq K$ be a partition of $\{1, \ldots, K\}$. For all $\zeta \in \Xi$ such that $f_\zeta = f_0$ there exists $t$ as defined above such that, up to a permutation of the labels,

$$\forall i = 1, \ldots, K_0, \zeta_{t_i - 1 + 1} = \ldots = \zeta_{t_i} = \zeta^0_i,$$

$$p(i) = \sum_{j=t_i-1+1}^{t_i} \lambda_j = \lambda^0_i$$

$$\lambda_{t_{K_0} + 1} = \ldots = \lambda_K = 0.$$

In other words $I_i = \{t_{i-1} + 1, \ldots, t_i\}$ represents the cluster of components in $\{1, \ldots, K\}$ having the same parameter as $\zeta^0_i$. Then define the following parameterization of $\zeta \in \Xi$ (up to a permutation)

$$\phi_t = ((\zeta_{j})_{j=t_1}^{t_{K_0}-1}, (\lambda_j)_{j=t_{K_0}+1}) \in \mathbb{R}^{dt_{K_0} + K_0 + K - t_{K_0} - 1}, \quad s_i = p(i) - \lambda^0_i, \ i = 1, \ldots, K_0,$$

and

$$\psi_t = ((q_{j})_{j=1}^{t_{K_0}}, \zeta_{t_{K_0}+1}, \ldots, \zeta_K), \quad q_j = \lambda_j / p(i), \quad \text{when } j \in I_i = \{t_{i-1} + 1, \ldots, t_i\}.$$

Note that $f_0$ corresponds to

$$\phi^0_t = (\zeta^0_1, \ldots, \zeta^0_1, \zeta^0_2, \ldots, \zeta^0_2, \ldots, \zeta^0_{K_0}, \ldots, \zeta^0_{K_0}, 0, \ldots, 0)$$

where $\zeta^0_i$ is repeated $t_i - t_{i-1}$ times in the above vector, for any $\psi_t$. Then we parametrize $\zeta$ as $\phi_t, \psi_t$, so that $f_\zeta = f_{(\phi_t, \psi_t)}$, and we denote $f'_{(\phi^0_t, \psi_t)}$ and $f''_{(\phi^0_t, \psi_t)}$ the first and second derivatives of $f_{(\phi_t, \psi_t)}$ with respect to $\phi_t$ and computed at $\zeta_0 = (\phi^0_t, \psi_t)$.

We also define $F_0(g) = \int f_0(x)g(x)dx$, $\nabla g_\zeta$ to be the vector of first order derivatives of $g_\zeta$ with respect to $\zeta$, and $D^2 g_\zeta$ to be the matrix of second order derivatives with respect to $\zeta$. Define for $\delta \geq 0$

$$\bar{g}_\zeta = \sup_{|\zeta' - \zeta| \leq \delta} (g_\zeta'),$$

$$\underline{g}_\zeta = \inf_{|\zeta' - \zeta| \leq \delta} (g_\zeta').$$
where \(| \cdot |\) refers to the \(L_1\) norm. We also denote by \(P_\pi(\cdot | Y)\) the posterior distribution, where \(Y^n = (y_1, \ldots, y_n)\) is the data. The assumptions in RM written in the notation of our paper are as follows:

**Assumption 1** (Assumption 1 of Theorem 1 of RM). \(L_1\)-consistency: there exists \(\delta_n \leq \log(n)^q / \sqrt{n}\), for some \(q \geq 0\) such that

\[
\lim_{M \to \infty} \limsup_n \left[ E_n \left\{ P_\pi(\|f_0 - f_\zeta\| \geq M\delta_n | Y^n) \right\} \right] = 0.
\]

**Assumption 2** (Assumption 2 of Theorem 1 of RM). Regularity: the model \(\zeta \in \Gamma \leftarrow g_\zeta\) is three times differentiable and regular in the sense that for all \(\zeta \in \Gamma\) the Fisher information matrix that is associated with the model \(g_\zeta\) is positive definite at \(\zeta\). Denote by \(D^{(3)}g_\zeta\) the array whose components are

\[
\frac{\partial^3 g_\zeta}{\partial \zeta_1 \partial \zeta_2 \partial \zeta_3}.
\]

For all \(i \leq K_0\), there exists \(\delta > 0\) such that

\[
F_0 \left( \frac{\vartheta^3 g_{\zeta_0}^{(i)}}{\vartheta^{\alpha} g_{\zeta_0}^{(i)}} \right) < \infty, \quad F_0 \left\{ \sup_{|\zeta - \zeta_0| \leq \delta} (|\nabla g_\zeta|) \right\} < \infty, \quad F_0 \left( \frac{|\nabla g_{\zeta_0}^{(i)}|}{f_0} \right) < \infty.
\]

\[
F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0| \leq \delta} (|D^2 g_\zeta|)}{g_{\zeta_0}^{(i)}} \right\} < \infty, \quad F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0| \leq \delta} (|D^3 g_\zeta|)}{g_{\zeta_0}} \right\} < \infty.
\]

Assume also that for all \(\ell = 1,\ldots, K_0\), \(\xi_\ell^0 \in \text{int}(\Gamma)\) the interior of \(\Gamma\).

**Assumption 3** (Assumption 3 of Theorem 1 of RM). Integrability: there exists \(\Gamma_0 \subset \Gamma\) satisfying \(\text{Leb}(\Gamma_0) > 0\) and, for all \(i \leq K_0\),

\[
d(\xi_\ell^0, \Gamma_0) = \inf_{\zeta \in \Gamma_0} |\zeta - \xi_\ell^0| > 0
\]

and such that, for all \(\zeta \in \Gamma_0\),

\[
F_0 \left( \frac{\vartheta_{\zeta}^{(i)}}{f_0} \right) < \infty, \quad F_0 \left( \frac{\vartheta^3_{\zeta_0}^{(i)}}{\vartheta^{\alpha} g_{\zeta_0}^{(i)}} \right) < \infty, \quad \forall i \leq K_0.
\]
Assumption 4 (An extension of Assumption 4 of Theorem 1 of RM). Strong identifiability: for all $t$ partitions of $\{1, \ldots, k\}$ as defined above, let $\theta \in \Theta_k$ and write $\theta$ as $(\phi_t, \psi_t)$; then

$$(\phi_t - \phi_0^T f'_{\phi_t, \psi_t} + \frac{1}{2} (\phi_t - \phi_0^T f''_{\phi_t, \psi_t} (\phi_t - \phi_0) = 0 \iff$$

$$\forall i \leq k_0, s_i = 0 \quad \text{and} \quad \forall j \in I_i \quad q_j (\zeta - \zeta_t^0) = 0, \quad \forall i \geq t_k + 1 \lambda_i = 0.$$
where $\psi$ is the set of parameters shared across all mixture components. We assume that $\zeta^0 \in \text{int}(Z)$ and $\psi^0 \in \text{int}(\Psi)$ are in the interiors of their respective parameter spaces and $(\lambda_1, \ldots, \lambda_K)$ is in the $(K - 1)$-dimensional probability simplex $\mathcal{V}_K$. We begin with the condition on the prior:

**Assumption 6** (Assumption (a)(i) of Theorem 4.1 of Fruhwirth-Schnatter et al. (2019)). For all $c > 0$, there exists $H_1 > 0$ and $R_n \leq n^{H_1}$ such that $\mathbb{P}(Z_n^c) \leq e^{-c \log n}$ and $\mathbb{P}(\Psi_n^c) \leq e^{-c \log n}$ where $Z_n = \{ \zeta : \| \zeta \| \leq R_n \}$ and $\Psi_n = \{ \psi : \| \psi \| \leq R_n \}$.

**Lemma 3** (Sufficient condition for Assumption 6). Assume that we have a parameter $\theta \in \mathbb{R}^p$ such that $\theta_i - \mathbb{E}[\theta_i]$ is sub-exponential with parameters $\nu^2_i$ and $b_i$. For all $c > 0$ there exists $H_1 > 0$ and $R_n \leq n^{H_1}$ such that $\mathbb{P}(\| \theta \| > R_n) \leq e^{-c \log n}$ for all sufficiently large $n$.

**Proof.** Using the union bound we see

$$
\mathbb{P}(\| \theta \| > R_n) \leq \sum_{j=1}^p \mathbb{P}(|\theta_i| > |R_n|/\sqrt{p}).
$$

If $|R_n|/\sqrt{p} > 2|\mathbb{E}[\theta_i]|$ and $|R_n|/\sqrt{p} > \nu^2_i/b_i$, then from the reverse triangle inequality and Hoeffding’s inequality for sub-exponential random, we have

$$
\mathbb{P}(|\theta_i| > |R_n|/\sqrt{p}) \leq \mathbb{P}
\left|
\left|
\theta_i
\right|
\right|
\geq \frac{|R_n|}{2\sqrt{p}} + |\mathbb{E}[\theta_i]|
\right) \leq \mathbb{P}(|\theta_i - \mathbb{E}[\theta_i]| > 2^{-1}|R_n|/\sqrt{p}) \\
\leq \exp\left(-\frac{|R_n|}{4\sqrt{p}b_i}\right).
$$

Let $\bar{b} = \max_{j=1,\ldots,p} b_i$, so we have

$$
\mathbb{P}(\| \theta \| > R_n) \leq p \exp\left(-\frac{|R_n|}{4b\sqrt{p}}\right) = \exp\left(-\left[\frac{|R_n|}{4b\sqrt{p}} - \log p\right]\right). (10)
$$

We will now show have to pick $R_n = n^{H_1(c)}$ such that the above bound holds (i.e. $R_n$ satisfies the conditions on the mean and $\nu^2_i/b_i$) and

$$
\frac{|R_n|}{4b\sqrt{p}} - \log p > c \log n (11)
$$

for all $n \geq 2$, which will establish the desired result. Define

$$
H_1(c) = 1 + \max\left(1, \log_2 \left\{ 4\sqrt{p} \bar{b} \left[ c \log 2 + \log p \right] \right\}, \log_2 \left( 4c \bar{b} \sqrt{p} \right), \log_2 \sqrt{p} \max_{j=1,\ldots,p} \frac{\nu^2_i}{b_i}, \log_2 2 \sqrt{p} \max_{j=1,\ldots,p} |\mathbb{E}[\theta_i]| \right) .
$$
As $R_n$ is greater than the last two components, we have $R_2/\sqrt{p} > 2|\mathbb{E}[\theta_j]|$ and $R_2/\sqrt{p} > \nu_i^2/b_j$ for all $j = 1, \ldots, p$. As $R_n$ is increasing in $n$, this holds for all $n$, so equation (.10) is also valid for all $n \geq 2$.

We now turn to (.11). From the second argument in the maximum, we see by the construction of $H_1(c)$ that (.11) holds for $n = 2$. To show that it holds for all $n \geq 2$, we next establish that the derivative of the difference is non-negative (in fact, we will show that is is increasing). The derivative with respect to $n$ is

$$\frac{H_1(c)n^{H_1(c)-1}}{4b\sqrt{p}} - \frac{c}{n} > 0 \iff \frac{H_1(c)n^{H_1(c)}}{4b\sqrt{p}} > c.$$  

As $H_1(c) \geq 1$ and $n^{H_1(c)} > 2^{H_1(c)}$, we see that it is sufficient to have

$$\frac{2^{H_1(c)}}{4b\sqrt{p}} > c,$$

which holds from the third condition in the max that defines $H_1(c)$. Therefore the inequality (.11) holds and for our choice of $H_1(c)$ we have

$$\mathbb{P}(\|\theta\| > R_n) \leq \exp\left(-\frac{|R_n|}{4b\sqrt{p}} - \log p\right) \leq e^{-c\log n}$$

for all $n \geq 2$.

**Lemma 4** (Total variation is Lipschitz for the Categorial distribution). Let $f(x \mid p)$ and $f(x \mid q)$ be the PMFs for the categorical distributions $\text{Cat}(p)$ and $\text{Cat}(q)$ over some finite set $A$ for probability vectors $p$ and $q$. Then

$$\|f(\cdot \mid p) - f(\cdot \mid q)\|_1 = \|p - q\|.$$  

**Proof.** Suppose $X \sim \text{Cat}(p)$ and $Y \sim \text{Cat}(q)$. It is trivial that

$$\|f(\cdot \mid p) - f(\cdot \mid q)\|_1 = \sum_{i=1}^{\|A\|} |\mathbb{P}(X = a_i) - \mathbb{P}(Y = a_i)| = \|p - q\|.$$  

**Lemma 5** (Total variation is Lipschitz for univariate Gaussians). Let $f(x \mid \mu, \sigma^2)$ be the density of $N(\mu, \sigma^2)$. On the parameter space $(\mu, \sigma^2) \in \mathbb{R} \times (L, \infty)$ we have

$$\|f(\cdot \mid \mu_1, \sigma_1^2) - f(\cdot \mid \mu_2, \sigma_2^2)\|_1 \leq \frac{3}{\min(L, \sqrt{L})}(|\mu_1 - \mu_2| + |\sigma_1^2 - \sigma_2^2|).$$
Furthermore, there does not exist a constant $C$ such that

$$
\|f(\cdot \mid \mu_1, \sigma_1^2) - f(\cdot \mid \mu_2, \sigma_2^2)\|_1 \leq C(|\mu_1 - \mu_2| + |\sigma_1^2 - \sigma_2^2|) \tag{.12}
$$

for all $(\mu_1, \sigma_1^2), (\mu_2, \sigma_2^2) \in \mathbb{R} \times (0, \infty)$.

Proof. From Theorem 1.4 of Devroye et al. (2018), we can bound the total variation distance between two univariate Gaussians by

$$
\text{TV}(N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)) \leq \frac{3|\sigma_1^2 - \sigma_2^2|}{2 \max(\sigma_1, \sigma_2)^2} + \frac{|\mu_1 - \mu_2|}{2 \max(\sigma_1, \sigma_2)}.
$$

Recalling that the total variation distance is half of the $L_1$ distance between the densities, we have

$$
\|f(\cdot \mid \mu_1, \sigma_1^2) - f(\cdot \mid \mu_2, \sigma_2^2)\|_1 \leq \frac{3|\sigma_1^2 - \sigma_2^2|}{\max(\sigma_1, \sigma_2)^2} + \frac{|\mu_1 - \mu_2|}{\max(\sigma_1, \sigma_2)}
\leq \frac{3|\sigma_1^2 - \sigma_2^2|}{L} + \frac{3|\mu_1 - \mu_2|}{\sqrt{L}}
\leq \frac{3}{\min(L, \sqrt{L})}(|\mu_1 - \mu_2| + |\sigma_1^2 - \sigma_2^2|),
$$

which completes the proof of the first statement. For the second statement, we see intuitively that for two Gaussians with different means and variances converging to zero, the total variation distance will converge to one. Formally, we have the following easy counter example for any $C$. Let $\mu_1 = 0, \mu_2 = 5\sigma_1$ and $\sigma_1^2 = \sigma_2^2 = (10C)^{-2}$. For the set $A = (-2\sigma_1, 2\sigma_1)$, we have $P(X \in A) > 0.95$ when $X \sim N(0, \sigma_1^2)$ and

$$
P(Y \in A) < P(Y < 2\sigma_1) = P(Z < -3) < 0.01
$$

for $Y \sim N(5\sigma_1, \sigma_1^2)$ and $Z \sim N(0, 1)$. This means that the total variation distance is at least 0.94 and the $L_1$ distance is at least 1.88. However, we have

$$
C(|\mu_1 - \mu_2| + |\sigma_1^2 - \sigma_2^2|) = C \frac{5}{10C} = 1/2,
$$

which shows that (.12) does not hold for this $C$. As $C$ was arbitrary, we can construct such an example for any choice of $C$, which proves the second claim.

If we pick $L$ smaller than one, then we can simplify by $\min(L, \sqrt{L}) = L$. This bound does not require us to bound $\sigma^2$ away from zero, and the second point shows that this is necessary.
Lemma 6 (Joint-marginal triangle inequality for TV). Let \((X_1, \ldots, X_d)\) and \((Y_1, \ldots, Y_d)\) be \(d\)-dimensional random variables each of which has independent marginals. We have the following inequality:

\[
\|\mathcal{L}(X_1, \ldots, X_d) - \mathcal{L}(Y_1, \ldots, Y_d)\|_{TV} \leq \sum_{i=1}^{d} \|\mathcal{L}(X_i) - \mathcal{L}(Y_i)\|_{TV}.
\]

Proof. Let \(\Gamma\) be the set of all couplings between the vectors and let \(\gamma\) be the coupling of \((X_1, \ldots, X_d)\) and \((Y_1, \ldots, Y_d)\) which is the product of independent maximal couplings between \(X_i\) and \(Y_i\) for all \(i = 1, \ldots, d\). Using the union bound and noting that \(\gamma\) is a coupling that will achieve

\[
P_\gamma(X_i \neq Y_i) = \|\mathcal{L}(X_i) - \mathcal{L}(Y_i)\|_{TV},
\]

we have

\[
\|\mathcal{L}(X_1, \ldots, X_d) - \mathcal{L}(Y_1, \ldots, Y_d)\|_{TV} = \inf_{\gamma} \left( \sum_{i=1}^{d} P_\gamma(X_i \neq Y_i) \right)
\]

\[
\leq \sum_{i=1}^{d} \|\mathcal{L}(X_i) - \mathcal{L}(Y_i)\|_{TV},
\]

which proves the result. \(\square\)

Lemma 7 (Products of Lipschitz densities is Lipschitz). Consider a density \(f(x \mid \theta) = \prod_{j=1}^{p} f_j(x_j \mid \theta)\) where

\[
\|f_j(x_j \mid \theta_1) - f_j(x_j \mid \theta_2)\|_1 \leq C_j|\theta_1 - \theta_2|.
\]

Then the joint density is Lipschitz with constant \(\sum_{j=1}^{p} C_j\):

\[
\|f(x \mid \theta_1) - f(x \mid \theta_2)\|_1 \leq |\theta_1 - \theta_2| \sum_{j=1}^{p} C_j.
\]

Proof. As the components are independent, we can apply Lemma 6 to get

\[
\|f(x \mid \theta_1) - f(x \mid \theta_2)\|_1 = 2TV(f(x \mid \theta_1), f(x \mid \theta_2))
\]

\[
\leq 2 \sum_{j=1}^{p} TV(f_j(x_j \mid \theta_1), f_j(x_j \mid \theta_2))
\]

\[
\leq \sum_{j=1}^{p} C_j|\theta_1 - \theta_2| = |\theta_1 - \theta_2| \sum_{j=1}^{p} C_j,
\]

which establishes the result. \(\square\)
Lemma 8 (Mixture of Lipschitz densities is Lipschitz). Let $f(x \mid \tilde{\theta}) = \sum_{k=1}^{K} w_k g_k(x \mid \theta_k)$ where $	ilde{\theta} = (w, \theta_1, \ldots, \theta_K)$ and assume that there exist constants $C_k$ such that

$$\|g_k(x \mid \theta_1) - g(x \mid \theta_2)\|_1 \leq C_k|\theta_1 - \theta_2|$$

for $k = 1, \ldots, K$. Then

$$\|f(x \mid \tilde{\theta}_1) - f(x \mid \tilde{\theta}_2)\|_1 \leq (\max_k C_k \lor 1)|\tilde{\theta}_1 - \tilde{\theta}_2|.$$

Proof. We have

$$\|f(x \mid \tilde{\theta}_1) - f(x \mid \tilde{\theta}_2)\|_1 = \int \left| \sum_{k=1}^{K} w_k g_k(x \mid \theta_1) - w_k g_k(x \mid \theta_2) \right| \, dx \leq \sum_{k=1}^{K} \int |w_k g_k(x \mid \theta_1) - w_k g_k(x \mid \theta_2)| \, dx.$$ 

The inner integral can be upper bounded using the triangle inequality:

$$\int |w_k g_k(x \mid \theta_1) - w_k g_k(x \mid \theta_2)| \, dx \leq \int |w_k g_k(x \mid \theta_1) - w_k g_k(x \mid \theta_1)| \, dx + \int |w_k g_k(x \mid \theta_1) - w_k g_k(x \mid \theta_2)| \, dx = |w_k - w_k| + w_k C_k |\theta_1 - \theta_2|$$

for all $k = 1, \ldots, K$. Combining these bounds give

$$\|f(x \mid \tilde{\theta}_1) - f(x \mid \tilde{\theta}_2)\|_1 \leq \sum_{k=1}^{K} |w_k - w_k| + w_k \max_k C_k \sum_{k=1}^{K} |\theta_1 - \theta_2| \leq \left( \max_k C_k \lor 1 \right) \left[ |w_1 - w_2| + \sum_{k=1}^{K} |\theta_1 - \theta_2| \right] = \left( \max_k C_k \lor 1 \right) |\tilde{\theta}_1 - \tilde{\theta}_2|,$$

which establishes the desired condition.

It is now simple to establish that the mixture model used in CLOG is Lipschitz.

Lemma 9 (CLOG density is Lipschitz). The CLOG density is Lipschitz in parameters.
Proof. As we assume that the kernels \( g \) are mixture distributions where each of the kernels of the mixture can be expressed as products of univariate Gaussians or Categorical distributions. As both of these are Lipschitz in parameters (by Lemmas 4 and 5), we know that each of the kernels that are mixed to form \( g \) are Lipschitz in parameters by Lemma 7. Applying Lemma 8 tells us that \( g \) is also Lipschitz in parameters, as desired. \( \Box \)

We now turn to assumption (b) of Theorem 4.1 of Fruhwirth-Schnatter et al. (2019), which states

**Assumption 7** (Assumption (b) of Theorem 4.1 of Fruhwirth-Schnatter et al. (2019)). There exists a constant \( H_3 \geq 0 \) such that

\[
\left\{ \| \psi - \psi^0 \| + \sum_{\ell=1}^{K} |\lambda_\ell - \lambda^0_\ell + \|\zeta_\ell - \zeta^0_\ell\| \right\} \subset S_n
\]

where

\[
S_n = \{ (\lambda, \eta, \psi) : KL(f_0, f(\cdot | \lambda, \eta, \psi)) + V(f_0, f(\cdot | \lambda, \eta, \psi)) \leq 1/n \}. 
\]

We thus wish to establish that if the distance between the true parameters and the parameters of some other mixture density \( f \) are small, then \( KL(f_0, f) + V(f_0, f) \) is small. In what follows, we will use \( w \) to denote a fixed probability vector and \( v \) to be another probability vector (assumed to be close to \( w \) in \( L_1 \)). For other parameters, parameters with subscript 1 will be denote fixed parameters (on which constants will be allowed to depend) and those with a subscript 2 will be allowed to vary.

Thus \( \mu_1 \) is fixed but \( \mu_2 \) can vary in an interval \( [\mu_1 - n^{-\alpha}, \mu_2 + n^{-\alpha}] \). This mean, as an example, that \( \| w - v \| \leq n^{-\alpha} \) is a condition about \( v \) being in a certain interval centered around a fixed \( w \), not a statement about any two vectors that are close. If \( w \) and \( v \) are probability vectors, then we write

\[
KL(w, v) = \sum_{j=1}^{p} w_j \log \frac{w_j}{v_j}
\]

\[
V(w, v) = \sum_{j=1}^{p} w_j \left( \log \frac{w_j}{v_j} \right)^2 - (KL(w, v))^2.
\]

We now generalize a result about the convexity of the KL divergence. Let \( a_1, \ldots, a_n \) and \( b_1, \ldots, b_n \) be non-negative real numbers and define \( a = \sum_{i=1}^{n} a_i \) and \( b = \sum_{i=1}^{n} b_i \). The log-sum
inequality states that
\[ a \log \frac{a}{b} \leq \sum_{i=1}^{n} a_i \log \frac{a_i}{b_i}. \] (13)

Using (13) it is simple to prove that for two mixtures \( f = \sum_{h=1}^{H} w_h f_h \) and \( g = \sum_{h=1}^{H} w_h g_h \) we have
\[ KL(f, g) \leq \sum_{h=1}^{H} w_h KL(f_h, g_h). \]

This result is for mixtures with equal weights, but we can easily generalize to mixtures with unequal weights using (13).

We begin with an elementary result to establish an upper bound of the second moment of a log-likelihood ratio.

**Lemma 10 (Log-squared bound).** For all \( x > 0 \) we have \( x \log x)^2 \leq (x - 1)^2 \).

**Proof.** Let \( f(x) = (x - 1)^2 - x \log x)^2 \). The function is continuous and from L'Hôpital’s rule we get \( \lim_{x \to 0} f(x) = 1 \), so we can define \( f(0) = 1 \) and have a continuous function on \([0, \infty)\).
The derivatives are \( f'(x) = 2(x - 1) - 2 \log x - (\log x)^2 \) and \( f''(x) = 2(1 - x^{-1} - x^{-1} \log x) \). We see that the second derivative is minimized at \( x = 1 \) with \( f''(1) = 0 \). Therefore \( f(x) \geq 0 \), which implies that \( f \) is convex. As \( f'(1) = 0 \), we see that \( f(0) = 0 \) is the global minimizer. Thus \( f(x) \geq 0 \), which implies the desired result.

Recall that the \( \chi^2 \) divergence is given by
\[ \chi^2(p, q) = \int q \left( \frac{p}{q} - 1 \right)^2 = \int \frac{p^2}{q} - 1. \] (14)

We can now show that the first and second moments of the log-likelihood ratio are both bounded by the \( \chi^2 \) divergence between the densities.

**Lemma 11 (Bound on second moment of LLR).** Let \( p \) and \( q \) be densities. We have \( KL(p, q) \leq \chi^2(p, q) \) and \( V(p, q) \leq \chi^2(p, q) \). Consequently \( KL(p, q) + V(p, q) \leq 2 \chi^2(p, q) \).
Proof. From the inequality \( \log(x) \leq x - 1 \) we get

\[
KL(p, q) = \int p \log \frac{p}{q} \leq p \left( \frac{p}{q} - 1 \right) = \int \frac{p^2}{q} - \int p = \chi^2(p, q).
\]

From Lemma 10 we get

\[
V(p, q) \leq \int p \left( \log \frac{p}{q} \right)^2 = \int q \frac{p}{q} \left( \log \frac{p}{q} \right)^2 \leq \int q \left( \frac{p}{q} - 1 \right)^2 = \chi^2(p, q).
\]

Thus \( KL(p, q) + V(p, q) \leq 2\chi^2(p, q) \). □

This means that it is sufficient to establish that the \( \chi^2 \) divergence is small when the parameters are close in order to establish that Assumption 7.

**Lemma 12** (LLR variance bound for mixtures). Suppose that \( p(x) = \sum_{j=1}^{p} w_j f_j(x) \) and \( q(x) = \sum_{j=1}^{p} v_j g_j(x) \). Then

\[
\chi^2(p, q) \leq \sum_{j=1}^{p} \left( w_j \frac{w_j}{v_j} \int \frac{f_j^2}{g_j} - 1 \right).
\]

(.15)

Proof. Recall the generalized log-sum inequality: if \( f \) is strictly convex at \( c = a/b \) where \( a = \sum_i a_i \) and \( b = \sum_i b_i \), then

\[
bf \left( \frac{a}{b} \right) \leq \sum_i b_i f \left( \frac{a_i}{b_i} \right).
\]

(.16)

This result follows because \( xf(x) \) is convex when \( f \) is strictly convex. As the function \( f(x) = (x - 1)^2 \) is strictly convex we have

\[
\chi^2(p, q) = \int q \left( \frac{p}{q} - 1 \right)^2 \leq \int \sum_{j=1}^{p} v_j g_j \left( \frac{w_j f_j}{v_j g_j} - 1 \right)^2
\]

\[
= \sum_{j=1}^{p} \frac{w_j^2}{v_j} \int \frac{f_j^2}{g_j} - 2 \sum_{j=1}^{p} w_j \int f_j + \sum_{j=1}^{p} v_j \int g_j = \sum_{j=1}^{p} w_j \frac{w_j}{v_j} \int \frac{f_j^2}{g_j} - 1.
\]

□

Equation (.15) allows us to reduce the calculation over the mixture distribution to a calculation over the kernels \( f_j \) and \( g_j \).

The following result will be used in establishing the necessary condition on the \( \chi^2 \) divergence.
Lemma 13 (Sequence control). For any constants $C > 0$ and $\beta > 0$, there exists an $\alpha$ such that for all $\alpha \geq \alpha$ we have $Cn^{-\alpha} < n^{-\beta}$ for all $n$.

Proof. This condition is equivalent to $n^{\beta - \alpha} < C^{-1}$. If $\alpha > \beta$, then this is a decreasing function in $n$, so it suffices to have $2^{\beta - \alpha} < C^{-1}$. This will occur if $\alpha > \log_2(C) + \beta$. Thus the result holds if we take $\alpha > \max(\beta, \beta + \log_2(C))$. $\Box$

Lemma 14 ($\chi^2$ divergence for discrete distributions). Suppose $w$ is a probability vector on the probability simplex. Then for any $\beta > 0$ there exists an $\alpha$ such that for all $\alpha \geq \alpha$ there exists a constant $C$ such that for all $v \in V_p$

$$\|w - v\| \leq n^{-\alpha} \implies \frac{w_j}{v_j} \leq 1 + Cn^{-\alpha} \quad (1.17)$$

and

$$\|w - v\| \leq n^{-\alpha} \implies \chi^2(w, v) = \sum_{j=1}^p \frac{w_j^2}{v_j^2} - 1 \leq n^{-\beta}. \quad (1.18)$$

Proof. For any $j$ such that $w_j = 0$, equation (1.17) trivially holds for any $C > 0$. We will assume that $\alpha$ is large enough that $w_j - n^{-\alpha} > 0$ for all other $j$. As $(1 - x)^{-1} \leq 1 + 2x$ for $x \in [0, 1/2]$, we have that

$$\frac{w_j}{v_j} \leq \frac{w_j}{w_j - n^{-\alpha}} = \frac{1}{1 - n^{-\alpha}w_j^{-1}} \leq 1 + 2n^{-\alpha}w_j^{-1}$$

when $\alpha$ is large enough that $n^{-\alpha}w_j^{-1} \leq 1/2$. If we take $C^{-1} = 2 \min_{j: w_j \neq 0} w_j$, then we have proved the first statement. The second statement follows easily; we have

$$\sum_{j=1}^p \frac{w_j}{v_j} \leq 1 + Cn^{-\alpha},$$

so if we take $\alpha$ such that $Cn^{-\alpha} \leq n^{-\beta}$ for all $n$ (by Lemma 13), then

$$\chi^2(w, v) \leq Cn^{-\alpha} \leq n^{-\beta},$$

which establishes the second claim. $\Box$
Lemma 15 (Combining marginal divergences). Fix any $\beta > 0$. Assume that for all $j = 1, \ldots, p$ we know that for all $\alpha_j \geq \alpha_j$ we have

$$\|\theta_j - \theta_j\| \leq n^{-\alpha_j} \quad \implies \quad \int_0^1 \frac{f(x \mid \theta_j)}{f(x \mid \theta_j)} \, dx \leq 1 + C_j n^{-\beta}. \tag{.19}$$

Then there exists an $\alpha$ and a constant $C$ such that for all weight vectors $w$ on the probability simplex we have

$$\|w - v\| + \|\theta_1 - \theta_2\| \leq n^{-\alpha} \implies \chi^2 \left( \sum_{j=1}^p w_j f(x \mid \theta_1), \sum_{j=1}^p v_j f(x \mid \theta_2) \right) \leq n^{-\beta}.$$

**Proof.** First, we take $\alpha$ large enough that (.19) holds for all $j$. We have that $\|w - v\| \leq n^{-\alpha}$ and $\|\theta_1 - \theta_2\| \leq n^{-\alpha}$ for all $j$. By Lemma 12 we have that

$$\chi^2(p, q) \leq \sum_{j=1}^p w_j (1 + C' n^{-\alpha}) (1 + C_j n^{-\alpha}) - 1 \leq \sum_{j=1}^p w_j (1 + C n^{-\alpha}) - 1 = C n^{-\alpha}$$

when the constant $C$ is picked large enough. If we pick $\alpha$ large enough that $C n^{-\alpha} < n^{-\beta}$ for all $n$ (which is possible by Lemma 13), then we have

$$\chi^2(p, q) \leq C n^{-\alpha} < n^{-\beta}$$

for all $n$. \qed

Lemma 16 ($\chi^2$ distribution for products). Suppose that we have densities $f(x) = \prod_{j=1}^p f_j(x_j)$ and $g(x) = \prod_{j=1}^p g_j(x_j)$ where $x = (x_1, \ldots, x_p)$. If there are constants $C_j$ such that

$$\int \frac{f_j^2}{g_j} \leq 1 + C_j n^{-\alpha},$$

for all $j = 1, \ldots, p$, then

$$\chi^2(f, g) \leq C n^{-\alpha}$$

for some constant $C$.

**Proof.** We can write

$$\int \frac{f_j^2}{g_j} = \prod_{j=1}^p \int \frac{f_j^2}{g_j} \leq \prod_{j=1}^p (1 + C_j n^{-\alpha}) \leq 1 + C n^{-\alpha}$$
for some constant $C$. From this it follows that

$$\chi^2(f, g) = \int \frac{f^2}{g} - 1 \leq C n^{-\alpha},$$

as desired. \hfill \Box

We now need to verify that the kernels we are interested in satisfy (.19). To do this, we first derive the $\chi^2$ divergence between Gaussians.

**Lemma 17** ($\chi^2$ divergence between Gaussians). *Let $p$ be the density of $N(\mu_1, \sigma_1^2)$ and $q$ be the density of $N(\mu_2, \sigma_2^2)$. Then

$$\chi^2(p, q) = \frac{\sigma_2^2}{\sigma_1^2} \sqrt{2\sigma_2^2 - \sigma_1^2} \exp \left( \frac{(\mu_1 - \mu_2)^2}{2\sigma_2^2 - \sigma_1^2} \right) - 1 \tag{.20}$$

when $\sigma_2^2 > \sigma_1^2/2$.***

**Proof.** The ratio can be written as

$$\frac{p^2}{q} = \frac{\sigma_2^2}{\sqrt{2\pi \sigma_1^2}} \exp \left( \frac{(x - \mu_2)^2}{2\sigma_2^2} - \frac{(x - \mu_1)^2}{\sigma_1^2} \right).$$

Expanding the polynomial in the exponential gives

$$\frac{(x - \mu_2)^2}{2\sigma_2^2} - \frac{(x - \mu_1)^2}{\sigma_1^2} = \frac{\mu_2^2 - 2\mu_2 \mu_1}{2\sigma_2^2} + \frac{2\mu_1 \mu_2 - 2\mu_2 \sigma_1^2}{2\sigma_2^2 \sigma_1^2} + \frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2 \sigma_2^2}.$$ 

Recalling the integral identity

$$\int_{-\infty}^{\infty} e^{-(ax^2 + bx + c)} \, dx = \sqrt{\frac{\pi}{a}} \exp \left( \frac{b^2}{4a} - c \right)$$

for $a > 0$, we find that

$$\int \frac{p^2}{q} = \left( \frac{\sqrt{\sigma_2^2}}{\sqrt{2\pi \sigma_1^2}} \right) \sqrt{\frac{2\sigma_1^2 \sigma_2^2 \pi}{2\sigma_2^2 - \sigma_1^2}} \exp \left( \frac{(\mu_1 - \mu_2)^2}{2\sigma_2^2 - \sigma_1^2} \right) \tag{.20}$$

after some simplification when $\sigma_2^2 > \sigma_1^2/2$. By subtracting one and canceling $\sqrt{2\pi}$, we recover (.20). \hfill \Box

We now show that the $\chi^2$ divergence between Gaussians is small when the parameters are close.
Lemma 18 ($\chi^2$ divergence bound for Gaussians). Let $p$ be the density of $N(\mu_1, \sigma_1^2)$ and $q$ be the density of $N(\mu_2, \sigma_2^2)$. Given $\beta > 0$ for sufficiently large $\alpha$ we have

$$|\mu_1 - \mu_2| + |\sigma_1^2 - \sigma_2^2| \leq n^{-\alpha} \implies \int \frac{p^2}{q} \leq 1 + n^{-\beta}.$$  

Proof. We will take $\alpha$ large enough that $\sigma_2^2 > \sigma_1^2/2$ so that (20) holds. We first bound

$$\frac{\sigma_2^2}{\sigma_1^2} \leq \frac{\sigma_1^2 + n^{-\alpha}}{\sigma_1^2} = 1 + n^{-\alpha} \sigma_1^{-2}.$$  

Recall that $(1 - x)^{-1} \leq 1 + 2x$ for $0 \leq x \leq 1/2$. As the square-root function is monotonic and convex for positive reals, we have

$$\sqrt{\frac{\sigma_1^2}{2\sigma_2^2 - \sigma_1^2}} \leq \sqrt{\frac{\sigma_1^2}{\sigma_1^2 - 2n^{-\alpha}}} \leq \sqrt{1 + 4n^{-\alpha} \sigma_1^{-2}} \leq 1 + 2n^{-\alpha} \sigma_1^{-2}$$  

when $\alpha$ large enough that $2n^{-\alpha} \sigma_1^{-2} \leq 1/2$. Assuming that $\alpha$ is large enough that $\sigma_1^2 - 2n^{-\alpha} > \sigma_1^2/2$, and recalling that $e^x \leq 1 + 2x$ for $0 \leq x \leq 5/4$ we have

$$\exp\left(\frac{(\mu_1 - \mu_2)^2}{2\sigma_2^2 - \sigma_1^2}\right) \leq \exp\left(2\sigma_1^{-2} n^{-2\alpha}\right) \leq 1 + 4\sigma_1^{-2} n^{-2\alpha}$$  

when $\alpha$ is large enough that $2\sigma_1^{-2} n^{-2\alpha} \leq 5/4$.

Putting these bounds together gives

$$\int \frac{p^2}{q} = \frac{\sigma_2^2}{\sigma_1^2} \sqrt{\frac{\sigma_1^2}{2\sigma_2^2 - \sigma_1^2}} \exp\left(\frac{(\mu_1 - \mu_2)^2}{2\sigma_2^2 - \sigma_1^2}\right)$$

$$\leq (1 + n^{-\alpha} \sigma_1^{-2})(1 + 2n^{-\alpha} \sigma_1^{-2})(1 + 4\sigma_1^{-2} n^{-2\alpha}) \leq 1 + Cn^{-\alpha}$$

for some constant $C$. By Lemma 13, we may pick $\alpha$ large enough to ensure that $1 + Cn^{-\alpha} \leq 1 + n^{-\beta}$ for all $n$. Thus for all sufficiently large $\alpha$ we have the desired result. \hfill \square

We are now ready to prove that CLOG satisfies Assumption 7.

Lemma 19. Assumption 7 is satisfied when

1. The true density is a mixture of CLOG kernels,

2. The true density is a mixture of mixture of CLOG kernels.
Proof. A CLOG kernel is of the form \( f(x) = \prod_{i=1}^{p} f_i(x_i) \) where \( f_i \) is either a univariate Gaussian or a categorical distribution. From Lemmas 14 and 18, we know that all of these kernels satisfy the condition of Lemma 15, so we have that the kernels satisfy condition (19) of Lemma 15. If we pick \( \beta \) small enough that \( 2n^{-\beta} \leq 1/n \) for all \( n \), then by Lemma 11 we have established Assumption 7 with \( H_3 = \alpha \).

As a mixture of mixtures can be written as a one-layer mixture, we can trivially extend the result to mixtures of mixtures. \( \square \)

Assumption 2

Lemma 20. For real number \( p \geq 1 \) and real numbers \( a_1, \ldots, a_n \), we have

\[
| \sum_{i=1}^{n} a_i|^p \leq n^{p-1} \sum_{i=1}^{n} |a_i|^p.
\]

Proof. The map \( x \mapsto x^p, x \geq 0 \) is convex, since its second order derivative is \( p(p-1)x^{p-2} \geq 0 \). By Jensen’s inequality, we have

\[
| \sum_{i=1}^{n} a_i/n|^p \leq \sum_{i=1}^{n} |a_i|^p/n.
\]

Thus

\[
| \sum_{i=1}^{n} a_i|^p \leq n^{p-1} \sum_{i=1}^{n} |a_i|^p.
\]


Proof. Let \( \delta_\ell = \min (\min_k \rho_{i_k}^0, \delta) \).

A. First we show for all \( \ell \leq K_0 \), there exists \( \delta' > 0 \) such that \( F_0 \left( \frac{g_3}{p_{i_k}^0} \right) < \infty \). We have that
for all \( k \leq H_\ell^0 = H \) there exists \( \delta > 0 \) such that \( F_0 \left( \frac{h_\zeta^3}{g_\zeta^3} \right) < \infty \). By definition, we have

\[
g_\zeta^0 = \inf_{\zeta' - \zeta^0 \leq \delta} (g_{\zeta'})
\]

\[
= \inf_{|\zeta' - \zeta^0| \leq \delta} \sum_{k=1}^H \rho_k' h_{\varphi_k'}(y)
\]

\[
\geq \sum_{k=1}^H \rho_k' \inf_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}(y)
\]

\[
\geq H \sum_{k=1}^H (\rho_k^0 - \delta) \inf_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}(y). \tag{.21}
\]

We can similarly show

\[
\bar{g}_\zeta^0 = \sum_{k=1}^H (\rho_k^0 + \delta) \sup_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}(y). \tag{.22}
\]

Using (.21) and (.22), we can upper bound \( F_0 \left( \frac{\bar{g}_\zeta^0}{g_\zeta^0} \right) \) as follows:

\[
F_0 \left( \frac{\bar{g}_\zeta^0}{g_\zeta^0} \right) \leq \frac{\sum_{k=1}^H (\rho_k^0 + \delta) \sup_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}^3(y)}{\sum_{k=1}^H (\rho_k^0 - \delta) \inf_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}^3(y)}
\]

\[
\leq \frac{H \sum_{k=1}^H (\rho_k^0 + \delta) \sup_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}^3(y)}{H \sum_{k=1}^H (\rho_k^0 - \delta) \inf_{|\varphi_k' - \varphi_k^0| \leq \delta} h_{\varphi_k'}^3(y)}
\]

\[
\leq \frac{H (\rho_k^0 + \delta)^3}{(\rho_k^0 - \delta)^3} F_0 \left( \frac{h_{\varphi_k'}^3}{h_{\varphi_k'}^3} \right) < \infty.
\]

Setting \( \delta' = \min \delta \), we have the desired result.

B. Next we show for all \( \ell \leq K_0 \), there exists \( \delta' \) such that

\[
F_0 \left( \frac{\sup_{|\zeta' - \zeta^0| \leq \delta'} (|\nabla g_\zeta|^3)}{g_\zeta^3} \right) < \infty.
\]

We show the result by taking the same \( \delta' \) defined above. Since \( \sum_{k=1}^H \rho_\ell = 1 \), without loss of generality we can set \( \rho_{\ell H} = 1 - \sum_{k=1}^{H-1} \rho_\ell k \). Taking the norm of the first order derivatives of \( g_\zeta \)
gives
\[ |\nabla g_\zeta(y)| = \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)| + \sum_{k=1}^{H-1} |h_{\phi_k}(y) - h_{\phi_H}(y)|. \]  

(23)

By Lemma 20, we can upper bound \( |\nabla g_\zeta(y)| \) as follows:
\[
|\nabla g_\zeta(y)|^3 \leq (2H - 1)^2 \left( \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)|^3 + \sum_{k=1}^{H-1} |h_{\phi_k}(y) - h_{\phi_H}(y)|^3 \right)
\]

\[
\leq (2H - 1)^2 \left( \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)|^3 + \sum_{k=1}^{H-1} |h_{\phi_k}(y) - h_{\phi_H}(y)|^3 \right) \text{ by Lemma 20}
\]

\[
\leq (2H - 1)^2 \left( \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)|^3 + 4 \sum_{k=1}^{H-1} (|h_{\phi_k}(y)|^3 + |h_{\phi_H}(y)|^3) \right)
\]

Therefore
\[
\sup_{|\zeta - \zeta_0| \leq \delta'} |\nabla g_\zeta(y)|^3
\]

\[
\leq (2H - 1)^2 \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)|^3 + 4(2H - 1)^2 \sum_{k=1}^{H-1} |h_{\phi_k}(y)|^3
\]

\[
+ 4(H - 1)(2H - 1)^2 \sup_{|\phi_k - \phi_H| \leq \delta'} |h_{\phi_H}(y)|^3 \quad (24)
\]

\[
\leq (2H - 1)^2 \sum_{k=1}^{H} |\nabla h_{\phi_k}(y)|^3 + 4(2H - 1)^2 \sum_{k=1}^{H-1} |h_{\phi_k}(y)|^3
\]

\[
+ 4(H - 1)(2H - 1)^2 \sup_{|\phi_k - \phi_H| \leq \delta} |h_{\phi_H}(y)|^3 \quad (25)
\]

By (21) and (25), we have
\[
F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0| \leq \delta'} (|\nabla g_\zeta|)}{g_{\zeta_0}} \right\}
\]

\[
\leq \sum_{k=1}^{H} \frac{(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left\{ \frac{\sup_{|\phi_k - \phi_H| \leq \delta}}{h_{\theta_k}^3} \right\} + \sum_{k=1}^{H-1} \frac{4(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left\{ \frac{\sup_{|\phi_k - \phi_H| \leq \delta}}{h_{\theta_k}^3} \right\}
\]

\[
+ \frac{4(H - 1)(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left\{ \frac{\sup_{|\phi_k - \phi_H| \leq \delta}}{b_{\theta_k}^3} \right\}
\]

\[
\leq \sum_{k=1}^{H} \frac{(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left\{ \frac{\sup_{|\phi_k - \phi_H| \leq \delta}}{h_{\theta_k}^3} \right\} + \sum_{k=1}^{H-1} \frac{4(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left\{ \frac{\sup_{|\phi_k - \phi_H| \leq \delta}}{h_{\theta_k}^3} \right\}
\]

\[
+ \frac{4(H - 1)(2H - 1)^2}{\rho_{\ell_k}^0 - \delta_{\ell}} F_0 \left( \frac{\bar{h}_{\theta_k}^3}{b_{\theta_k}^3} \right) < \infty \quad \text{by assumption 2.}
\]
C. Next we show $F_0 \left( \frac{\| \nabla g_0 \|_4}{f_0} \right) < \infty$. We refer to (.23) for an expression of the norm of the first order derivative and apply Lemma 20 to upper bound its 4th power:

$$\left| \nabla g_0 \right|^4 = (2H-1)^3 \left( \sum_{k=1}^{H} \left| \nabla h_{\mathcal{G}_k}(y) \right|^4 + \sum_{k=1}^{H-1} \left| h_{\mathcal{G}_k}(y) - h_{\mathcal{G}_H}(y) \right|^4 \right)$$

$$\leq (2H-1)^3 \left( \sum_{k=1}^{H} \left| \nabla h_{\mathcal{G}_k}(y) \right|^4 + 8 \sum_{k=1}^{H-1} \left| h_{\mathcal{G}_k}(y) \right|^4 + 8(H-1) \left| h_{\mathcal{G}_H}(y) \right|^4 \right)$$

by Lemma 20 (.26)

Next we show $F_0 \left( \frac{\| h_{\mathcal{G}_0} \|_4}{f_0} \right) < \infty$. $f_0(y)$ can be viewed as a finite mixture of $h_{\mathcal{G}_k}(y)$, we can re-express $f_0(y)$ as $f_0(y) = \sum_{j=1}^{J} \pi_j^0 h_{\mathcal{G}_j}(y)$.

$$F_0 \left( \frac{\| h_{\mathcal{G}_0} \|_4}{f_0^4} \right)$$

$$= F_0 \left( \frac{h_{\mathcal{G}_0}^4}{(\sum_{j=1}^{J} \pi_j^0 h_{\mathcal{G}_j}(y))^4} \right)$$

$$< F_0 \left( \frac{h_{\mathcal{G}_0}^4}{\sum_{j=1}^{J} (\pi_j^0)^4 h_{\mathcal{G}_j}^4(y)} \right)$$

$$= F_0 \left( \frac{1}{(\pi_k^0)^4 + \sum_{j \neq k} (\pi_j^0)^4 h_{\mathcal{G}_j}^4(y)} \right)$$

$$< F_0(1/(\pi_k^0)^4) = 1/(\pi_k^0)^4 < \infty.$$

In addition, we have $F_0 \left( \frac{\| \nabla h_{\mathcal{G}_0} \|_4}{f_0} \right) < \infty$. Hence, (.26) is less than infinity, meaning $F_0 \left( \frac{\| \nabla g_0 \|_4}{f_0} \right) < \infty$.

D. Next we show

$$F_0 \left\{ \sup_{|\zeta - \zeta_0| \leq \delta^0} (|D^2 g_\zeta|^2) \right\} < \infty.$$  

Taking second order derivatives involves two partial derivatives. We discuss them by cases.
1. One w.r.t. \( \vartheta_k \) and the other w.r.t \( \rho_{\ell k'} \).

\[
\frac{\partial^2 g_\zeta}{\partial \vartheta_k \partial \rho_{\ell k'}} = \begin{cases} 
0 & k \neq k' \\
\nabla h_{\vartheta_k} & k = k'.
\end{cases}
\]

2. Both w.r.t weights. \( \frac{\partial^2 g_\zeta}{\partial \rho_{\ell k} \partial \rho_{\ell k'}} = 0 \) for both \( k \neq k' \) and \( k = k' \).

3. Both w.r.t component parameters

\[
\frac{\partial^2 g_\zeta}{\partial \vartheta_k \partial \vartheta_{k'}} = \begin{cases} 
0 & k \neq k' \\
D^2 h_{\vartheta_k} & k = k'.
\end{cases}
\]

Therefore, \( |D^2 g_\zeta| = \sum_{k=1}^{H} |D^2 h_{\vartheta_k}| + 2 \sum_{k=1}^{H} |\nabla h_{\vartheta_k}|. \)

\[
|D^2 g_\zeta|^2 = \left( \sum_{k=1}^{H} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|) \right)^2 
\]

\[
= \sum_{k=1}^{H} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|)^2 + 2 \sum_{k<k'} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|)(|D^2 h_{\vartheta_{k'}}| + 2|\nabla h_{\vartheta_{k'}}|) 
\]

\[
\leq \sum_{k=1}^{H} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|)^2 + \sum_{k<k'} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|)^2 + (|D^2 h_{\vartheta_{k'}}| + 2|\nabla h_{\vartheta_{k'}}|)^2 
\]

by AM-GM inequality

\[
\leq 3 \sum_{k=1}^{H} (|D^2 h_{\vartheta_k}| + 2|\nabla h_{\vartheta_k}|)^2 
\]

\[
= \sum_{k=1}^{H} 3|D^2 h_{\vartheta_k}|^2 + 12|\nabla h_{\vartheta_k}|^2 + 12D^2 h_{\vartheta_k} \nabla h_{\vartheta_k} 
\]

\[
\leq \sum_{k=1}^{H} 3|D^2 h_{\vartheta_k}|^2 + 12|\nabla h_{\vartheta_k}|^2 + 6|D^2 h_{\vartheta_k}|^2 + 6|\nabla h_{\vartheta_k}|^2 \quad \text{by AM-GM inequality} 
\]

\[
= \sum_{k=1}^{H} 9|D^2 h_{\vartheta_k}|^2 + 18|\nabla h_{\vartheta_k}|^2 
\]

We know that \( F_0 \{ \sup_{|\vartheta_k| \leq \delta} (|D^2 h_{\vartheta_k}|^2) \} < \infty \) and \( F_0 \{ \sup_{|\vartheta_k| \leq \delta} (|\nabla h_{\vartheta_k}|) \} < \infty \), Using (.21), we
can upper bound $F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0^p| \leq \delta'} (|D^2 g_{\zeta}|^2)}{g_{\zeta_0}^2} \right\}$ as follows:

$$F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0^p| \leq \delta'} (|D^2 g_{\zeta}|^2)}{g_{\zeta_0}^2} \right\} \leq F_0 \left\{ \frac{\sum_{k=1}^{H} 9 |D^2 h_{\vartheta_k}|^2 + 18 \sup_{|\zeta - \zeta_0^p| \leq \delta} |\nabla h_{\vartheta_k}|^2}{g_{\zeta_0}^2} \right\} \leq F_0 \left\{ \frac{\sum_{k=1}^{H} 9 \sup_{|\zeta - \zeta_0^p| \leq \delta} |D^2 h_{\vartheta_k}|^2 + 18 \sup_{|\zeta - \zeta_0^p| \leq \delta} |\nabla h_{\vartheta_k}|^2}{\sum_{k=1}^{H} (\rho_{tk}^0 - \delta \ell)^2 \inf_{|\vartheta_k^p - \vartheta_k^0| \leq \delta} h_{\vartheta_k}^2} \right\} \leq F_0 \left\{ \frac{\sum_{k=1}^{H} 9 \sup_{|\zeta - \zeta_0^p| \leq \delta} |D^2 h_{\vartheta_k}|^2 + 18 \sup_{|\zeta - \zeta_0^p| \leq \delta} |\nabla h_{\vartheta_k}|^2}{\sum_{k=1}^{H} (\rho_{tk}^0 - \delta \ell)^2 \inf_{|\vartheta_k^p - \vartheta_k^0| \leq \delta} h_{\vartheta_k}^2} \right\} = \sum_{k=1}^{H} \frac{9}{(\rho_{tk}^0 - \delta \ell)^2} F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0^p| \leq \delta} |D^2 h_{\vartheta_k}|^2}{h_{\vartheta_k}^2} \right\} + \frac{18}{(\rho_{tk}^0 - \delta \ell)^2} F_0 \left\{ \sup_{|\zeta - \zeta_0^p| \leq \delta} |\nabla h_{\vartheta_k}|^2 \right\} < \infty.

E. Next we show $F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0^p| \leq \delta'} (|D^3 g_{\zeta}|)}{g_{\zeta_0}^2} \right\} < \infty$. Based on the discussion of second order derivatives in $D$, we know that the third order derivatives is non-zero for only 2 cases:

1. Two w.r.t $\vartheta_k$ and one w.r.t $\rho_{tk}$.

$$\frac{\partial^3 g_{\zeta}}{\partial \vartheta_k^2 \partial \rho_{tk}} = D^2 h_{\vartheta_k}.$$

2. All w.r.t $\vartheta_k$.

$$\frac{\partial^3 g_{\zeta}}{\partial \vartheta_k^3} = D^3 h_{\vartheta_k}.$$

Therefore,

$$|D^3 g_{\zeta}| = \sum_{k=1}^{H} |D^2 h_{\vartheta_k}| + \sum_{k=1}^{H} |D^3 h_{\vartheta_k}|.$$

Since we know $F_0 \left\{ \frac{\sup_{|\vartheta - \vartheta_0^p| \leq \delta} (|D^2 h_{\vartheta}|^2)}{h_{\vartheta_0}^2} \right\} < \infty$, we have $F_0 \left\{ \frac{\sup_{|\vartheta - \vartheta_0^p| \leq \delta} (|D^2 h_{\vartheta}|^2)}{h_{\vartheta_0}^2} \right\} < \infty$. Since we also know that $F_0 \left\{ \frac{\sup_{|\vartheta - \vartheta_0^p| \leq \delta} (|D^3 h_{\vartheta}|^2)}{h_{\vartheta_0}^2} \right\} < \infty$ and inequality (.21), we can show $F_0 \left\{ \frac{\sup_{|\zeta - \zeta_0^p| \leq \delta'} (|D^3 g_{\zeta}|)}{g_{\zeta_0}^2} \right\} < \infty$ using the same method we followed in A, B and D. 

66
Assumption 3

Lemma 22. **OF-CLOG satisfies Assumption 3.**

*Proof.* We want to show there exists \( \Gamma_0 \subset \Gamma \) satisfying \( \text{Leb}(\Gamma_0) > 0 \) and for all \( \ell \leq K_0 \)

\[
d(\zeta_0^\ell, \Gamma_0) = \inf_{\zeta \in \Gamma_0} |\zeta - \zeta_0^\ell| > 0
\]

and such that for all \( \zeta \in \Gamma_0 \),

\[
F_0 \left( \frac{g^4_0}{f^4_0} \right) < \infty, \quad F_0 \left( \frac{g^3_0}{f^3_0} \right) < \infty, \forall \ell \leq K_0.
\]

We know that there exists \( \Theta_0 \subset \Theta \) satisfying \( \text{Leb}(\Theta_0) > 0 \) and for all \( k \leq H \),

\[
d(\vartheta_0^k, \Theta_0) = \inf_{\vartheta \in \Theta_0} |\vartheta - \vartheta_0^k| > 0
\]

and such that for all \( \vartheta \in \Theta_0 \),

\[
F_0 \left( \frac{h^4_\vartheta}{f^4_0} \right) < \infty, \quad F_0 \left( \frac{h^3_\vartheta}{h^3_0} \right) < \infty, \forall k \leq H.
\]

Let \( \Gamma_0 = (0, 1)^H \times \Theta_0^H \). By Lemma 20, \( g^4_\zeta = \left( \sum_{k=1}^{H} \rho_k h_{\vartheta_k}(y) \right)^4 \leq H^3 \sum_{k=1}^{H} \rho^4_k h^4_{\vartheta_k}(y) \), \( g^3_\zeta = \left( \sum_{k=1}^{H} \rho_k h_{\vartheta_k}(y) \right)^3 \leq H^2 \sum_{k=1}^{H} \rho^3_k h^3_{\vartheta_k}(y) \). By (.21), \( \sum_{k=1}^{H} (\rho^0_{\ell k} - \delta_{\ell k}) \inf_{|\vartheta'_{k} - \vartheta_{k,0}^\ell| \leq \delta} h^3_{\vartheta'_{k}}(y) \).

Therefore,

\[
F_0 \left( \frac{g^4_0}{f^4_0} \right) \leq H^3 \sum_{k=1}^{H} \rho^4_k F_0 \left( \frac{h^4_{\vartheta_k}}{f^4_0} \right) < \infty.
\]

\[
F_0 \left( \frac{g^3_0}{g^3_{\zeta_0^\ell}} \right) \leq H^2 \sum_{k=1}^{H} \rho^3_k F_0 \left( \frac{h_{\vartheta_k}^3}{\sum_{k=1}^{H} (\rho^0_{\ell k} - \delta_{\ell k}) \inf_{|\vartheta'_{k} - \vartheta_{k,0}^\ell| \leq \delta} h^3_{\vartheta'_{k}}} \right)
\]

\[
< H^2 \sum_{k=1}^{H} \rho^3_k \frac{h_{\vartheta_k}^3}{\rho^0_{\ell k} - \delta_{\ell k}} F_0 \left( \frac{h^3_{\vartheta_k}}{h_{\vartheta_k}^3} \right) < \infty.
\]

\[ \square \]

Assumptions 4 and 5

Our model does not satisfy Assumption 4 in RM, because when the variance associated with one mixture component of the inner layer goes to zero, \( \int g(x) d\mu(x) \in (0, 1) \), whereas Assumption 4
in RM only allows \( \int g(x) d\mu(x) \in \{0, \infty\} \). We henceforth extend the assumption and prove that it leads to the same consistency result (i.e. Theorem 1) as that in RM.

**Proof.** We show that the extended assumption together with Assumptions 1, 2, 3 and 5 in RM are sufficient for Theorem 1 in RM. Following the proof in Appendix A of RM up until P.707 where \( \Gamma \) is not compact, since our modification only applies to the non-compact cases. If for any converging subsequences of \( \theta_n \) to a point in \( \Theta_0 \) for which all components parameters \( \zeta_j \) belong to \( \Gamma \) or for which all components \( \zeta_j \) correspond to a probability density we can follow the argument in the original proof. For subsequences that do not converge to such a point, or in other words \( \zeta_n \) converges to a point in \( \bar{\Xi}_0 \), where at least one of the components’ parameters belong to \( \partial \bar{\Gamma} \), where \( \partial \bar{\Gamma} = \{ \zeta \in \partial \Gamma; \int g_\zeta(x) d\mu(x) \in (0, 1) \cup \infty \} \). Let \( J = \{ j \leq k; \zeta_{j,n} \to \partial \bar{\Gamma} \} \neq \emptyset \). By definition of \( t, J \subset \{ t_{k_0} + 1, \ldots, k \} \) and choosing \( \sigma \) accordingly we can write \( J = \{ k_1, \ldots, k \} \) with \( k_1 \geq t_{k_0} + 1 \). Hence, for all \( j < k_1 \), there exists \( \zeta_j \in \Gamma \) such that \( \zeta_{j,n} \to \zeta_j \). We split \( L'_{w} \) into \( L'_{n,(1)} \) and \( L'_{n,(2)} \) where \( L'_{n,(2)} = (g_{\zeta_{n,j}} - g_{\zeta_{n,j_0}}, j = k_1, \ldots, k) \) and by definition of \( k_1, L'_{n,(1)} \) converges to \( L'_{\infty,(1)} \) so inequality (9) in RM, i.e.

\[
|(1 - \eta_{w,n})w(\zeta_{r,n})^T L'_{r,n} + \eta_{w,n} w(\lambda_{r,n})^T L'' w(\lambda_{r,n})| \leq c_n,
\]

becomes, in the limit,

\[
|(1 - \eta)w_{(1)}^T L'_{\infty,(1)} + (1 - \eta)w_{(2)}^T L'_{n,(2)} + \eta w(\lambda)^T L'' w(\lambda)|_1 \to 0 \quad (27)
\]

as \( n \to \infty \), where the only term depending on \( n \) is \( L'_{n,(2)} \). If \( \eta < 1 \) then expression (27) can be written as follows: there exists \( h \) integrable such that

\[
\lim_{n \to \infty} \left\{ \left| k-k_{1}+1 \sum_{j=1}^{k-k_{1}+1} w_{(2)}(\zeta_{j}) g_{\zeta_{j+k_{1}-1,n}} - h \right|_1 \right\} = 0;
\]

if \( w_{(2)}(\zeta) \neq 0 \) then set \( \bar{w}_2 = \sum_l w_{(2),l} \) and, since \( w_{(2),l} \geq 0 \) for all \( l \), then expression (27) can be expressed as

\[
\left| k-k_{1}+1 \sum_{j=1}^{k-k_{1}+1} p_{j} g_{\zeta_{j+k_{1}-1,n}} - h^*/(1 - \bar{w}_2) \right|_1 \to 0, \quad p_{j} = w_{(2),j}/\bar{w}_2. \quad (28)
\]

Thus \( h^*/(1 - \bar{w}_2) \) is a probability density and \( \sum_{j=1}^{k-k_{1}+1} p_{j} g_{\zeta_{j+k_{1}-1,n}} \) converges towards a proper probability density which contradicts the definition of \( J \). Hence \( w_{(2)} = 0 \) and we can apply the
same arguments as in the compact case to conclude. If \( \eta = 1 \), then we can use the same argument as in the compact case since \( L'_{n,(2)} \) has no influence. The remaining proof is exactly the same as that for Theorem 1 in RM.

**Lemma 23.** OF-CLOG satisfies Assumption 4.

**Proof.** First we show the first part of Assumption 4, which is

\[
(\phi_t - \phi^0_t)^T f'_{\phi^0_t, \psi_t} + \frac{1}{2} (\phi_t - \phi^0_t)^T f''_{\phi^0_t, \psi_t} (\phi_t - \phi^0_t) = 0 \iff \\
\forall i \leq k_0, s_i = 0 \quad \text{and} \quad \forall j \in I_i \quad q_j (\zeta_j - \zeta^0_j) = 0, \quad \forall i \geq t_{k_0} + 1 \quad \lambda_i = 0.
\]

The first-order derivative is

\[
f'_{\phi^0_t, \psi_t} = \left( \lambda_1 g^T_{\zeta_1}, \ldots, \lambda_{t_{K_0}} g^T_{\zeta_{t_{K_0}}}, \sum_{j \in I_1} g_{\zeta_j} \frac{\partial \lambda_j}{\partial s_1}, \ldots, \sum_{j \in I_{t_{K_0} - 1}} g_{\zeta_j} \frac{\partial \lambda_j}{\partial s_{t_{K_0} - 1}}, g_{\zeta_{t_{K_0} + 1}}, \ldots, g_{\zeta_{K_2}} \right)^T,
\]

We notice that the first-order derivative can be grouped into three blocks according to \((\zeta_j)_{j=1}^{t_{K_0}}, (s_i)_{i=1}^{K_0-1}, (\lambda_j)_{j=t_{K_0}+1}^{K_2}\). Thereupon, the second-order derivative can be represented by

\[
\begin{pmatrix}
B_{11} & B_{12} & B_{13} \\
B_{21} & B_{22} & B_{23} \\
B_{31} & B_{32} & B_{33}
\end{pmatrix},
\]

where \(B_{ij}\) represents the partial derivative with respect to \(Bj\) and \(Bi\) respectively. \(B_{11}\) is a block-diagonal matrix of dimension \(dK \times dK\) with the \(j\)th block being \(\lambda_j g''_{\zeta_j}, j = 1, \ldots, t_{K_0}\), or in other words \(B_{11} = \sum_j \lambda_j (e_j e_j^T \otimes g''_{\zeta_j})\), where \(e_j\) is the \(j\)th standard basis vector in \(\mathbb{R}^K\).

\(B_{12} = B_{21}^T\). \(B_{12}\) is a \(dK \times (K_0 - 1)\) block diagonal matrix, with the \(i\)th diagonal block being \(\sum_{j \in I_i} (\tilde{e}_j e_i)^T \otimes (g''_{\zeta_j} \frac{\partial \lambda_j}{\partial s_i})\), where \(\tilde{e}_j\) is the \(j\)th standard basis vector in \(\mathbb{R}^{t_i-1}\), and \(e_i\) is the \(i\)th standard basis vector in \(\mathbb{R}^{K}\). \(B_{22}\) is a \((K_0 - 1) \times (K_0 - 1)\) diagonal matrix, with the \(i\)th diagonal term being \(\sum_{j \in I_i} g_{\zeta_j} \frac{\partial^2 \lambda_j}{\partial s_i^2}\). The other blocks only contain zeros.

Evaluating \(f'_{\phi^0_t, \psi_t}\) and \(f''_{\phi^0_t, \psi_t}\) at \(\phi_t = \phi^0_t\) gives
\[ f'_{\phi_t, \psi_t} = (\lambda_0, \ldots, \lambda_{t_1})^T \otimes g'^{T}_{\zeta_1}, \ldots, (\lambda_{t_{K_0} - 1}, \ldots, \lambda_{t_{K_0}})^T \otimes g'^{T}_{\zeta_{K_0}} \]
\[ g_{\zeta_0}, \ldots, g'_{\zeta_{K_0}}, \ldots, g'_{\zeta_{K_0}} \]

and \( B_{11} = \sum_j \lambda_j (e_j e_j^T \otimes g''_{\zeta_0}) \), the \( i \)th block in \( B_{12} \) is \( \sum_{j \in I_i} (e_j e_j^T) \otimes (g'_{\zeta_0} \frac{\partial \lambda_j}{\partial s_i}) \big|_{s_i = 0} \), and \( B_{22} \) only contains zero entries.

Therefore,
\[ (\phi_t - \phi^0_t)^T f'_{\phi_t, \psi_t} = \sum_{i=1}^{K_0} \sum_{j \in I_i} \lambda_j (\zeta_j - \zeta_0^0)^T g'_{\zeta_i} + \sum_{i=1}^{K_0 - 1} s_t g'_{\zeta_0} + \sum_{j=t_{K_0} + 1}^{K} \lambda_j g_{\zeta_j}, \]

and
\[ (\phi_t - \phi^0_t)^T f''_{\phi_t, \psi_t} (\phi_t - \phi^0_t) = \sum_{i=1}^{K_0} \sum_{j \in I_i} \lambda_j (\zeta_j - \zeta_0^0)^T g''_{\zeta_i} (\zeta_j - \zeta_0^0) + 2 \sum_{i=1}^{K_0 - 1} \sum_{j \in I_i} (\zeta_j - \zeta_0^0)^T \left( g'_{\zeta_i} \frac{\partial \lambda_j}{\partial s_i} \right) \big|_{s_i = 0} s_t. \]

Therefore,
\[ (\phi_t - \phi^0_t)^T f''_{\phi_t, \psi_t} + \frac{1}{2}(\phi_t - \phi^0_t)^T f''_{\phi_t, \psi_t} (\phi_t - \phi^0_t) = \sum_{i=1}^{K_0 - 1} s_t g'_{\zeta_i} + \sum_{j=t_{K_0} + 1}^{K} \lambda_j g_{\zeta_j} + \sum_{i=1}^{K_0 - 1} \sum_{j \in I_i} (\zeta_j - \zeta_0^0)^T g'_{\zeta_i} \frac{\partial \lambda_j}{\partial s_i} \big|_{s_i = 0} s_t + \frac{1}{2} \sum_{i=1}^{K_0} \sum_{j \in I_i} \lambda_j (\zeta_j - \zeta_0^0)^T g''_{\zeta_i} (\zeta_j - \zeta_0^0). \]

\[ \Rightarrow \] The fact that (27) equals to 0 everywhere means the terms associated with \( g^0_{\zeta_i}, g_{\zeta_j}, g'_{\zeta_i} \) and \( g''_{\zeta_i} \) are 0. To make the term \( \frac{1}{2} \sum_{i=1}^{K_0} \sum_{j \in I_i} \lambda_j (\zeta_j - \zeta_0^0)^T g''_{\zeta_i} (\zeta_j - \zeta_0^0) = 0 \) everywhere, \( \lambda_j (\zeta_j - \zeta_0^0) = 0 \), or in other words \( g_j (\zeta_j - \zeta_0^0) = 0 \) \( \forall i \leq K_0, \forall j \in I_i \). To make the term \( \sum_{i=1}^{K_0 - 1} s_t g'_{\zeta_i} = 0 \) everywhere, \( s_t = 0 \) \( \forall i \leq K_0 \). In addition, to make the term \( \sum_{j=t_{K_0} + 1}^{K} \lambda_j g_{\zeta_j} = 0 \) everywhere, \( \lambda_j = 0 \) \( \forall j \geq t_{K_0} + 1 \). \[ \Leftarrow \] Since \( g_j (\zeta_j - \zeta_0^0) = 0 \) implies \( \lambda_j (\zeta_j - \zeta_0^0) \), we have \( \frac{1}{2} \sum_{i=1}^{K_0} \sum_{j \in I_i} \lambda_j (\zeta_j - \zeta_0^0)^T g''_{\zeta_i} (\zeta_j - \zeta_0^0) = 0 \) everywhere. With the other conditions, we easily see that (27) is zero everywhere.

\[ \textbf{Lemma 24.} \ \textit{OF-CLOG satisfies Assumption 5.} \]
Proof. The prior on $\lambda$ is Dirichlet with density

$$B(\alpha, \ldots, \alpha) \lambda_1^{\alpha-1} \cdots \lambda_k^{\alpha-1},$$

where $B(\alpha, \ldots, \alpha) = \frac{\Gamma^K(\alpha)}{\Gamma(\alpha \cdot K)}$, which satisfies the requirement of assumption 5.

.4 Posterior update for $(I_h^{(j)}, \gamma_h^{(j)})$

We want to integrate out $\gamma_h^{(j)}$. Assume that $j \in I_c$ so that we have a parameter $\gamma_h^{(j)}$. We have

$$P(y_{ij} = k \mid I_h^{(j)}, s_{ij} = h) = \int P(y_{ij} = k \mid \gamma_h^{(j)}, s_{ij} = h)p(\gamma_h^{(j)} \mid I_h^{(j)}) \, d\gamma_h^{(j)}$$

$$= \int \gamma_h^{(j)} p(\gamma_h^{(j)} \mid I_h^{(j)}) \, d\gamma_h^{(j)}$$

$$= \begin{cases} 1/d_j & I_h^{(j)} = 1 \\ e_{hk} & I_h^{(j)} = 0 \end{cases}$$

$$= I_h^{(j)}/d_j + (1 - I_h^{(j)})e_{hk}.$$

The joint full conditional for $I_h^{(j)}$ and $\gamma_h^{(j)}$ is

$$p(I_h^{(j)}, \gamma_h^{(j)} \mid -) \propto p(\gamma_h^{(j)} \mid I_h^{(j)})p(I_h^{(j)} \mid -),$$

where $p(I_h^{(j)} \mid -)$ is the distribution conditional on everything except $\gamma_h^{(j)}$. We have used that $\gamma_h^{(j)}$ is conditionally independent of everything else given $I_h^{(j)}$. The distribution of $I_h^{(j)}$ given everything satisfies

$$P(I_h^{(j)} = 1 \mid -) \propto P(I_h^{(j)} = 1 \mid \epsilon) \prod_{i: s_{ij} = h} P(y_{ij} = k_i \mid I_h^{(j)} = 1) = \frac{\epsilon}{d_j^{\{i: s_{ij} = h\}}}$$

$$P(I_h^{(j)} = 0 \mid -) \propto P(I_h^{(j)} = 0 \mid \epsilon) \prod_{i: s_{ij} = h} P(y_{ij} = k_i \mid I_h^{(j)} = 0) = (1 - \epsilon) \prod_{i: s_{ij} = h} e_{hk_i}. $$

Note that if any observation takes a value $k_i$ that is not in the support of $e_{h}$, then we have $P(I_h^{(j)} = 1 \mid -) = 1$.

.5 Additional results for the applications
**Figure 14:** Supplementary clustering results of the diamond data in the dimensions of size and certification colored by the clustering estimates.
Figure .15: Clustering results of the diamond data in the dimensions of size and clarity colored by the clustering estimates.
Chapter 2

Distributed Bayesian Clustering using Finite Mixture of Mixtures

Key words: Distributed algorithm, Model-based clustering, Bayesian methods, Markov chain Monte Carlo, Loss function

2.1 Introduction

Recent technological advances have greatly accelerated data collection processes, leading to explosively growing data sizes. These data sets are often too big to be stored on a single computer and too costly to move across computers. One common query to these data sets is cluster analysis, which seeks to group observations that are cohesive and separated from other groups. Large scale data sets from astronomy, flow cytometry and many other fields raise questions as to how to discover underlying clusters quickly while allowing for statistical inference.

To cluster large scale data sets, parallel and distributed clustering algorithms have been proposed. A common procedure underlying these algorithms is splitting the data into subsets and determining cluster assignments independently for each subset. Because clusters describe inherent relationships among all the data points, the independent local clustering must be carefully adjusted on the global scale via communication of local results.

Most such algorithms are based on either distance or density without a likelihood specification; examples include density based distributed clustering (Januzaj et al., 2004), K-Means with Map-Reduce (PKMeans) (Zhao et al., 2009) and co-clustering with Map-Reduce (Papadimitriou and Sun, 2008). These methods do not, in general, have established statistical properties. An alternative method is model-based clustering, which considers the data as coming from a mixture distribution with different distributions for each cluster. Unlike the aforementioned methods, model-based clustering uses a soft assignment, where each data point has a probability of belonging to each cluster; it also allows density estimation and other statistical inference.
One commonly used framework for model-based clustering is finite mixtures. Let \( Y = (y_1, \ldots, y_N) \), \( y_i \in \mathbb{R}^d \) be a sample of size \( N \). A finite mixture model assumes that \( y_i \) \((i = 1, \ldots, N)\) of dimension \( d \) is generated from a finite mixture with \( K \) exchangeable mixture components:

\[
f(y_i \mid \Theta, \eta) = \sum_{k=1}^{K} \eta_k f_k(y_i \mid \theta_k), \quad \Theta = (\theta_1, \ldots, \theta_K),
\]

(2.1)

where \( \eta_k \) is the weight associated with component \( k \) satisfying \( \sum_{k=1}^{K} \eta_k = 1 \) and \( f_k(y_i \mid \theta_k) \) is the component density specified by parameter \( \theta_k \). With each component interpreted as a cluster, this model has been successfully applied to many areas, including agriculture, astronomy, bioinformatics, biology, economics, engineering, genetics, etc.

Finite mixture models often require a predetermined \( K \), but the number of clusters is generally unknown. Even though one can fit multiple models with different \( K \) and identify the best one based on model selection criteria (e.g. BIC), such a procedure can be time-consuming given a large data set and less appealing than the natural Bayesian approach, which is to treat the true number of clusters \( K_{true} \) as an unknown parameter to be estimated jointly with the component-specific parameters. Models that adopt this Bayesian approach include Bayesian nonparametric (BNP) mixture models and overfitted finite mixture models, where the number of clusters is automatically inferred from the observed data.

BNP mixture models have been widely used for clustering in topic modelling (Ge et al., 2015) and biomedical applications, because the model complexity adapts to the increasing amount of data. In particular, the number of clusters grows with sample size. Although this is a conceptually appealing property, in massive datasets this can lead to an enormous number of clusters. This in turn creates an associated large computational burden and decreases interpretability and data simplification, two of the primary goals of clustering. In addition, the inferred clusters may not represent actually distinct groups in the data, as extra clusters can arise as an artifact of BNP priors and due to inadequacies of typical kernels (e.g., Gaussian) in describing cluster shapes. To address the computational problem, there is a rich and growing literature on scalable algorithms, using sequential approximations (Wang and Dunson, 2011; Lin, 2013; Tank et al., 2015) and parallelization.

One common strategy underlying some parallel algorithms (Williamson et al., 2013; Dubey et al., 2014; Ge et al., 2015) is exploiting conditional independence of cluster allocation given all
other parameters to run Markov chain Monte Carlo (MCMC) in parallel. For example, Williamson et al. (2013) proposed parallel inference for Dirichet process (DP) mixture and hierarchical DP mixture models through re-parametrisation of a DP mixture model as a mixture of DPs. They further assume that each cluster only resides on one processor, which means that conditional on the processor allocations, the data points are distributed according to independent Dirichlet processes, facilitating parallel draws of the local cluster assignment. A global procedure to ensure that each cluster indeed resides on a single processor is run at every iteration and can require immense data transfer in a distributed system. This assumption can also lead to load imbalance if the cluster sizes are not uniform. Based on an alternative representation of a DP random measure, Ge et al. (2015) developed a slice sampler under the Map-Reduce framework for the same models, which is better suited for a distributed system. Summary statistics, instead of raw data, are transmitted from every mapper to a reducer for drawing global parameters at every iteration. Such communication, given the poor mixing and slow convergence of MCMC in the presence of latent variables, can be expensive.

Other more efficient parallel algorithms focus on approximate inference under a BNP model; see, for example, distributed algorithms SNOB and SIGN by Zuanetti et al. (2019) and Ni et al. (2020) respectively. Instead of communicating at every iteration, they draw samples of cluster assignments locally, determine an optimal clustering estimate for each subset and then communicate sufficient statistics and clustering results for adjustment. SIGN uses a multi-step approach to local clustering adjustment; one deficiency is that if clusters are incorrectly merged at some step, then there is no hope of recovering the true clustering structure because they can never be split. In addition, both algorithms use a loss function multiple times at different stages to arrive at a final clustering estimate, raising the question as to whether the final clustering is actually a good approximation to the minimizer of the posterior expected loss.

Despite the proliferation of fast inference algorithms for BNP mixtures, there have been few similar advances for finite mixtures. Our view is that finite mixtures provide a more practically reasonable framework for clustering in massive datasets. Most BNP approaches, including widely used DP mixtures and Pitman-Yor (PY) process mixtures, carry an implicit assumption that as the
sample size goes to infinity, the number of clusters inevitably tends to infinity. For the DP mixtures $K_{true} \sim \alpha \log(N)$ (Korwar and Hollander, 1973), while for PY mixtures $K_{true} \sim N^\beta$ (Miller and Harrison, 2014; Orbanz, 2014), where $\alpha$ and $\beta \in [0,1)$ are constants. This means the posterior of DP or PY mixtures fails to concentrate at the true number of components for data from a finite mixture.

In contrast, overfitted finite mixtures apply a finite mixture model with the number of components $K$ intentionally set to be greater than the true number of clusters $K_{true}$. This approach is more useful for data with a moderate number of clusters that does not increase as the number of observations $N$ increases. Setting the prior on the mixture weights $\eta$ to be Dir($\epsilon_0$), Rousseau and Mengersen (2011) studied the asymptotic behavior of its posterior distribution: if $\epsilon_0 < d/2$, where $d$ is the dimension of the cluster-specific parameters $\theta_k$, the posterior expectation of the weights associated with empty clusters asymptotically converges to zero. Therefore the true number of clusters can be identified asymptotically despite the identifiability problems inherent in mixture models.

A critical component of any mixture model specification is the choice of kernel. Although the Gaussian distribution is typically used, most data clusters in real world applications deviate from such a simple symmetric choice. Such misspecification can lead to identification of extraneous clusters as a means to improve model fit, which essentially destroys the interpretation of a mixture component as one cluster. A remedy is to instead model each cluster by a finite Gaussian mixture, a model that can accurately approximate a wide class of distributions (Malsiner-Walli et al., 2017) (henceforth MWFSG). Formally, this means each component distribution $f_k(y_i \mid \theta_k)$ in (2.1) is assumed to be a mixture of $L$ Gaussian subcomponents:

$$f_k(y_i \mid \theta_k) = \sum_{l=1}^L \omega_{kl} p(y_i \mid \mu_{kl}, \Sigma_{kl}),$$  

(2.2)

where $\theta_k = \{\omega_{kl}, \mu_{kl}, \Sigma_{kl}\}_{l=1}^L$ and $p(y_i \mid \mu_{kl}, \Sigma_{kl})$ are Gaussian densities used to approximate $f_k(y_i \mid \theta_k)$. Setting $K$ in (2.1) to be an upper bound of the number of clusters in the data yields a model called overfitted finite mixture of Gaussian mixtures. This model encounters identifiability issues due to exchangeability of all the subcomponents; fortunately MWFSG developed prior specification for this model that encourages subcomponents within a cluster to be close and clusters to
be spread out, minimizing these issues.

In this article we propose a distributed Bayesian inference method based on (overfitted) finite mixture of Gaussian mixtures for clustering, which we refer to as DIB-C. Our main contributions lie in developing a decision theoretic approach to identifying a reliable clustering estimate while minimizing data transmission between the master and workers, a key consideration in distributed computing. We adopt a strategy used in SNOB and SIGN, which is to produce MCMC samples of local clustering assignments in an embarrassingly parallel manner to minimize data communication. Unlike SNOB or SIGN, our adjustment to local clusters permits both cluster merging and splitting, and our clustering estimate is more reliable as we only apply a loss function once in the entire framework to the samples of adjusted local clusterings. These steps are enabled by one of the simplest parallel programming paradigm, master-worker, and the communication of summary statistics at some iterations between the master and workers.

In addition to clear computational gains, DIB-C exhibits superior clustering performance in comparison to its non-distributed counterpart. In addition, DIB-C can accommodate any loss function on the space of partitions for cluster estimation and enables density estimation, quick classification of new subjects, sampling from the posterior predictive distribution and uncertainty quantification of cluster-specific parameters (such as cluster centers). As a side effect, DIB-C also works for semi-supervised clustering: when the true number of clusters is greater than that represented in the labeled data, DIB-C can automatically determine the number of clusters via the use of an overfitted finite mixture model.

The rest of the paper is organized as follows. In Section 2.2, we review the model and prior specification of finite mixture of mixtures. In Section 2.3, we describe the DIB-C framework. Section 2.4 presents extensive experimental results to illustrate the performance of our framework.


2.2 Model Specification: Finite Mixture of Gaussian Mixtures

Clusters are groups of data points that are cohesive and connected within a group but are separated from other groups. Assume \( \mathcal{Y} \) can be partitioned into \( R \) non-overlapping subsets, with subset \( r \), denoted by \( \mathcal{Y}_r \), residing on worker \( r \) (\( r = 1, \ldots, R \)). The model formulation, as defined by (2.1) and (2.2), leads to a hierarchy: the upper level (2.1) captures a heterogeneous population with \( K \) different clusters and each cluster corresponds to a mixture component; the lower level (2.2) approximates each cluster distribution via a mixture of Gaussian densities \( p(y_i \mid \mu_{kl}, \Sigma_{kl}) \). To distinguish the upper and lower level components, we adopt the convention in MWFSG to call \( f_k(y_i \mid \theta_k) \) cluster distribution \( k \) and \( p(y_i \mid \mu_{kl}, \Sigma_{kl}) \) subcomponent distribution \( l \) in cluster \( k \).

Combining (2.1) and (2.2) provides an alternative expression of the likelihood of \( y_i \):

\[
f(y_i \mid \Theta, \eta) = \sum_{k=1}^{K} \sum_{l=1}^{L} \eta_{kl} p(y_i \mid \mu_{kl}, \Sigma_{kl}),
\]

(2.3)

where \( \eta_{kl} = \eta_k \omega_{kl} \). (2.3) is invariant to permutations of the \( K \cdot L \) subcomponents: exchanging subcomponents between clusters does not alter the likelihood, but goes contrary to the general characterization of data clusters, which is a densely connected cloud of data points far away from other densely connected ones. To incorporate such structure, MWFSG proposed a two-level hierarchical prior that repulses the cluster centers and attracts subcomponent means towards the cluster centers. Additionally, since cluster structure is invariant to the ordering of both clusters and subcomponents within a cluster, symmetric priors should be used for clusters on the upper and lower level, respectively, to ensure exchangeability.

Let \( \varphi_0 = (e_0, d_0, c_0, G_0, B_0, m_0, M_0, \nu) \) be a set of fixed hyper-parameters. The priors at the cluster level are specified so that the \( K \) clusters are exchangeable:

\[
p(\eta, \theta_1, \ldots, \theta_K \mid \varphi_0) = p(\eta \mid e_0) \prod_{k=1}^{K} p(\theta_k \mid \varphi_0),
\]

(2.4)

where \( \eta \mid e_0 \sim \text{Dir}_K(e_0) \), and \( \theta_k \mid \varphi_0 \) are independent and identically distributed a priori. Within
each cluster $k$, the prior distribution can be factored as:

$$p(\theta_k \mid \varphi_0) = p(w_k \mid d_0)p(\mu_{k1}, \mu_{k2}, \ldots, \mu_{kL} \mid B_0, m_0, M_0, \nu)p(\Sigma_{k1}, \Sigma_{k2}, \ldots, \Sigma_{kL} \mid c_0, g_0, G_0),$$  

(2.5)

where $\omega_k \mid d_0 \sim \text{Dir}_L(d_0)$. $\mu_{k1}, \ldots, \mu_{kL}$ are independently distributed conditional on $B_0, m_0, M_0, \nu$, and $\Sigma_{k1}, \ldots, \Sigma_{kL}$ are independent conditional on $c_0, g_0, G_0$.

To create the conditional independence in (2.4) and (2.5), MWFSG formulated hierarchical “random effects” priors: first the cluster-specific parameters $(C_0k, b_{0k}, \Lambda_k)$ are drawn from the same set of distributions and then, conditional on these, the subcomponent-specific parameters $(\mu_{kl}, \Sigma_{kl})_{l=1}^L$ within cluster $k$ are drawn from another set of distributions for all $k$.

Specifically, cluster-specific parameters $(C_0k, b_{0k})$ and $\Lambda_k = \text{diag}(\lambda_{k1}, \ldots, \lambda_{kd})$, $k = 1, \ldots, K$ are drawn from:

- $C_0k \mid g_0, G_0 \sim W_d(g_0, G_0)$,
- $b_{0k} \mid m_0, M_0 \sim N_d(m_0, M_0)$,
- $(\lambda_{k1}, \ldots, \lambda_{kd}) \mid \nu \sim \text{Ga}(\nu, \nu)$,

where $m_0$ is the overall data center. Cluster centers $b_{0k}$ are generated around $m_0$ with $M_0$ controlling the shrinkage of $b_{0k}$ towards $m_0$. MWFSG set $M_0 \gg S_Y$, where $S_Y$ is the sample covariance of all data so that cluster centers lie relatively far from each other.

Conditional on the cluster-specific random hyperparameters $(C_0k, b_{0k}, \Lambda_k)$ and the fixed lower level hyperparameters $(B_0, c_0)$, the $L$ subcomponent means $\mu_{kl}$ and covariance matrices $\Sigma_{kl}$ are drawn independently for all $l = 1, \ldots, L$:

$$\mu_{kl} \mid B_0, b_{0k}, \Lambda_k \sim N_d(b_{0k}, \sqrt{\lambda_k}B_0\sqrt{\lambda_k}),$$  

(2.6)

$$\Sigma_{kl}^{-1} \mid c_0, C_0k \sim W_d(c_0, C_0k).$$  

(2.7)

$\mu_{kl}$ ($l = 1, \ldots, L$) should be close to the cluster center $b_{0k}$ to ensure no gaps among subcomponents and $\Sigma_{kl}$ ($l = 1, \ldots, L$) should be diffuse so that the boundary or outlier points can be well fitted. Therefore, we need $B_0$ to impose strong shrinkage of $\mu_{kl}$’s toward $b_{0k}$ and $c_0$ to be small to induce large variances in $\Sigma_{kl}$’s while permitting large variation of $\Sigma_{kl}$’s.
To elicit the prior, MWFSG decomposes the variation of an observation \( y \) into three sources:

\[
\text{cov}(y) = \sum_{k=1}^{K} \eta_k \sum_k + \sum_{k=1}^{K} \eta_k \mu_k \mu_k' - \mu \mu' \\
\text{within cluster variation} \quad \text{between cluster variation}
\]

\[
= \sum_{k=1}^{K} \eta_k \sum_{l=1}^{L} \omega_{kl} \sum_{kl} + \sum_{k=1}^{K} \left( \sum_{l=1}^{L} \omega_{kl} \mu_k \mu_k' - \mu \mu' \right) \\
\text{within subcomponent variation} \quad \text{within cluster, between subcomponent variation}
\]

\[
+ \sum_{k=1}^{K} \eta_k \mu_k \mu_k' - \mu \mu' \\
\text{between cluster variation}
\]

\[
:= (1 - \phi_W)(1 - \phi_B)\text{cov}(y) + \phi_W (1 - \phi_B)\text{cov}(y) + \phi_B \text{cov}(y),
\]

where \( \phi_B \) is the proportion of variability explained by the cluster centers \( \mu_k \) around the overall mean \( \mu = \sum_{k=1}^{K} \eta_k \mu_k \), and \( \phi_W \) is the proportion of total variation explained by subcomponent means \( \mu_{kl} \) \((l = 1, \ldots, L)\) around cluster center \( \mu_k = \sum_{l=1}^{L} \omega_{wl} \mu_{kl} \), for all \( k \). By setting \( \phi_B = 0.5 \) and \( \phi_W = 0.1 \), we balance the variation from the three sources and can elicit prior parameters accordingly.

In this article, the MCMC sampling scheme is derived based on the above specified prior distributions, but our DIB-C framework is broadly applicable to any prior specification for a finite mixture of mixtures.

### 2.3 DIB-C Framework

The general idea of the DIB-C framework is to produce approximate MCMC samples of cluster assignments under the assumption of a finite mixture of mixtures model in a distributed computing paradigm. Specifically, we sample cluster assignments in each partition of the data and combine them for approximate samples of global cluster allocations.

Because clusters describe inherent relationships among all the data points, the independent clustering performed on each subset must be carefully merged. To minimize data transmission, we communicate only sufficient statistics between the master and workers to refine local cluster assignments across all partitions of the data in order to produce reliable global cluster allocations.
Figure 2.1: Algorithm flowchart of DIB-C. In step 2, the copies of frames refer to samples of local clustering, with the red ones representing those that are adjusted in step 3. Based on the adjusted samples of local clustering, a global clustering is estimated in step 4. Step 5 Sampling model parameters is excluded due to space constraint.
Based on the above discussion, we propose a DIB-C framework that consists of five steps: a. **partitioning**: randomly partitioning the data $Y$ and distributing them over $R$ workers; b. **sampling local clustering**: running embarrassingly parallel MCMC to obtain samples of subcomponent and cluster assignments on each partition of the data based on a finite mixture of mixtures model; c. **global cluster refinement**: refining some samples of local clustering via sufficient statistics from each subset; d. **global clustering estimation**: from refined local cluster assignments, identifying a global clustering estimate that minimizes the expected posterior loss defined on the space of partitions; and e. **sampling model parameters**: conditional on the optimal global cluster and subcomponent estimate, drawing model parameters from MCMC and performing inference. See Figure 2.1 for a diagram of the procedures.

Global cluster refinement (step c), global clustering estimation (step d) and sampling model parameters (step e) incur a small amount of data transmission between the master and workers. In the rest of this section, we introduce the notation used throughout the paper and elaborate on steps b, c, d and e.

### 2.3.1 Notation

We represent any quantity associated with subset or worker $r$ by adding a subscript $r$, $r = 1, \ldots, R$. Let the sample size of data subset $Y_r$ be $n_r$, and $Y_r = \{y_{r1}, y_{r2}, \ldots, y_{rn_r}\}$. Assume subset $Y_r$ is distributed to and processed on worker $r$, $r = 1, \ldots, R$. As discussed in Section 2.2, the finite mixture of mixtures model induces both latent cluster and subcomponent allocations of the data. Let a vector of latent cluster allocations be $c = (c_1, \ldots, c_N)$, where $c_i = j$ indicates that data point $i$ belongs to cluster $j$. Similarly let a vector of latent subcomponent allocations be $s$, where $s_i = j$ indicates that data point $i$ belongs to subcomponent $j$.

In a distributed computing paradigm, we can also express $c$ (resp. $s$) as $\{c_1, \ldots, c_R\}$ (resp. $\{s_1, \ldots, s_R\}$), where $c_r$ (resp. $s_r$) represents a vector of latent cluster (resp. subcomponent) allocations for $Y_r$. Both $s$ and $c$ are updated during global cluster refinement. Let the refined cluster allocations be $\tilde{c} := \{\tilde{c}_1, \ldots, \tilde{c}_R\}$ (resp. $\tilde{s} := \{\tilde{s}_1, \ldots, \tilde{s}_R\}$), where $\tilde{c}_r$ (resp. $\tilde{s}_r$) represents the refined cluster allocations based on $c_r$ (resp. $s_r$).
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y = {y_1, \ldots, y_N} )</td>
<td>Sample of size ( N )</td>
</tr>
<tr>
<td>( K )</td>
<td>A pre-determined upper bound of number of clusters</td>
</tr>
<tr>
<td>( L )</td>
<td>Number of subcomponents per cluster</td>
</tr>
<tr>
<td>( r, r = 1, \ldots, R )</td>
<td>Subset or worker index</td>
</tr>
<tr>
<td>( Y = {Y_1, \ldots, Y_R} )</td>
<td>The sample partitioned into ( R ) non-overlapping subsets</td>
</tr>
<tr>
<td>( Y_r = {y_{r1}, y_{r2}, \ldots, y_{rn_r}} )</td>
<td>Data subset ( r ) of size ( n_r ), with ( y_{ri} ) being data point ( i )</td>
</tr>
<tr>
<td>( c = {c_1, \ldots, c_R} )</td>
<td>Latent cluster allocations</td>
</tr>
<tr>
<td>( s = {s_1, \ldots, s_R} )</td>
<td>Latent subcomponent allocations</td>
</tr>
<tr>
<td>( c_r = {c_{r1}, \ldots, c_{rn_r}} )</td>
<td>Latent cluster allocations for subset ( r ), where ( c_{ri} = j ) indicates data point ( i ) belongs to cluster ( j )</td>
</tr>
<tr>
<td>( s_r = (s_{r1}, \ldots, s_{rn_r}) )</td>
<td>Latent subcomponent allocations for subset ( r ), where ( s_{ri} = j ) indicates data point ( i ) belongs to subcomponent ( j )</td>
</tr>
<tr>
<td>( \tilde{c} = {\tilde{c}_1, \ldots, \tilde{c}_R} )</td>
<td>Updated cluster allocations</td>
</tr>
<tr>
<td>( \tilde{s} = (\tilde{s}_1, \ldots, \tilde{s}_R) )</td>
<td>Updated subcomponent allocations</td>
</tr>
<tr>
<td>Item ((r, j))</td>
<td>First item indexing set; ( r ) is the worker/subset index and ( j ) is the item index within the worker</td>
</tr>
<tr>
<td>Item ((r, L(k-1) + l))</td>
<td>Item index for the ( l )th subcomponent in cluster ( k ) in subset ( r )</td>
</tr>
<tr>
<td>( \mathcal{L}_r )</td>
<td>Item counts in subset ( r )</td>
</tr>
<tr>
<td>( B = \sum_{r=1}^R \mathcal{L}_r )</td>
<td>Total number of items across subsets</td>
</tr>
<tr>
<td>item ( b )</td>
<td>Second item indexing set</td>
</tr>
<tr>
<td>( b = 1, \ldots, B )</td>
<td>created by an ordering induced by the workers</td>
</tr>
<tr>
<td>( z_j \in {1, \ldots, H} ) and ( \tilde{z}_j )</td>
<td>Group allocation and updated label of item ( j ) respectively</td>
</tr>
<tr>
<td>( r_j )</td>
<td>The worker where item ( j ) resides</td>
</tr>
<tr>
<td>( r^* )</td>
<td>The reference subset</td>
</tr>
<tr>
<td>( \mathcal{J}_j )</td>
<td>The set of data indices in item ( j )</td>
</tr>
<tr>
<td>( y_{j1}, n_{j} ) and ( \bar{y}_j )</td>
<td>Data, sample size, first and second moment associated with item ( j ) respectively</td>
</tr>
<tr>
<td>( Y_h, N_{h}, \bar{Y}_h ) and ( \delta_h )</td>
<td>Data, sample size, first and second moment associated with group ( h ) respectively</td>
</tr>
<tr>
<td>Subscript ( h \backslash b )</td>
<td>Group ( h ) without taking item ( b ) into account</td>
</tr>
</tbody>
</table>

(Global Clustering Estimation)

| \( C_i \) | The set of data indices in cluster \( i \) of the true clustering |
| \( \tilde{C}_j \) | The set of data indices in cluster \( j \) of clustering candidate \( \tilde{c} \) |
| \( N_{ci} \) | \( |C_i| \) under the true clustering |
| \( N_{cj} \) | \( |\tilde{C}_j| \) under clustering candidate \( \tilde{c} \) |
| \( N_{ij} = |C_i \cap \tilde{C}_j| \) | The number of data points in both \( C_i \) and \( \tilde{C}_j \) |
| \( \mathcal{T} \) | The set of iterations refined in global cluster refinement |
| \( \mathcal{R} \subset \mathcal{T} \) | The set of iterations associated with the clustering candidates |

Superscript \((t)\) or \((t, t')\) | Iteration \( t \) or iteration \( t \) and \( t' \) |

\textbf{Table 2.1}: Selected notation and descriptions
During global cluster refinement stage, we align and refine local cluster assignments at the level of subcomponents. We henceforth refer to *non-empty subcomponents as items*, using the word “item” to distinguish these objects from single observations. The cluster assignment of an item is termed *group assignment*, using the word “group” to distinguish it from the cluster assignment of a single observation. Additional notation is introduced in the relevant sections. See Table 2.1 for a collection of important notation and descriptions used in this article.

### 2.3.2 Sampling Local Clustering

The overall posterior density given \( \mathcal{Y} \) can be represented by

\[
p(\Theta, \eta \mid \mathcal{Y}) = \frac{\{\prod_{r=1}^{R} \prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\} p(\Theta, \eta)}{\int \int \{\prod_{r=1}^{R} \prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\} p(\Theta, \eta) d\Theta d\eta}.
\]

We define the \( r \)th subset posterior density given \( \mathcal{Y}_r, r = 1, \ldots, R \) by

\[
p_r(\Theta, \eta \mid \mathcal{Y}_r) = \frac{\{\prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\} p(\Theta, \eta)}{\int \int \{\prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\} p(\Theta, \eta) d\Theta d\eta}, \tag{2.8}
\]

respectively. In our algorithm, we run MCMC on \( R \) workers in parallel based on (2.8), producing draws from each subset posterior \( p_r(\Theta, \eta \mid \mathcal{Y}_r), r = 1, \ldots, R \).

The subset posterior is reminiscent of that in (Li et al., 2017), which is defined as

\[
\frac{\{\prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\}^R p(\Theta, \eta)}{\int \int \{\prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta)\}^R p(\Theta, \eta) d\Theta d\eta}
\]

when \( n_1 = \cdots = n_R \). Their goal was to estimate posterior quantile functions (such as the posterior interval) and by raising the subset likelihood to the power of number of workers, the subset posteriors rescale their variance to be roughly of the same order as the variance of the overall posterior \( p(\Theta, \eta \mid \mathcal{Y}) \) as a stochastic approximation of the overall posterior (Li et al., 2017). However, we are interested in identifying clusters. Raising the subset likelihood to such a power amounts to replicating noise and frequently results in identification of noise clusters.

Our definition of the subset posterior is roughly equivalent to raising the overall likelihood to a fractional power. This strategy, also called tempering, has been commonly used for improving the mixing of Markov chains when sampling from a multi-modal posterior distribution CITE. It was
also proven to be an approximation to the “coarserend” posterior, which is obtained by conditioning on a neighbourhood of the empirical distribution instead of the data exactly so as to improve robustness to perturbations from the model (Miller and Dunson, 2018).

To produce draws from \( p_r(\Theta, \eta \mid Y_r), r = 1, \ldots, R \) (or \( p(\Theta, \eta \mid Y) \) in a serial algorithm), we can run a block conditional Gibbs sampler with data augmentation. The sampler alternates between imputing cluster allocations and updating parameters specific to each cluster, with the latter step alternating between imputing subcomponent allocations and updating subcomponent-specific parameters. We provide a detailed sampling scheme in Appendix 1. One can, however, adopt any other sampling scheme to improve mixing (such as a collapsed Gibbs sampler) as long as samples of cluster and subcomponent allocations are produced.

### 2.3.3 Global Cluster Refinement

**Motivation and General ideas**

The clustering \( c = \{c_1, \ldots, c_R\} \), generated from naively combining the clustering allocations from the subsets, does not, in general, mimic a sample of cluster allocations from \( p(\Theta, \eta \mid Y) \) for two reasons. First, the cluster labeling can vary across workers. Figure 2.2 shows a sample of local cluster allocations from three different workers when the data set is partitioned and distributed to 4 workers: cluster 9 in subset 1 corresponds to cluster 10 in subset 3. Second, the clustering structure could vary across subsets. For example, a single cluster in one subset (e.g. cluster 1 in subset 1 in Figure 2.2) can correspond to several smaller ones (e.g. cluster 1 and 3 in subset 2) in another subset. Even worse, a cluster in one subset can correspond to a significant number of, but not all, data points from multiple clusters in another subset. Therefore, an algorithm to adjust samples of local cluster assignments, particularly enabling merging and splitting clusters for handling the second issue, is essential.

Inspired by Ni et al. (2020), we propose a simple and communication efficient algorithm that permits both cluster merging and splitting. Their algorithm SIGN uses a multi-step recursive approach to cluster merging, in which the local clusters are frozen recursively, meaning that the ob-
Figure 2.2: A sample of local cluster (top) and subcomponent (bottom) allocations before refinement for subset 1, 2 and 3 when the data set is partitioned and distributed to 4 workers. 6% randomly selected data points are plotted.

Figure 2.3: Representation of items via the two indexing systems. Blue and white represent non-empty and empty subcomponents respectively.
servations within each cluster will never be split but possibly merged in the subsequent steps.

In our algorithm, in contrast, observations are frozen at the subcomponent level, rather than at the cluster level. One big advantage is that heavily overlapping subcomponents provide a natural solution: we can merge or group the frozen local subcomponents across subsets based on their degrees of overlap and map the updated subcomponent labels in an appropriate way to the cluster level, which results in automatic merges or splits of the clusters. Such a joint grouping scheme also ensures that a unified set of subcomponent labels is applied across subsets. Another advantage, since each subcomponent can be described by a Gaussian distribution, is that the natural model to enable such joint grouping is simply a finite mixture of Gaussians: if the distributions of any two subcomponents can be well approximated by a single Gaussian kernel, the subcomponents are likely to be grouped together. In Figure 2.2, for example, subcomponent 1, 7 and 16 from subset 1, 2 and 3 (in the bottom left triangular cluster), respectively, are likely to be assigned to the same group due to their high overlap.

**Notation**

This process depends on the data only through some summary statistics of the non-empty subcomponents, which we refer to as items, using the word “item” to distinguish these objects from single observations.

For ease of demonstration, we herein introduce two different indexing systems for these items. The first is represented by \((r, j)\), where \(r\) is the worker/subset index and \(j\) is the item index within the worker. Since each cluster contains \(L\) subcomponents, we can index the \(l\)th subcomponent in cluster \(k\) of subset \(r\) by \(L(k-1) + l\); if this subcomponent is non-empty, its item index is given by \(L(k-1) + l\).

The second indexing method is represented by the item index created by an ordering induced by the ordering of the workers. Let the item counts in subset \(r\) be \(\mathcal{L}_{r}\) and the total number of items across all subsets be \(B\); we have \(B = \sum_{r=1}^{R} \mathcal{L}_{r}\). Hence, the set of items can be naturally indexed by \(\{1, 2, \ldots, B\}\). Using this notation, for example, the worker where item \(b\) resides can be expressed as \(r_{b} = \{r : y_{b} \subset \mathcal{Y}_{r}, r = 1, \ldots, R\}\). See Figure 2.3 for an intuitive illustration of these two
indexing systems.

We term the cluster assignment of an item *group assignment*, using the word “group” to distinguish it from the cluster assignment of a single observation. These groups are defined by items from a randomly chosen reference subset, denoted by \( r^* \). Denote the associated quantities of an item, including its original group allocation, group allocation defined by \( r^* \), data, set of data indices, sample size, first and second moment and the worker to which it belongs by \( z \cdot, \tilde{z} \cdot, \mathcal{I} \cdot, n \cdot, \bar{y} \cdot, \delta \cdot \), respectively, where \( \cdot \) represents the index of the underlying item. Denote the associated quantities of group \( h \), including the data, sample size, first and second moment, by \( Y_h, N_h, \bar{Y}_h \) and \( S_h \) respectively. Following the second indexing rule, we have \( Y_h = \{ y_b : z_b = h, b = 1, \ldots, B \} \).

Subscript \( h \setminus b \) represents group \( h \) without taking item \( b \) into account. Let \( Q_{b,h} \) be the collection of three summary statistics, with \( Q_{b,h} = (N_{h \setminus b}, \bar{Y}_{h \setminus b}, S_{h \setminus b}) \).

**Methodology**

We use a finite mixture of Gaussian to align the items from across the subsets by assigning each of them to a mixture component. The mixture model assumes that the density of \( y_b \) is given as follows:

\[
f(y_b | \Xi, \tau) = \sum_{h=1}^{H} \tau_h N_h(y_b | \xi_h),
\]

where \( H \) is the number of components, each of which represents a *group*, \( \Xi = \{ \xi_1, \ldots, \xi_H \} \), \( \xi_h = \{ m_h, \mathcal{C}_h \} \) and \( \sum_{h=1}^{H} \tau_h = 1 \). In particular,

\[
N_h(y_b | \xi_h) = \prod_{i \in \mathcal{I}_b} N(y_i | \xi_h).
\]

Placing commonly used conjugate priors on \( \Xi \) and \( \tau \)

\[
(\tau_1, \ldots, \tau_H) \sim \text{Dir}(\alpha_0, \ldots, \alpha_0),
\]

\[
m_h | \mathcal{C}_h \sim N(0, \mathcal{C}_h) \quad \text{and} \quad \mathcal{C}_h \sim \text{IW}(\nu_0, S_0) \quad h = 1, \ldots, H,
\]

yields a Bayesian model. The refinement procedure is not sensitive to the specific choice of prior parameters because each item contains many observations. In addition, as we shall demonstrate below, this model, functioning at the level of items, allows for both cluster merging and splitting.
Computation

The inference based on (2.9) involves two key ideas. They are the use of reference subsets and a collapsed Gibbs sampler.

Reference subsets

First, the use of a reference subset is a simple idea to ensure that the subcomponents across all subsets are properly aligned and the resulting updated clusters are reasonably contiguous. Specifically, we first choose a reference subset \( r^* \) at random. By setting \( H \) in (2.9) to be the number of items in the reference subset \( r^* \) and having each item in \( r^* \) representing a mixture component, we are effectively aligning each item from all subsets to an item in \( r^* \). In addition, since the mapping of \( \frac{\tilde{\pi}(r^*,M)}{L} \) yields reasonably contiguous clusters in the reference subset, the mapping defined by \( \frac{\tilde{\pi}}{L}, b = 1, \ldots, B \) also creates reasonably contiguous clusters, and we thus set the refined cluster labels associated with each item to be \( \frac{\tilde{\pi}}{L}, b = 1, \ldots, B \).

We demonstrate our idea through an example below. Assume three subcomponents (i.e. \( L = 3 \)) are used to approximate a cluster and the item index set in the reference subset \( r^* \) is \( \{(r^*, 1), (r^*, 2), (r^*, 5), (r^*, 9)\} \) based on the first indexing rule. A possible mapping between the items and the mixture components is given in Table 2.2. If item \( b \) from a different subset is assigned to mixture component or group 3 in (2.9) (i.e. \( z_b = 3 \)), it is equivalently aligned with item \( (r^*, 5) \); then its group allocation defined by \( r^* \) is 5, i.e. \( \tilde{z}_b = 5 \). Consequently its associated cluster label is updated to 2 (\( \frac{5}{3} \)). If item \( (r^*, 5) \) is assigned to group 2 (i.e. \( z_{(r^*, 5)} = 2 \)), then it is aligned with \( (r^*, 2) \) and its associated cluster label is updated to 1, which amounts to the merge of cluster 1 and 2 in the reference subset.

This example shows that our refinement procedure allows items originating from different clusters to be assigned to the same or adjacent groups and then become subcomponents of the same cluster. It also allows items originating from the same cluster to be assigned to different groups and then become subcomponents of different clusters. This means that our global cluster refinement procedure is flexible enough to accommodate both cluster merging and splitting.

While the choice of a reference subset involves uncertainty, we refine many iterations of local clustering samples and for each refined iteration we draw a reference subset at random. Therefore,
Item index  |  $(r^*, 1)$  |  $(r^*, 2)$  |  $(r^*, 5)$  |  $(r^*, 9)$  
--- | --- | --- | --- | ---  
Mixture component in (2.9)  |  1  |  2  |  3  |  4  
Refined cluster label  |  1  |  1  |  2  |  3  

Table 2.2: An example dictionary of item indices, mixture components and cluster labels

the final clustering estimate, obtained from these refined iterations, is not sensitive to the choice of reference.

**Collapsed Gibbs sampler**

Our model specification, (2.9), (2.10) and (2.11), enables derivation of a collapsed Gibbs sampler, where we integrate out the model parameters $\{\Xi, \tau\}$ from the joint posterior and only update the latent group allocation through MCMC sampling; such a sampler in general accelerates convergence to the posterior distribution. In addition, the sampler, as we shall demonstrate, can be implemented in parallel and depends on the data only through three summary statistics of each item.

The sampling step for item $b$ depends on:

$$
P(z_b = h \mid z_{\setminus b}, Y) \propto P(z_b = h \mid z_{\setminus b})p(y_b \mid Y_{h\setminus b}), \quad h = 1, \ldots, H; b = 1, \ldots, B, \quad (2.12)
$$

where the first term is

$$
P(z_b = h \mid z_{\setminus b}) = \frac{\Gamma(N + H\alpha_0 - n_b)\Gamma(N_{h\setminus b} + n_b + \alpha_0)}{\Gamma(N + H\alpha_0)\Gamma(N_{h\setminus b} + \alpha_0)}, \quad b = 1, \ldots, B, \quad (2.13)
$$

and the second term is the joint marginal density of observations in item $b$, which is given by a product of $t$-densities:

$$
p(y_b \mid Y_{h\setminus b}) = \prod_{j \in I_b} t_d \left( y_j \mid m_{h\setminus b}, \frac{\kappa_{h\setminus b} + 1}{\kappa_{h\setminus b}(\nu_{h\setminus b} - d + 1)}S_{h\setminus b}, \nu_{h\setminus b} - d + 1 \right), \quad b = 1, \ldots, B, \quad (2.14)
$$

where

$$
\begin{align*}
\kappa_{h\setminus b} &= 1 + N_{h\setminus b}, \\
\nu_{h\setminus b} &= \nu_0 + N_{h\setminus b}, \\
m_{h\setminus b} &= \frac{N_{h\setminus b}Y_{h\setminus b}}{\kappa_{h\setminus b}} \quad \text{and} \quad S_{h\setminus b} = S_0 + N_{h\setminus b}S_{h\setminus b} - \kappa_{h\setminus b}m_{h\setminus b}m_{h\setminus b}^T.
\end{align*}
$$
The details of the derivation are included in Appendix 2.

**Details of the parallel computing procedure**

Computation of (2.13) and (2.14) requires statistics $N_{h|b}$, $\bar{Y}_{h|b}$, $S_{h|b}$, $h = 1, \ldots, H$, which are simple functions of the sample size, mean, and second moment associated with the items:

$$N_{h|b} = \left( \sum_{i \in b, \{j: z_{ij} = h\}} n_i \right) - n_b,$$  \hspace{1cm} (2.15)

$$\bar{Y}_{h|b} = \left( \sum_{i \in b, \{j: z_{ij} = h\}} n_i \bar{y}_i \right) - n_b \bar{y}_b \bigg/ N_{h|b},$$ \hspace{1cm} (2.16)

$$S_{h|b} = \left( \sum_{i \in b, \{j: z_{ij} = h\}} n_i \delta_i \right) - n_b \delta_b \bigg/ N_{h|b},$$ \hspace{1cm} (2.17)

Therefore, it suffices to have the workers communicate these summary statistics, namely sample size $n_b$, mean $\bar{y}_b$ and second moment $S_b$ ($b = 1, \ldots, B$) to the master. Then the master evaluates $\{Q_{b,h} : \forall b \text{ and } h\}$ and their function (2.13). $\{Q_{b,h} : \forall b \text{ and } h\}$ are then communicated to the appropriate worker $r_b = \{r : y_b \subset Y_r, r = 1, \ldots, R\}$ to evaluate (2.14) for all $b$, which can be completed in an embarrassingly parallel manner. The resulting values will again be communicated to the master for evaluation of (2.12) and updating latent group allocations $z$, based on which the associated cluster labels of subset $r$ are updated to be $\tilde{c}_r := \{\tilde{c}_{r1}, \ldots, \tilde{c}_{rn_r}\}$, with

$$\tilde{c}_{ri} = \left[ \frac{\tilde{z}_{(r,j)}}{L} \right], \text{ for } i \in J_{(r,j)} \hspace{1cm} (2.18)$$

where $i = 1, \ldots, n_r$ and $r = 1, \ldots, R$. The resulting $\tilde{c}$ is considered an approximate sample from $p(c \mid Y)$. See Algorithm 1 for a clear outline and the flow chart below for an intuitive illustration of refinement of one clustering sample. We find refining 100 local clustering samples $c = \{c_1, \ldots, c_R\}$ by running one iteration of the collapsed Gibbs sampler sufficient for excellent performance.

1. A sample of subcomponent allocations
2. Random choice of the reference: $r^* = 3$

![Diagram of subcomponent allocations in the reference subset]

**Figure 2.4:** The sample of sub-component allocations in the reference subset

<table>
<thead>
<tr>
<th>Item index</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,18)</td>
<td>1</td>
</tr>
<tr>
<td>(3,17)</td>
<td>2</td>
</tr>
<tr>
<td>(3,16)</td>
<td>3</td>
</tr>
<tr>
<td>(3,3)</td>
<td>4</td>
</tr>
<tr>
<td>(3,2)</td>
<td>5</td>
</tr>
<tr>
<td>(3,1)</td>
<td>6</td>
</tr>
</tbody>
</table>

**Table 2.3:** A bijective map between the item indices and groups

3. Initialization to the closest item in the reference
4. Results from running one iteration of collapsed Gibbs sampler

5. Item labels updated based on Table 2.3
6. Cluster labels updated based on (2.18)

Algorithm 1 Global Cluster Refinement (one iteration)

**Input:** Summary statistics \( \{(n_b, \bar{y}_b, S_b) : b = 1, \ldots, B\} \) sent from Workers to Master

**Output:** Refined cluster labels \( \tilde{c} = \{\tilde{c}_1, \ldots, \tilde{c}_R\} \)

1: \( \triangleright \) Define the mixture components in (2.9) using a reference subset

2: **On Master:**

3: \( r^* \sim \text{Unif}([1, \ldots, R]) \quad \triangleright \) Set a subset as reference

4: \( H \leftarrow \mathcal{L}_{r^*} \quad \triangleright \) Set number of mixture components to be number of items in \( r^* \)

5: Determine a bijective map \( g : \text{items in subset } r^* \rightarrow \{1, \ldots, H\} \)

6: \( \triangleright \) Initialize

7: **for** \( b = 1 \) to \( B \) **do**

8: \( z_b \leftarrow g (\arg \min_j \| \bar{y}_b - \bar{y}_{r^*} \|) \quad \triangleright \) Initialize \( z_b \) to the closest item in the reference

9: **end for**

10: \( \triangleright \) Run one iteration of the collapsed Gibbs sampler

11: **for** \( b = 1 \) to \( B \) **do**

12: \( \quad \text{for } h = 1 \) to \( H \) **do**

13: Compute \( \mathcal{Q}_{b,h} \) using (2.15), (2.16) and (2.17)

14: Send \( \mathcal{Q}_{b,h} \) to Worker \( r_b = \{r : y_b \subset \mathcal{Y}_r, r = 1, \ldots, R\} \)

15: Compute \( P(z_b = h \mid z_{-b}) \) using (2.13)
On Workers:
\textbf{parfor} Worker $r = 1$ to $R$ do
\hspace{1em} for all item $(r, j)$ do
\hspace{2em} for $h = 1$ to $H$ do
\hspace{3em} Compute $p(y_{(r,j)} | \mathcal{Y}_{h\setminus(r,j)})$ using $Q_{(r,j),h}$ and (2.14)
\hspace{3em} Send $p(y_{(r,j)} | \mathcal{Y}_{h\setminus(r,j)})$ to Master
\hspace{2em} end for
\hspace{1em} end for
\end{parfor}

On Master:
\textbf{for} $b = 1$ to $B$ do
\hspace{1em} for $h = 1$ to $H$ do
\hspace{2em} $P(z_b = h | z_{\setminus b}, \mathcal{Y}) \leftarrow p(y_b | \mathcal{Y}_{h\setminus b})P(z_b = h | z_{\setminus b})$
\hspace{1em} end for
\hspace{1em} Draw $z_b$ from a categorical distribution with probability vector
\hspace{2em} $p \propto [P(z_b = 1 | z_{\setminus b}, \mathcal{Y}), \ldots, P(z_b = H | z_{\setminus b}, \mathcal{Y})]$ \hspace{1em} \noindent \text{end for}
\textbf{Map the group allocation of items to determine new cluster labels}

On Workers:
\textbf{parfor} Worker $r = 1$ to $R$ do
\hspace{1em} for all item $(r, j)$ do
\hspace{2em} $\tilde{c}_{ri} \leftarrow \left[ \frac{\tilde{z}_{(r,j)}}{L} \right]$, for $i \in \mathcal{J}_{(r,j)}$
\hspace{2em} end for
\hspace{1em} return $\tilde{c}_r$
2.3.4 Global Clustering Estimation

After refining iterations of local clustering samples, it is natural to wonder how to find an optimal clustering estimate. Decision theory provides an elegant solution to this. Specifically, the optimal clustering estimate is one that minimizes the posterior expectation of a loss function defined on the space of partitions:

$$c^* = \arg \min_{\hat{c}} E[L(c, \hat{c}) \mid \mathcal{Y}].$$

$E[L(c, \hat{c}) \mid \mathcal{Y}]$ is often simplified to be a function that depends on the posterior only through the posterior similarity matrix, which is defined to be an $N$ by $N$ matrix with entry $(i, j)$ being the posterior probability that data points $i$ and $j$ are assigned to the same cluster $P(c_i = c_j \mid \mathcal{Y})$. Since $P(c_i = c_j \mid \mathcal{Y})$ can be readily estimated by the proportion of posterior samples that cluster data points $i$ and $j$ together, practitioners often estimate the posterior similarity matrix to obtain $c^*$.

This, however, becomes computationally infeasible for big data of size $N$: the $N$ by $N$ posterior similarity matrix incurs both large storage and computational costs, with the latter being $O(N^2)$ for each iteration involved in creating the posterior similarity matrix.

We propose a simple solution that exploits an inherent property of any loss function defined on the space of partitions and as a result is general enough to accommodate any eligible loss. Since clustering is invariant to the permutation of data point indices, any loss $L(c, \hat{c})$ must be a function of the joint counts $N_{ij} = |C_i \cap \hat{C}_j|$, which is the number of data points in both $C_i$, the set of data indices in cluster $i$ under $c$, and $\hat{C}_j$, the set of data indices in cluster $j$ under $\hat{c}$, $i = 1, \ldots, \text{number of clusters in } c$ and $j = 1, \ldots, \text{number of clusters in } \hat{c}$ (Binder, 1978). These joint counts can be easily obtained through parallel computation, because the refined cluster allocations $\{\hat{c}^{(t)}_i, t \in \mathcal{T}\}$ make the following relationships hold:

$$N_{ij}^{(t)} = \sum_{r=1}^{R} N_{r,ij}^{(t)} \quad N_{+j} = \sum_{r=1}^{R} N_{r,j}$$

where $N_{ij}^{(t)}$ and $N_{+j}$ are computed on the master using the statistics $N_{r,ij}^{(t)}$ and $N_{r,j}$ communicated from the workers. We demonstrate this idea and provide a detailed outline of the computing...
procedures in Algorithm 2 through an example loss, variation information (VI) loss; computing procedures based on other losses can be similarly derived.

The VI loss is an information theoretic criterion for comparing two clustering structures and is defined as:

$$VI(c, \hat{c}) = H(c) + H(\hat{c}) - 2I(c, \hat{c})$$

where $H$ is the entropy function and $I$ is the mutual information (Meilă, 2007), $k_N$ and $\hat{k}_N$ are the number of clusters in $c$ and $\hat{c}$ respectively, and $N_{i+}$ and $N_{+j}$ represent the counts in $C_i$ under $c$ and $C_j$ under $\hat{c}$ respectively. The posterior expected VI loss can be simplified to:

$$E[VI(c, \hat{c}) | \mathcal{Y}] = \sum_{n=1}^{N} \log \left( \sum_{n=1}^{N} \mathbb{1}(\hat{c}_n = \hat{c}_n) \right) - 2 \sum_{n=1}^{N} E \left[ \log(\sum_{n'=1}^{N} \mathbb{1}(c_{n'} = c_n, \hat{c}_{n'} = \hat{c}_n)) | \mathcal{Y} \right]$$

(2.20)

up to a constant. To improve the computational efficiency, Wade and Ghahramani (2018) developed a lower bound of (2.20) and reduced the computation cost to $O(N^2)$ for a given clustering candidate $\hat{c}$, which still remains astronomical for our case. Our approach, which depends on the joint counts, can estimate (2.20) directly without relying on the lower bound. Let $\mathcal{T} \in \{1, \ldots, T\}$ be the set of iterations adjusted in the global cluster refinement step. Consider a random subset $\mathcal{M} \subset \mathcal{T}$ and clustering candidates $\hat{c} \in \{\hat{c}^{(t)}, t \in \mathcal{M}\}$. For clustering candidate $\hat{c}$, (2.20) can be estimated by

$$E[VI(c, \hat{c}) | \mathcal{Y}] = \sum_{j=1}^{k_N} \frac{N_{+j}}{N} \log \left( \frac{N_{+j}}{N} \right) - \frac{2}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_{i=1}^{k_N} \sum_{j=1}^{\hat{k}_N} \frac{N_{ij}^{(t)}}{N} \log \left( \frac{N_{ij}^{(t)}}{N} \right)$$

(2.21)

where superscript $(t)$ represents iteration $t$. Here both $N_{ij}^{(t)}$ and $N_{+j}$ can be computed in parallel.

Then the optimal point estimate of clustering is given by:

$$c^* = \arg \min_{\hat{c}^{(t)}, \hat{c} \in \mathcal{M}} E[VI(c, \hat{c}^{(t)}) | \mathcal{Y}]$$

For each clustering candidate, the evaluation of $\{N_{ij}^{(t)} : \forall t \in \mathcal{T}, i = 1, \ldots, k_N^{(t)}, j = 1, \ldots, \hat{k}_N\}$ and $\{N_{+j}, j = 1, \ldots, \hat{k}_N\}$ is $O(n|\mathcal{T}|)$, where $n := \max_r n_r$ is the largest sample size on any
worker. The cost of communicating these statistics to the master depends on the total number of variables being transferred, which is \(\hat{k}_N R \left( \sum_{t \in \mathcal{T}} k_N^{(t)} + 1 \right)\) for every clustering candidate. Although \(\hat{k}_N R \sum_{t \in \mathcal{T}} k_N^{(t)}\) is approximately linear in the number of iterations, \(|\mathcal{T}|\) can be chosen to be much smaller than the total number of iterations for sampling local clustering. Our experiments show superior clustering performance with \(|\mathcal{T}| = 100\) and \(|\mathcal{M}| = 20\).

**Algorithm 2** Global Clustering Estimation (for VI loss as an example)

**Input:** Output of refined samples of local cluster allocations from Algorithm 1

**Output:** Optimal clustering estimate \(c^*\)

1:▷ Calculate necessary statistics to estimate the posterior expectation of the VI loss

2: **On Workers:**

3: **parfor** Worker \(r = 1\) to \(R\) do

4:▷ Iterate through the set of clustering candidates

5: for iteration \(t' \in \mathcal{M}\) do

6:for cluster \(j = 1\) to \(k_N^{(t')}\) do▷ \(k_N^{(t')}\): number of clusters in candidate \(t'\)

7: Compute \(N_r^{(t')}\)

8: Send \(N_r^{(t')}\) to Master

9:▷ Iterate through the set of refined local cluster allocations samples

10: for iteration \(t \in \mathcal{T}\) do

11:for cluster \(i = 1, \ldots, k_N^{(t)}\) do

12: Compute \(N_r^{(t,t')}\)

13: Send \(N_r^{(t,t')}\) to master

14: end for

15: end for

16: end for

17: end for

18: **end parfor**

19:▷ Estimate the posterior expectation of the VI loss

20: **On Master:**
for iteration \( t' \in \mathcal{M} \) do

\[ \text{Term}_{1}(t') \leftarrow 0 \quad \triangleright \text{Term}_{1}(t') : \text{the first term in (2.21)} \]

\[ \text{Term}_{2}(t') \leftarrow 0 \quad \triangleright \text{Term}_{2}(t') : \text{the second term in (2.21)} \]

for cluster \( j = 1 \) to \( k_{N}^{(t')} \) do

\[ N_{+,j}^{(t')} \leftarrow \sum_{r=1}^{R} N_{r,+j}^{(t')} \]

\[ \text{Term}_{1}(t') \leftarrow \text{Term}_{1}(t') + \left( \frac{N_{+,j}^{(t')}}{N} \right) \log \left( \frac{N_{+,j}^{(t')}}{N} \right) \]

end for

for iteration \( t \in \mathcal{T} \) do

\[ N_{ij}^{(t,t')} \leftarrow \sum_{r=1}^{R} N_{r,ij}^{(t,t')} \]

\[ \text{Term}_{2}(t') \leftarrow \text{Term}_{2}(t') + \left( \frac{N_{ij}^{(t,t')}}{N} \right) \log \left( \frac{N_{ij}^{(t,t')}}{N} \right) \]

end for

end for

\[ \mathbb{E} \left[ \text{VI}(\hat{c}, \overline{e}(t') \mid \mathcal{Y}) \right] \leftarrow \text{Term}_{1}(t') - \text{Term}_{2}(t') \]

end for

\[ \triangleright \text{Identify the clustering that minimizes the expected VI loss} \]

\[ c^{\ast} \leftarrow \arg \min_{\hat{c}} \mathbb{E} \left[ \text{VI}(\hat{c}, \overline{e}(t') \mid \mathcal{Y}) \right] \]

\[ \hat{e}(t'), t' \in \mathcal{M} \]

\[ \text{return } c^{\ast} \]

### 2.3.5 Sampling Model Parameters

This section only applies if one is interested in quickly classifying future subjects, density estimation or generating from posterior predictive for inference. These goals make drawing (approximate) posterior samples of model parameters \((\Theta, \eta)\) necessary. For example, to classify a new subject, the Bayes classifier—evaluating the posterior probability of belonging to each cluster and assigning to the cluster that yields the highest posterior probability—is commonly used and depends on the model parameters.

The general idea of our algorithm is to sample the model parameters conditional on the cluster and subcomponent assignments associated with the optimal clustering estimate found in the last
step (i.e. global clustering estimation), allowing the MCMC updates to depend on the data only through summary statistics. These statistics include subcomponent sizes \(|i : \tilde{s}_i = l, \tilde{c}_i = k|\), the sum of squares \(\sum_{i: \tilde{s}_i = l, \tilde{c}_i = k} y_i y_i^T\) and the data sums \(\sum_{i: \tilde{s}_i = l, \tilde{c}_i = k} y_i, i = 1, \ldots, N\). Refer to Appendix .1 for the details of MCMC updates.

Such an approach has several advantages. First, subcomponent assignments are aligned across all subsets through global cluster refinement, which makes the following relationships hold:

\[
\sum_{i: y_i \in Y} 1_{\tilde{s}_i = l, \tilde{c}_i = k} = \sum_{r=1}^R \sum_{i: y_i \in Y_r} 1_{\tilde{s}_i = l, \tilde{c}_i = k},
\]

\[
\sum_{y_i \in Y, \tilde{s}_i = l, \tilde{c}_i = k} y_i y_i^T = \sum_{r=1}^R \sum_{y_i \in Y_r, \tilde{s}_i = l, \tilde{c}_i = k} y_i y_i^T,
\]

\[
\sum_{y_i \in Y, \tilde{s}_i = l, \tilde{c}_i = k} y_i = \sum_{r=1}^R \sum_{y_i \in Y_r, \tilde{s}_i = l, \tilde{c}_i = k} y_i, l = 1, \ldots, L, k = 1, \ldots, k_N^*, \]

where \(k_N^*\) is the number of clusters in \(c^*\).

These relationships justify our parallel algorithm: the workers first compute the relevant statistics in parallel and then communicate them to the master node for summing. Such communication only occurs once; because conditional on the subcomponent assignments, the summary statistics are fixed throughout the MCMC updates. Second, the fixed subcomponent assignments and the use of summary statistics significantly reduce computation per iteration and lead to faster convergence due to reduced dependence between iterations. Third, posterior estimates of parameters are often non-identifiable in finite mixture models due to label switching, a phenomenon which occurs when exchangeable priors are placed on the parameters. It is particularly challenging to resolve, despite many available post-sampling relabeling schemes (Stephens, 2000; Sperrin et al., 2010; Papastamoulis and Iliopoulos, 2010; Papastamoulis, 2014; Rodríguez and Walker, 2014), for overfitted mixture models, as superfluous clusters may merge or overlap with other ones, or be empty. By fixing the latent cluster assignments, we bypass label switching at the cluster level, eliminating the need to post process MCMC outputs. See Appendix .3 for an outline of the algorithm.
2.4 Experiments

In this section, we provide extensive numerical experiments to assess the performance of DIB-C, as measured by the scalability of the computation time and classification performance with the data size and number of computing workers. We use training sets of size $N = 12$ thousand, 120 thousand, and 1 million with test sets of 3 thousand, 30 thousand, and 250 thousand observations, respectively (20% of the total data size). All the experiments are conducted on the Duke Compute Cluster; due to limitations of the computing cluster, we only benchmark the time of DIB-C up to 30 workers and performance up to 120 workers.

When the data are distributed to a single node, global cluster refinement is not necessary; that is, a full MCMC for sampling (local) clustering is immediately followed by (global) cluster estimation. The prior setup is shared by all experiments from the same data set. In the sampling local clustering step, we run 1000 iterations of MCMC across all experiments, with the first 500 being burn-ins. Then we refine 100 iterations of local clustering samples, from which we randomly select 20 to be the clustering candidates considered for the global clustering estimation based on the variation of information loss. To sample the model parameters, we run 2000 iterations of MCMC conditional on the optimal subcomponent and cluster assignments.

2.4.1 Experimental setup

Synthetic data sets

The simulation setup originated in Baudry et al. (2010) and was later augmented by Malsiner-Walli et al. (2017). The data sets contain four clusters of varying shapes, including one triangle, one L, one cross, and one ellipse. They are generated from an eight-component Gaussian mixture in $\mathbb{R}^2$ with component means

$$(\mu_1, \ldots, \mu_8) = \begin{pmatrix} 6 & 4 & 8 & 22.5 & 20 & 22 & 22 & 6.5 \\ 1.5 & 6 & 6 & 1.5 & 8 & 31 & 31 & 29 \end{pmatrix},$$
Figure 2.5: Left: the scatter plot of a synthetic data set simulated with \( N = 12,000 \) observations. Right: the scatter plot of 10,000 observations randomly drawn from the 1 million training set of the flow cytometry data, where chl small, pe and fsc perp represent chlorophyll level, phycoerythrin level, and forward scatter respectively.

covariance matrices

\[
\Sigma_1 = \begin{pmatrix} 4.84 & 0 \\ 0 & 2.89 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 3.61 & 5.05 \\ 5.05 & 14.44 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 3.61 & -5.05 \\ -5.05 & 14.44 \end{pmatrix}, \quad \Sigma_4 = \begin{pmatrix} 12.25 & 0 \\ 0 & 3.24 \end{pmatrix}, \\
\Sigma_5 = \begin{pmatrix} 3.24 & 0 \\ 0 & 12.25 \end{pmatrix}, \quad \Sigma_6 = \begin{pmatrix} 14.44 & 0 \\ 0 & 2.25 \end{pmatrix}, \quad \Sigma_7 = \begin{pmatrix} 2.25 & 0 \\ 0 & 17.64 \end{pmatrix}, \quad \Sigma_8 = \begin{pmatrix} 2.25 & 4.20 \\ 4.20 & 16.00 \end{pmatrix},
\]

cluster weight vector \( \eta = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}) \) and subcomponent weight vectors \( \omega_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), \omega_2 = (\frac{1}{2}, \frac{1}{2}), \omega_3 = (\frac{1}{2}, \frac{1}{2}), \omega_4 = 1. \) Figure 2.5 shows a scatter plot of a data set simulated from this setup.

In our experiment, we simulate 10 data sets for each sample size. To estimate the model, we let \( K = 10 \) and \( L = 3. \)

Flow cytometry data

We consider high-frequency, continuous flow cytometry data collected from particles in aquatic environments by SeaFlow instruments (Hyrkas et al., 2016). Specifically, the SeaFlow instruments continuously sample surface seawater, generating a time series of cytometry samples (one every 3 minutes) containing measurements of the optical properties, including light scatter and intrinsic
fluorescence, of small phytoplankton cells during a research cruise (Hyrkas et al., 2016). The data set (Armbrust Lab, 2015) contains four optical measurements: forward scatter, side scatter, phycoerythrin level, and chlorophyll level. One important problem of interest is to classify phytoplankton cells based on their optical measurements.

Currently the dominant classification method in this application area is manual gating, where a scientist manually identifies the physical boundaries for clusters of cells. This process is subjective and can be infeasible given the massive amount of data collected during a research cruise (Hyrkas et al., 2016). In addition, properties of the same phytoplankton species may vary as environmental conditions (e.g., daylight) change over time and space (Sosik et al., 2010), leading to variation in the shapes and centers of the clusters.

Figure 2.5 shows a scatter plot of the three variables (side scatter, phycoerythrin and chlorophyll) that explain most variation in the data. The plot includes a random sample from the first one million observations with the variables on a logarithmic scale. We note that synecho, a species of phytoplankton, has visually disjoint components, but nano and ultra, two distinct species, are not well separated. In addition, crypto is a rare cluster with wide spread. All these factors increase the difficulty of clustering this data set.

A recent state-of-the-art clustering method for flow cytometry data first segments the data based on visual inspection of a change point in cluster formation and independently clusters these segments using a Gaussian mixture model with a pre-determined number of clusters (Hyrkas et al., 2016). In evaluating the clustering performance, they excluded observations whose assignment probability to any cluster is less than 0.7 and benchmarked the remaining observations against their manually-gated labels; this procedure resulted in an average F-measure of 0.864. Our method, in contrast, does not require a pre-determined number of clusters as an input or exclude observations based on their assignment probabilities.

To avoid time dependent variation in the clusters and given limited computing resources, we only use the first 1.25 million observations from the data set and similarly benchmark against the manually-gated labels. Since the data are partially labelled, the training set contains around 3.5% unlabeled data for each data size setting $N$. These unlabeled data are used in model fitting, but are
excluded in clustering performance evaluation. Noticing that each variable is highly right skewed, we transform the four variables to their logarithmic scales for ease of approximation. We set the upper bound of the number of clusters $K = 8$ and $L = 2$ for the experiments.

### 2.4.2 Results

This section includes experiment results for both the synthetic and flow cytometry data. Reported results include benefits from the global clustering refinement algorithm, classification performance, simulations from the posterior predictive distributions, and computation times.

The classification performance is evaluated for both the training and test sets by three frequently used cluster validation metrics—accuracy, adjusted Rand index (ARI) (Hubert and Arabie, 1985) and F-measure; see Appendix .4 for the definitions. When two partitions agree perfectly, accuracy, ARI and F-measure take value 1; in general, higher values of these statistics represent better clustering performance. To evaluate the test set performance, we first find an optimal mapping that minimizes the number of mismatches between the labels of the optimal clustering estimate of the training set and the manually-gated labels, so that each label refers to the same cluster in both the training and test set. The extra clusters identified in the estimates are coded as $unknown1$, $unknown2$ and so on. All results, unless explicitly stated, are averaged across 10 replicated experiments.

**Benefits from global cluster refinement algorithm**

Global cluster refinement has benefits beyond label alignment; its general ability to improve the clustering performance for each subset is illustrated by Figure 2.6 and 2.7. Figure 2.6 illustrates the increase in accuracy associated with the subsets after this procedure. Specifically, for each node setting, we randomly select a run from the ten replicates and draw a violin plot of the changes in accuracy associated with each subset. The before-refinement accuracy is identified by the optimal iteration that minimizes the expected loss locally, whereas the after-refinement accuracy is identified by the final global clustering estimate mapped to local subsets. We notice a general gain in accuracy, which is especially prominent for the flow cytometry data with one million observations.

The benefits from global cluster refinement are also corroborated by Figure 2.7, which shows
Figure 2.6: Increase in classification accuracy after refinement in local subsets of the training sets from the synthetic (top row) and the flow cytometry data (bottom row). 12k, 120k and 1m indicate the data size of the training sets. For each data and node setting, we randomly select one of ten replicates and include a violin plot of $R$ statistics, where $R$ is the number workers (or subsets).
Figure 2.7: Scatter plots of clustering results associated with the training set of one of the synthetic data sets with \( N = 10^6 \). The plot on the left represents the final clustering estimates when the data are randomly distributed to 1, 5, 20 and 120 workers respectively. The plot on the right shows subsets results associated with the 20 workers on the left; specifically, it represents a sample of cluster allocations of subset 5, 8, 15 and 20 before global cluster refinement.

The clustering results of selected subsets before refinement (in the bottom row) and the entire data set after refinement (in the top row) using 20 workers. The spurious clusters, coming in various shapes and locations, are identified in local subsets but are eliminated from the final clustering estimate by the global cluster refinement step.

Classification performance

DIB-C exhibits promising classification performance, as measured by accuracy, ARI and F-measure, for both training and test sets, with results illustrated in Figure 2.7, 2.8, 2.9 and Table 2.4. Figure 2.8 shows that these measures are robust to the growing number of workers for large (i.e. \( N = 120k \) and \( 1m \)) data sets; in fact, they increase steadily for the data sets with 1 million observations as the number of workers increases to 120, likely due to the improved mixing of MCMC as the sample size per node drops. This may also explain the sharp rises in performance as we increase the number of workers from one to two. These observations indicate a win-win position: DIB-C is not only scalable, but also robust to the increasing number of workers.

Figure 2.7 and 2.9 show the scatter plots of the clustering results associated with a training set
**Figure 2.8**: Classification performance of the training and test sets associated with the synthetic (left) and flow cytometry (right) data, as measured by accuracy, ARI, and F-measure.
Figure 2.9: Scatter plots of clustering results of the test set associated with $N = 1$ million data size setting. The top left plot corresponds to the manually-gated labels. The remaining four represent the clustering results from using 1, 60 and 100 workers respectively. In each scatter plot, only a random subset of the data points are included; and the $x$, $y$ and $z$ coordinates are $\log$(side scatter), $\log$(phycoerythrin level) and $\log$(chlorophyll level), respectively. These three explain the most variation in the data among the four variables (according to principle component analysis results).
of the synthetic data and a test set of the flow cytometry data, respectively, when $N = 1$ million. As evidenced by both figures, the clustering tends to be less noisy and visually more reasonable as the number of workers increases. In Figure 2.9, the names of the species generally match their manually-gated labels when the number of workers is large, with synecho—which is disconnected in the manually-gated labels—decomposed into two clusters.

We have also compared DIB-C with popular distributed clustering algorithms, including DBSCAN (density-based spatial clustering of applications with noise) and K-means. In terms of the tuning parameters, for DBSCAN, we set $\epsilon$ based on a widely used k-nearest neighbour distance plot and $\text{minPts}$ at the value that maximizes the accuracy, ARI and F-measure for the training set (they all happen to be maximized by the same $\text{minPts}$); for K-means, we set the number of clusters to be 5 (the truth according to the manual gating), 6 and 8 respectively. The distributed version of DBSCAN and K-means are much faster than DIB-C and yield the same clustering performance as their non-distributed counterparts. However, their performance, as shown in Table 2.4, is very sensitive to tuning parameters; taking K-means as an example, the slight deviation of $K$ from the truth leads to drastically worse performance. In addition, DIB-C, for which the median performance statistics are reported, consistently outperforms all methods with the exception of K-means when the number of clusters is correctly specified. However, even K-means with the number of clusters correctly specified cannot provide a generative model, based on which we can simulate similar data and perform density estimation and other inference. Our method, however, enjoys such an advantage, as is shown in Section 2.4.2.

We also attempted comparison with Dirichlet-process Gaussian mixtures, another widely used model-based approach for clustering. Since the computation was already extremely slow for $N = 12k$ (taking over 20 hours), we dropped the comparison.

**Simulation from posterior predictive**

Figure 2.10 shows scatter plots of 10,000 data points simulated from the posterior predictive distributions resulting from analyses in which the synthetic and the flow cytometry data are randomly distributed across 40 workers. The posterior predictive distributions show striking resemblance to
<table>
<thead>
<tr>
<th>Method</th>
<th>12k</th>
<th>120k</th>
<th>1m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train</td>
<td>test</td>
<td>train</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>ARI</td>
<td>F-measure</td>
</tr>
<tr>
<td>DIB-C</td>
<td>0.938</td>
<td>0.849</td>
<td>0.955</td>
</tr>
<tr>
<td>DBSCAN-default</td>
<td>0.719</td>
<td>0.350</td>
<td>0.691</td>
</tr>
<tr>
<td>DBSCAN-optimal</td>
<td>0.909</td>
<td>0.803</td>
<td>0.895</td>
</tr>
<tr>
<td>(minPts = 40)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means, $K = 5$</td>
<td>0.977</td>
<td>0.945</td>
<td>0.980</td>
</tr>
<tr>
<td>K-means, $K = 6$</td>
<td>0.699</td>
<td>0.593</td>
<td>0.796</td>
</tr>
<tr>
<td>K-means, $K = 8$</td>
<td>0.621</td>
<td>0.537</td>
<td>0.748</td>
</tr>
<tr>
<td>DIB-C</td>
<td>0.966</td>
<td>0.938</td>
<td>0.971</td>
</tr>
<tr>
<td>DBSCAN-default</td>
<td>0.904</td>
<td>0.793</td>
<td>0.886</td>
</tr>
<tr>
<td>DBSCAN-optimal</td>
<td>0.904</td>
<td>0.793</td>
<td>0.886</td>
</tr>
<tr>
<td>(minPts = 5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means, $K = 5$</td>
<td>0.979</td>
<td>0.950</td>
<td>0.982</td>
</tr>
<tr>
<td>K-means, $K = 6$</td>
<td>0.706</td>
<td>0.706</td>
<td>0.803</td>
</tr>
<tr>
<td>K-means, $K = 8$</td>
<td>0.625</td>
<td>0.543</td>
<td>0.751</td>
</tr>
<tr>
<td>DIB-C</td>
<td>0.913</td>
<td>0.868</td>
<td>0.901</td>
</tr>
<tr>
<td>DBSCAN-default</td>
<td>0.718</td>
<td>0.356</td>
<td>0.692</td>
</tr>
<tr>
<td>DBSCAN-optimal</td>
<td>0.904</td>
<td>0.791</td>
<td>0.885</td>
</tr>
<tr>
<td>(minPts = 12)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means, $K = 5$</td>
<td>0.979</td>
<td>0.949</td>
<td>0.982</td>
</tr>
<tr>
<td>K-means, $K = 6$</td>
<td>0.708</td>
<td>0.599</td>
<td>0.803</td>
</tr>
<tr>
<td>K-means, $K = 8$</td>
<td>0.630</td>
<td>0.544</td>
<td>0.755</td>
</tr>
</tbody>
</table>

**Table 2.4**: The clustering performance of the flow cytometry data across various methods for $N = 12k$, 120k and 1m. In particular, the performance of DBSCAN and K-means are invariant to the number of workers. For DIB-C, the median of each statistic obtained from using varying number of workers is reported. Yellow and green indicate the best and the second best performance under each scenario respectively.
Figure 2.10: Scatter plots of 10,000 data points simulated from the posterior predictive distributions resulting from an analysis conducted with data distributed across 40 workers for the synthetic data (left) and the flow cytometry data (right). In the plot on the right, chl small, pe and fsc perp represent chlorophyll level, phycoerythrin level, and forward scatter respectively.

the true data; the cluster labels, created by finding a map that minimizes the number of mismatches to the true or manually-gated labels, also largely match those in the original data set.

Computation time

Figure 2.11 displays the computation time for sampling local clustering, global cluster refinement, global clustering estimation and the total run time, respectively. We notice similar trends in both data settings, most notably the drastic decrease in the total computation time of the data sets with one million observations; the dive, quickly plateauing at around 15 workers, is mostly attributable to a significant speedup, dropping from around 8.5 hours to 15 minutes, from parallel MCMC in sampling local clustering. No global cluster refinement or global clustering estimation is necessary in a conventional single machine scenario, which explains the jump in the associated time when switching to two workers.
**Figure 2.11:** The computation time associated with the training sets of the synthetic (top) and flow cytometry data (bottom). For each data setting, we include sampling local clustering (the first column), global cluster refinement (the second column), global clustering estimation time (the third column) and total time of the above steps (the fourth column).
2.5 Conclusion

In this paper, we have proposed a nearly embarrassingly parallel framework named DIB-C for distributed Bayesian clustering under a finite mixture of Gaussian mixtures model. DIB-C accommodates any loss function on the space of partitions for cluster estimation, quickly classifies future subjects, and allows density estimation and other posterior inference. Our extensive experiments also demonstrate that DIB-C is not only scalable, but also robust to the increasing number of workers for large data sets.
.1 MCMC procedures for sampling of local clustering (step 2)

We adapt Malsiner-Walli et al. (2017)’s MCMC sampling scheme for finite mixture of mixtures for sampling of local clustering (step 2) and model parameters (step 5) in DIB-C. The sampling scheme is based on a block conditional Gibbs sampler with data augmentation. In this section, we introduce procedures for sampling of local clustering (step 2).

Assume $\mathcal{Y}$ can be partitioned into $R$ non-overlapping subsets, with subset $r$, denoted by $\mathcal{Y}_r$, residing on worker $r$ ($r = 1, \ldots, R$). We run MCMC on $R$ workers in an embarrassingly parallel manner, producing draws from subset posterior $p_r(\Theta, \eta \mid \mathcal{Y}_r)$, $r = 1, \ldots, R$. Without loss of generality, we introduce the sampling for subset $r$. Recall the definition of subset posterior $r$

$$p_r(\Theta, \eta \mid \mathcal{Y}_r) = \frac{\{ \prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta) \} p(\Theta, \eta)}{\int \{ \prod_{i=1}^{n_r} f(y_{ri} \mid \Theta, \eta) \} p(\Theta, \eta) d\Theta d\eta}.$$ 

Recall the vector of latent cluster allocations $c_r = (c_{r1}, \ldots, c_{rn_r})$ is the vector of latent cluster allocations, which take values in $\{1, \ldots, K\}^{n_r}$, indicating the cluster to which each observation belongs such that

$$p(y_{ri} \mid \theta_1, \ldots, \theta_K, c_{ri} = k) = p_k(y_{ri} \mid \theta_k), \quad \text{and} \quad P\{c_{ri} = k \mid \eta\} = \eta_k.$$ 

Recall the vector of latent subcomponent allocations $s_r = (s_{r1}, \ldots, s_{rn_r})$, which take values in $\{1, \ldots, L\}^{n_r}$, to indicate the subcomponent to which an observation within a cluster is assigned such that

$$p_k(y_{ri} \mid \theta_k, c_{ri} = k, s_{ri} = l) = f_N(y_{ri} \mid \mu_{kl}, \Sigma_{kl}), \quad \text{and} \quad P\{s_{ri} = l \mid c_{ri} = k, w_k\} = \omega_{kl}.$$ 

Using the priors specified in Section 2.2, the sampling steps are given as follows:

1. **Sampling steps on the level of the cluster distribution:**

   A.1 Parameter simulation step conditional on the classification $c_r$. Sample $\eta \mid c_r$ from $\text{Dir}(e_1, \ldots, e_K)$, $e_k = e_0 + n_k$, $k = 1, \ldots, K$, where $n_k = \#\{i : c_{ri} = k\}$ is the number of observations allocated to cluster $k$. 

115
A.2 Classification step for each observation $y_{ri}$ conditional on cluster-specific parameters.

For each $i = 1, \ldots, n_r$ sample the cluster assignment $c_{ri}$ from

$$P\{c_{ri} = k \mid y_{ri}, \theta, \eta\} \propto \eta_k p_k(y_{ri} \mid \theta_k), k = 1, \ldots, K,$$

where $p_k(y_{ri} \mid \theta_k)$ is the semi-parametric mixture approximation of the cluster density:

$$p_k(y_{ri} \mid \theta_k) = \sum_{l=1}^L w_{kl} f_N(y_{ri} \mid \mu_{kl}, \Sigma_{kl}).$$

2. Within each cluster $k, k = 1, \ldots, K$:

B.1 Classification step for all observations $y_{ri}$ that are assigned to cluster $k$ (i.e. $c_{ri} = k$), conditional on the subcomponent weights and the subcomponent-specific parameters.

For each $i \in \{i : c_{ri} = k\}$ sample $s_{ri}$ from

$$Pr\{s_{ri} = l \mid y_{ri}, \theta_k, c_{ri} = k\} \propto w_{kl} f_N(y_{ri} \mid \mu_{kl}, \Sigma_{kl}), l = 1, \ldots, L.$$

B.2 Parameter simulation step conditional on the classifications $s_r$ and $c_r$:

i. Sample $w_k \mid c_r, s_r$ from $\text{Dir}(d_{k1}, \ldots, d_{kL})$, $d_{kl} = d_0 + n_{kl}, l = 1, \ldots, L$, where $n_{kl} = \#\{i : s_{ri} = l, c_{ri} = k\}$ is the number of observations allocated to subcomponent $l$ in cluster $k$.

ii. For $l = 1, \ldots, L$: Sample $\Sigma_{kl}^{-1} \mid c_r, s_r, \mu_{kl}, C_0 k, \mathcal{Y}_r \sim \mathcal{W}_d(c_{kl}, C_{kl})$, where

$$c_{kl} = c_0 + n_{kl},$$

$$C_{kl} = C_0 k + \sum_{i : s_{ri} = l, c_{ri} = k} (y_{ri} - \mu_{kl}) (y_{ri} - \mu_{kl})^T.$$ 

iii. For $l = 1, \ldots, L$: Sample $\mu_{kl} \mid c_r, s_r, b_{0k}, \Sigma_{kl}, \Lambda_k, \mathcal{Y}_r \sim \mathcal{N}_d(b_{kl}, B_{kl})$, where

$$B_{kl} = \left(\tilde{B}_{0k}^{-1} + n_{kl} \Sigma_{kl}^{-1}\right)^{-1},$$

$$b_{kl} = B_{kl} \left(\tilde{B}_{0k}^{-1} b_{0k} + \Sigma_{kl}^{-1} n_{kl} \bar{y}_{kl}\right),$$

with $\tilde{B}_{0k} = \sqrt{\Lambda_k} B_0 \sqrt{\Lambda_k}, \Lambda_k = \text{diag}(\lambda_k1, \ldots, \lambda_kd)$, and

$$\bar{y}_{kl} n_{kl} = \sum_{i : s_{ri} = l, c_{ri} = k} y_{ri}.$$
3. For each cluster \( k, k = 1, \ldots, K \): Sample the random hyperparameters \( \lambda_{kj}, C_{0k}, b_{0k} \) from their full conditionals:

**C.1** For \( j = 1, \ldots, d \): Sample \( \lambda_{kj} \mid b_{0k}, \mu_{k1}, \ldots, \mu_{kL} \sim \text{GIG}(p_{kL}, a_{kj}, b_{kj}) \), where GIG is the generalized inverse Gaussian distribution and

\[
p_{kL} = -L/2 + \nu,
\]

\[
a_{kj} = 2\nu,
\]

\[
b_{kj} = \sum_{l=1}^{L} (\mu_{kl,j} - b_{0k,j})^2 / B_{0,jj}.
\]

**C.2** Sample \( C_{0k} \mid \Sigma_{k1}, \ldots, \Sigma_{kL} \sim W_d \left( g_0 + Lc_0, G_0 + \sum_{l=1}^{L} \Sigma_{kl}^{-1} \right) \).

**C.3** Sample \( b_{0k} \mid \tilde{B}_{0k}, \mu_{k1}, \ldots, \mu_{kL} \sim N_d \left( \tilde{m}_k, \tilde{M}_k \right) \), where

\[
\tilde{M}_k = \left( M_0^{-1} + L\tilde{B}_{0k}^{-1} \right)^{-1},
\]

\[
\tilde{m}_k = \tilde{M}_k \left( M_0^{-1} m_0 + \tilde{B}_{0k}^{-1} \sum_{l=1}^{L} \mu_{kl} \right).
\]

### .2 Collapsed Gibbs sampler for global cluster refinement

The key quantity in updating the group allocation is the posterior probability of item \( b \) being assigned to group \( h \):

\[
P(z_b = h \mid z_{\setminus b}, \mathcal{Y}) \propto P(z_b = h \mid z_{\setminus b}) P(\mathcal{Y} \mid z_b = h, z_{\setminus b})
\]

\[
= P(z_b = h \mid z_{\setminus b}) P(y_b \mid \mathcal{Y}_{\setminus b}, z_b = h, z_{\setminus b}) P(\mathcal{Y}_{\setminus b} \mid z_b = h, z_{\setminus b})
\]

\[
\propto P(z_b = h \mid z_{\setminus b}) P(y_b \mid \mathcal{Y}_{\setminus b}, z_b = h, z_{\setminus b}), \quad (22)
\]

where \( h = 1, \ldots, H; b = 1, \ldots, B \). Note that

\[
P(z) = \frac{\Gamma(H\alpha_0)}{\Gamma(N + H\alpha_0)} \prod_{h=1}^{H} \frac{\Gamma(N_h + \alpha_0)}{\Gamma(\alpha_0)},
\]
so

\[ P(z_b = h, z_{\backslash b}) = \frac{\Gamma(H_0) \Gamma(N_{h \backslash b} + n_b + \alpha_0)}{\Gamma(N + H_0) \Gamma(\alpha_0)} \prod_{j=1,j \neq h}^{H} \frac{\Gamma(N_j + \alpha_0)}{\Gamma(\alpha_0)}, \] (23)

\[ P(z_{\backslash b}) = \frac{\Gamma(H_0) \Gamma(N - n_b + H_0)}{\Gamma(N) \Gamma(\alpha_0)} \prod_{j=1,j \neq h}^{H} \frac{\Gamma(N_j + \alpha_0)}{\Gamma(\alpha_0)}. \] (24)

hold. By (23) and (24), the first term in (22) is

\[ P(z_b = h | z_{\backslash b}) = \frac{P(z_b = h, z_{\backslash b})}{P(z_{\backslash b})} = \frac{\Gamma(N + H \alpha_0 - n_b) \Gamma(N_{h \backslash b} + n_b + \alpha_0)}{\Gamma(N + H \alpha_0) \Gamma(N_{h \backslash b} + \alpha_0)}. \]

Let \( y^* \) be a new data vector.

\[ p(y^* | \mathcal{Y}) = t_d \left( y^* | m_N, \frac{\kappa_N + 1}{\kappa_N (\nu_N - d + 1)} S_N, \nu_N - D + 1 \right), \] (25)

where \( m_N = N\bar{y}/\kappa_N, \kappa_N = 1 + N, \nu_N = \nu_0 + N, \) and \( S_N = S_0 + \sum_{n=1}^{N} y_n y_n^T - \kappa_N m_N m_N^T. \)

The second term in (22) can be re-expressed as

\[ p(y_b | \mathcal{Y}_{\backslash b}, z_b = h, z_{\backslash b}) = p(y_b | \mathcal{Y}_{h \backslash b}), \]

which is given by a product of \( t \)-densities based on (25)

\[ p(y_b | \mathcal{Y}_{h \backslash b}) = \prod_{j=1}^{n_b} t_d \left( y_{b,j} | m_{h \backslash b}, \frac{\kappa_{h \backslash b} + 1}{\kappa_{h \backslash b} (\nu_{h \backslash b} - d + 1)} S_{h \backslash b}, \nu_{h \backslash b} - d + 1 \right), \]

where

\[ \kappa_{h \backslash b} = 1 + N_{h \backslash b}, \quad \nu_{h \backslash b} = \nu_0 + N_{h \backslash b}, \]

\[ m_{h \backslash b} = \frac{N_{h \backslash b} y^T_{h \backslash b}}{\kappa_{h \backslash b}} \quad \text{and} \quad S_{h \backslash b} = S_0 + N_{h \backslash b} S_{h \backslash b} - \kappa_{h \backslash b} m_{h \backslash b} m_{h \backslash b}^T. \]

.3 Sampling model parameters (step 5)

Algorithm 3 outlines the steps in approximate posterior sampling of model parameters (step 5).

**Algorithm 3 Sampling Model Parameters**

**Input:** Output of refined samples of local cluster allocations from Algorithm 1

**Output:** Optimal clustering estimate \( c^* \)

118
1: On Workers:
2: \textbf{parfor} Worker $r = 1$ to $R$ do
3: \hspace{1em} for $k = 1$ to $k_N$ do
4: \hspace{2em} for $l = 1$ to $L$ do
5: \hspace{3em} $n_{rkl} \leftarrow \sum_{y_i \in Y_r} \mathbb{1}_{\tilde{s}_i = l, \tilde{c}_i = k}$
6: \hspace{3em} $G_{rkl} \leftarrow \sum_{y_i \in Y_r} y_i y_{i}^{T}$
7: \hspace{3em} $\delta_{rkl} \leftarrow \sum_{y_i \in Y_r} y_i$
8: \hspace{3em} Send $n_{rkl}, G_{rkl}$ and $\delta_{rkl}$ to Master
9: \hspace{2em} end for
10: \hspace{1em} end for
11: end parfor
12: On Master:
13: for $k = 1$ to $k_N$ do
14: \hspace{1em} for $l = 1$ to $L$ do
15: \hspace{2em} $n_{kl} \leftarrow \sum_{r} n_{rkl}$, $G_{kl} \leftarrow \sum_{r} G_{rkl}$ and $s_{kl} \leftarrow \sum_{r} s_{rkl}$
16: \hspace{2em} end for
17: \hspace{1em} $n_{k} \leftarrow \sum_{l} n_{kl}$
18: end for
19: Initialize $(G(0), \eta(0))$ \hspace{1em} \textgreater{} Start of the sampling; see Appendix .1 for more details
20: for Iteration $t = 1$ to $T$ do
21: \hspace{1em} Sample $\eta^{(t)}$ from Dir $(e_1, \ldots, e_{k_N})$, where $e_k = e_0 + n_k$ \hspace{1em} See Appendix .1 A.1
22: \hspace{1em} for $k = 1$ to $k_N$ do
23: \hspace{2em} \hspace{1em} \hspace{1em} \hspace{1em} \textgreater{} See Appendix .1 B.2
24: \hspace{2em} Sample $\omega^{(t)}_k$ from Dir$(d_{k1}, \ldots, d_{kL})$, where $d_{kl} = d_0 + n_{kl}$
25: \hspace{2em} for $l = 1$ to $L$ do
26: \hspace{3em} Sample $(\Sigma_{kl}^{-1})^{(t)}$ from $W_d(c_{kl}, C_{kl})$, where $c_{kl} = c_0 + n_{kl}$, and $C_{kl} = C_{0k}^{(t-1)} + G_{kl} - 2 \mu_{kl} s_{kl} + n_{kl} \mu_{kl}^{(t-1)} (\mu_{kl}^{(t-1)})^T$
27: Sample $\mu_{kl}^{(t)}$ from $N_d(b_{kl}, B_{kl})$, where $B_{kl} = \left( \tilde{B}_{0k}^{-1} + n_{kl}(\Sigma_{kl}^{-1})^{(t)} \right)$, and $b_{kl} = B_{kl} \left( \tilde{B}_{0k}^{-1} b_{0k}^{(t-1)} + (\Sigma_{kl}^{-1})^{(t)} n_{kl} \right)$

28: end for

29:▷ See Appendix .1 3

30: for $j = 1$ to $d$ do

31: Sample $\lambda_{kj}$ from $GIG(p_{kl}, a_{kj}, b_{kj})$, where $b_{kj} = \sum_l \left( \mu_{kl,j}^{(t-1)} - b_{0k,j}^{(t-1)} \right)^2 / B_{0,jj}$

32: end for

33: Sample $C_{0k}$ from $W_d \left( g_0 + Lc_0, G_0 + \sum_{l=1}^{L} (\Sigma_{kl}^{-1})^{(t)} \right)$

34: Sample $b_{0k}$ from $N_d \left( \tilde{m}_k^{(t)}, \tilde{M}_k^{(t)} \right)$

35: end for

36: end for

37: return $\{ (\Theta^{(t)}, \eta^{(t)}) : t = t_0 + 1, \ldots, T \}$, where first $t_0$’s are burn-in iterations

.4 Definition of clustering validation metrics

Write true positive, true negative, false positive and false negative as TP, TN, FP and FN respectively.

The definition of accuracy, F-measure and adjusted Rand index is given as follows:

1. Accuracy

\[
\text{Accuracy} = \frac{\#TP + \#TN}{\#\text{observations}}
\]

2. F-measure

Precision and Recall are defined as follows:

\[
\text{Precision} = \frac{\#TP}{\#TP + \#FP}, \quad \text{and} \quad \text{Recall} = \frac{\#TP}{\#TP + \#FN}
\]

F-measure is the harmonic mean of precision of recall:

\[
\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.
\]
3. Adjusted Rand index (ARI) (Hubert and Arabie, 1985)

Adjusted Rand index is a corrected-for-chance version of Rand index; the definition of Rand index is similar to accuracy. See Hubert and Arabie (1985) for a formal definition of adjusted Rand index.
Chapter 3

Curved factor analysis with the Ellipsoid-Gaussian distribution

Key words: Dimensionality reduction; Ellipse; Latent factors; PCA; Sphere; von Mises-Fisher distribution

3.1 Introduction

The multivariate Gaussian distribution is routinely used, relying on a rich collection of methods for inference on the covariance structure. Factor analysis is particularly popular, due to its combination of simplicity and flexibility. The generic form of a Gaussian linear factor model is

\[ x_i = c + \Lambda \eta_i + \epsilon_i, \quad \epsilon_i \sim N_p(0, \Sigma), \quad (i = 1, \ldots, n), \]

where \( x_i \) is \( p \)-dimensional, \( \Lambda \) is a \( p \times k \) factor loadings matrix, \( \eta_i \sim N_k(0, I_k) \) are latent factors and \( \epsilon_i \) is an idiosyncratic error with covariance \( \Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2) \). Conditional on the factors, the elements of \( x_i \) are independent; dependence is induced by marginalizing out the latent factors to obtain \( x_i \sim N_p(c, \Omega) \) with \( \Omega = \Lambda \Lambda^T + \Sigma \).

In genomics, for instance, \( x_i \) can be a massive vector of gene expression values; with genes within common pathways tending to co-express, it is natural to regard \( \eta_i \) as pathway characterizing factors (Carvalho et al., 2008). In practice, one typically chooses \( k \ll p \). It is reasonable to suppose that the loadings matrix contains many zero entries, so that any single factor only impacts a relatively small number of elements of \( x_i \). Hence, dimension reduction is often carried out through low rank and sparsity assumptions on the loadings matrix. See, for instance, West (2003); Ghosh and Dunson (2009); Bhattacharya and Dunson (2011) and Ma (2013).

Our work begins from the observation that the multivariate Gaussian distribution cannot characterize nonlinear relationships in data. Consider the Gaussian linear factor model. To generate data from \( x_i \sim N_p(0, \Lambda \Lambda^T + \Sigma) \), we can first generate data from \( N_p(0, \Lambda \Lambda^T) \) and then add random
Figure 3.1: Demonstration of the Gaussian linear factor model with $p = 3, k = 2$. The yellow plane represents the column space of $\Lambda$ and contains all of the blue points sampled from $N_3(0, \Lambda\Lambda^T)$. The red vectors represent eigenvectors of $\Lambda\Lambda^T$; equivalently the left singular vectors of $\Lambda$. The figure on the right shows the view perpendicular to the plane, illustrating that the data have a Gaussian distribution centered around the plane.

However, curved relationships are commonplace in real data. We provide two motivating examples; one is a speed flow data set on a California freeway (Einbeck and Dwyer, 2011) and the other is an ecological data set of horse mussels in New Zealand (Camden, 1989). Figure 3.2a shows that vehicle flow, per 5 minutes, and speed, in miles per hour, on the freeway have a curved relationship. Figure 3.2b shows that body measurements, such as edible muscle mass (M) and shell width (W), of the horse mussels have curved dependence.

There is a rich literature on nonlinear factor models that can capture curved dependence by replacing the linear loadings on $\eta_i$ with a more complex mapping. A general model is given by
(a) Scatter plot of California freeway data. (b) Scatter plot matrix of horse mussel data.

**Figure 3.2**: Scatter plots of two curved data sets. In particular, H, L, S, W, and M represent shell height (mm), shell length (mm), shell mass (g), shell width (mm) and muscle mass (g) respectively.
\[ x_i = g(\eta_i) + \epsilon_i, \] with the same assumptions as a Gaussian linear factor model but a nonlinear function \( g \) mapping \( \mathbb{R}^k \to \mathbb{R}^p \). Popular nonparametric approaches include Gaussian process latent variables models, which give \( g(\cdot) \) a Gaussian process prior (Titsias and Lawrence, 2010; Li and Chen, 2016), and variational auto encoders, which model \( g(\cdot) \) using a deep neural network (Kingma and Welling, 2014; Rezende et al., 2014). Such methods are highly flexible but tend to be complex black boxes that have issues with reproducibility of results and non-identifiability, while being opaque to interpret.

An alternative is to mix Gaussian linear factor models to capture nonlinear structure in the data by local linear Gaussian models; for example, refer to the rich literature on mixtures of factor analyzers (Ghahramani and Hinton, 1997; McLachlan et al., 2003; Murphy et al., 2020). However, in defining a factor model for every component of a mixture model, such models are heavily parameterized and can be difficult to fit reliably and interpret. If there are not distinct clusters in the data but data have curved support, data will be broken up into many small clusters.

Other attempts to develop flexible parametric families of multivariate distributions have focused on capturing skewness and heavy tails in the data, among which the generalization of skew-elliptical distributions (Azzalini and Capitanio, 1999; Branco and Dey, 2001) constitutes a prominent subset. Starting with multivariate skew-normal distributions (Azzalini and Valle, 1996), there are extensions to multivariate skew-t (Gupta, 2003; Kim and Mallick, 2003; Arevalillo and Navarro, 2015), scale mixtures of skew-normal (Kim, 2008; Capitanio, 2020), skew-symmetric (Jupp et al., 2016) and even broader families (Genton and Loperfido, 2005; Azzalini and Capitanio, 2003; Landsman et al., 2017). An alternative extension of the skew-normal is the multivariate skew-slash distribution (Wang and Genton, 2006). Certain of the above distributions can induce curvature, but in an indirect and hard to interpret manner.

Hence, there is a need to develop simple parametric models to characterize nonlinear dependence in data. In this article, we propose a class of parametric factor models that induce multivariate distributions supported near the surface of a hyper-ellipsoid. This Ellipsoid-Gaussian class is induced by using a Gaussian linear factor model with latent variables following a von Mises-Fisher distribution on a unit hyper-sphere. We show that the Ellipsoid-Gaussian distribution is surpris-
ingly flexible in allowing curved relationships among variables and modeling of lower-dimensional structure, while including the Gaussian linear factor model as a special case. All proofs are in the Supplementary Material.

3.2 The von Mises-Fisher linear factor model

To induce curved relationships among \( x_{i1}, \ldots, x_{ip} \), a starting point is distributions on a sphere. Let \( S^{k-1} \) be a unit sphere in \( \mathbb{R}^k \) centered at the origin and \( S^{k-1} \) be the probability measure of the uniform distribution on \( S^{k-1} \). A simple density with respect to \( S^{k-1} \) for a random vector \( z \in S^{k-1} \) is

\[
f(z; \mu, \tau) = C_k(\tau) \exp(\tau \mu^T z),
\]

where \( \mu \) is the mean direction with \( \|\mu\| = 1 \), \( \tau \geq 0 \) is a concentration parameter, \( C_k(\tau) = (\tau/2)^{k/2-1} \Gamma(k/2) I_{k/2-1}(\tau) \) is the normalizing constant, and \( I_v(\cdot) \) denotes the modified Bessel function of the first kind of order \( v \). This von Mises-Fisher density is symmetric about the mean direction \( \mu \), with \( \tau \) controlling concentration—a larger \( \tau \) corresponds to higher concentration around \( \mu \). The inner product between two unit vectors \( \mu^T z \) parametrizes the cosine distance between \( \mu \) and \( z \). This metric has been used in high-dimensional directional data clustering, such as for text and gene-expression data (Banerjee et al., 2005; Reisinger et al., 2010; Gopal and Yang, 2014).

While the von Mises-Fisher distribution is not directly useful for the data \( x_i \), since these data points are not exactly on a sphere, it can be used for latent factors in model (3.1) as follows:

\[
x_i = c + \Lambda \eta_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}_p(0, \Sigma), \quad \eta_i \sim \text{vMF}(\mu, \tau), \quad i = 1, \ldots, n. \tag{3.2}
\]

To better understand this model, we first define some notation. Let the singular value decomposition of \( \Lambda \) be \( U_{p \times k} S_{k \times k} V_{k \times k}^T \), where \( U^T U = I, V^T V = I \) and \( S_{k \times k} = \text{diag}(s_1, \ldots, s_k) \). The image of a unit sphere \( S^{k-1} \) under the affine transformation \( c + \Lambda S^{k-1} \) is a \( k \)-dimensional ellipsoid in \( \mathbb{R}^p \); when \( k < p \), the ellipsoid is degenerate. The center of the ellipsoid is \( c \), its principal axes are represented by the left singular vectors \( U \), and its semi-axes lengths by the singular values \( S \). The image of \( S^{k-1} \) after multiplication by \( \Lambda \) can be understood as sequentially transforming the sphere by each matrix in the factorization: since the sphere is rotationally invariant, the image \( S^{k-1} \) under the rotation \( V^T \) is still itself; \( S \) stretches the sphere \( S^{k-1} \) into an ellipsoid centered at the origin.
Data generated from a noise-less von-Mises Fisher linear factor model with center $c$ at the origin.

(b) Data generated by adding random noise from $N_3(0, \Sigma)$ to the points in Fig. 3.3a

Figure 3.3: Demonstration of the von Mises-Fisher linear factor model with $p = 3, k = 2$. The green and red lines represent the mean direction $\Lambda\mu$ and principal axes of the ellipse, respectively.

with principal axes parallel to the coordinate axes and semi-axes lengths equal to the singular values $s_k$; finally, $U$ rotates the ellipsoid such that the principal axes are parallel to the column vectors of $U$ and embeds it into a $k$-dimensional linear subspace of $\mathbb{R}^p$. Adding $c$ translates the ellipsoid such that its center becomes $c$. We then add Gaussian noise.

To visualize data generated from this model, consider an example with $p = 3, k = 2$. If we generate noiseless data from the model with $c = 0$, the points will lie on an ellipse as shown in Fig. 3.3a, with the center of the ellipse at the origin and the shape of the ellipse determined by $\Lambda$. The red line segments represent the principal axes of the ellipsoid and are also the left singular vectors of $\Lambda$. We then add Gaussian noise around the points in Fig. 3.3a to obtain the desired distribution, as visualized in Fig. 3.3b, where the points are distributed around a curved surface, the ellipse, and this is how the curvature is captured in the model.

Figure 3.4 shows more simulated examples from this model. The various shapes of the data clouds suggest tremendous flexibility of the model in accommodating different data — data that exhibit symmetry, asymmetry, uneven curvature or data that are close to Gaussian.
Figure 3.4: Scatter plots of data sampled from the Ellipsoid-Gaussian in $\mathbb{R}^3$ with varying parameters.
3.3 Ellipsoid-Gaussian distribution

Analogous to the Gaussian latent factor model, we wish to marginalize over the distribution of the latent factor \( \eta \) to obtain the marginal distribution of \( x \); we call the resulting distribution Ellipsoid-Gaussian. The density of \( x \) can be calculated in closed-form, and \( x \) is supported on the entirety of \( \mathbb{R}^p \). In order to state the density, we recall the Fisher-Bingham distribution, which is a distribution on \( S^{k-1} \) that includes the von Mises-Fisher distribution as a special case. A random vector \( y \) with a Fisher-Bingham distribution has density with respect to \( S_{k-1} \):

\[
f(y; \kappa, \vartheta, A) = \frac{1}{\varsigma(\kappa \vartheta, A)} \exp \left( \kappa \vartheta^T y - y^T A y \right),
\]

where \( y, \vartheta \in S^{k-1}, \kappa \geq 0, A \in \mathbb{R}^{p \times p} \) is a symmetric matrix, and \( \varsigma(\kappa \vartheta, A) \) is a normalizing constant that is commonly estimated via a saddlepoint approximation (Kume and Wood, 2005) which is accurate, fast and numerically stable. The Fisher-Bingham distribution has been studied in Wood (1988); Hoff (2009); Kent et al. (2018) and many other works.

Proposition 6 shows the marginal density of the Ellipsoid-Gaussian distribution.

**Proposition 6.** Assume \( x = c + \Lambda \eta + \epsilon, \epsilon \sim N_p(0, \Sigma), \eta \sim vMF(\mu, \tau), \) and \( \Sigma = \text{diag} (\sigma_1^2, \ldots, \sigma_p^2) \).

1. Marginalizing over the distribution of \( \eta \) yields density

\[
f_{\text{EG}}(x) = \frac{C_k(\tau)}{(2\pi)^{\frac{p}{2}} \prod_{i=1}^p \sigma_i} \exp \left\{ -\frac{1}{2} (x-c)^T \Sigma^{-1} (x-c) \right\} \\
\varsigma \left\{ \tau \mu + \Lambda^T \Sigma^{-1} (x-c), \frac{\Lambda^T \Sigma^{-1} \Lambda}{2} \right\},
\]

with respect to the Lebesgue measure on \( \mathbb{R}^p \), where \( \varsigma(\kappa \vartheta, A) \) is the normalizing constant in a Fisher-Bingham density.

2. The Fisher-Gaussian distribution (Mukhopadhyay et al., 2020) is a special case of the Ellipsoid-Gaussian distribution with \( \Lambda = r I_p \) and \( \Sigma = \sigma^2 I_p \).

3. The marginal distribution of any sub-vector of \( x \) also follows an Ellipsoid-Gaussian distribution. Specifically, suppose the index set of the random vector is \( I \subset [p] \) and let subscript \( I \)
represents elements from the rows of a matrix or elements from a vector, whose index is in set $I$. Then $x_I$ follows a Ellipsoid-Gaussian with density

$$f_{EG}(x_I) = \frac{C_k(\tau)}{(2\pi)^{\frac{|I|}{2}} \prod_{i \in I} \sigma_i} \exp \left\{-\frac{1}{2} (x_I - c_I)^T \Sigma_I^{-1} (x_I - c_I) \right\} \varsigma \left\{ \tau \mu + \Lambda_I^T \Sigma_I^{-1} (x_I - c_I), \frac{\Lambda_I^T \Sigma_I^{-1} \Lambda_I}{2} \right\},$$

where $\Sigma_I = \text{diag}(\sigma_j, j \in I)$.

4. The expectation of $x$ is $c + \rho_k(\tau) \Lambda \mu$, where $\rho_k(\tau) = I_{k/2}(\tau)/I_{k/2-1}(\tau)$.

5. The covariance of $x$ is $\rho_k(\tau)/\tau \Lambda \Lambda^T + \left\{ 1 - \frac{k}{2} \rho_k(\tau) - \rho_k^2(\tau) \right\} \Lambda \mu (\Lambda \mu)^T + \Sigma$.

This Proposition has several interesting ramifications. The term $\varsigma$ is not part of the normalising constant as it depends on the random variable $x$; this leads to some challenges in model fitting. The Fisher-Gaussian distribution in Mukhopadhyay et al. (2020) is a special case, which corresponds to the Ellipsoid-Gaussian with no dimension reduction ($p = k$), spherical Gaussian noise, and $\Lambda$ being a scalar multiple of the identity matrix. In addition, unlike the Gaussian linear factor model, $c$ is no longer the expectation of $x$, suggesting that we cannot simply center the data prior to analysis and remove $c$.

### 3.3.1 Identifiability of the model parameters

An Ellipsoid-Gaussian distribution contains parameters $\{c, \Lambda, \mu, \tau, \Sigma\}$ of total dimension $(p + 1)(k + 1) + p$. In this section, we show that some of these parameters are not identifiable.

**Proposition 7.** In model (3.2), $\Lambda$ is only identifiable up to orthogonal transformation.

Figure 3.5 gives a visual representation of this proposition, which shows that the nonidentifiability of $\Lambda$ is due to a transformation property of the von Mises-Fisher distribution. Model (3.1) shares this property, due to rotational invariance of the distribution of the latent factors $N(0, I_k)$, with the image of $\eta$ under an orthogonal transformation $\Gamma$ still being $N(0, I_k)$. In (3.2), however, an orthogonal transformation $\Gamma$ of $\eta$ results in a different von Mises-Fisher distribution, having the
same concentration parameter $\tau$ but with a different mean vector $\Gamma \mu$. Therefore, a rotation of $\eta$ necessarily results in a rotation in $\Lambda$ as well. The following proposition provides further insights.

**Proposition 8.** The moment generating function of the Ellipsoid-Gaussian distribution is

$$M_{EG}(t) = \exp \left( t \Gamma^T c + \frac{1}{2} t \Sigma t \right) \frac{C_k(\tau)}{C_k(\|\Lambda^T t + \tau \mu\|)}.$$

This proposition suggests a different parameterization. Let

$$\|\Lambda^T t + \tau \mu\| = \sqrt{t^T \Lambda \Lambda^T t + 2 \tau t^T \Lambda \mu + \tau^2}.$$

Hence, the moment generating function depends on $\Lambda$ and $\mu$ only through $\Lambda \Lambda^T$ and $\Lambda \mu$. By Proposition 6, $\Lambda \mu$ is the mean direction if the underlying ellipsoid is centered at the origin. The symmetric matrix $\Lambda \Lambda^T$ with spectral decomposition $US^2U^T$ defines a $k$-dimensional ellipsoid consisting of points satisfying $x^T US^{-2} U^T x = 1$. Instead of using $\Lambda$ and $\mu$, we can view the distribution as being parameterized by the supporting ellipsoid and the mean direction $\Lambda \mu$.

### 3.3.2 Limiting behavior

In this section, we show that the Gaussian linear factor model is a limiting case of the Ellipsoid-Gaussian distribution. The following Lemma states that, as the concentration of a von Mises-Fisher
distribution increases, the distribution approaches a degenerate Gaussian; a visualization of this lemma is included in the Supplementary Material along with its proof.

Lemma 25. Assume that \( \eta \sim vMF(\mu, \tau) \). As \( \tau \to \infty \), the quantity \( \tau^{1/2}(\eta - \mu) \) converges in distribution to a multivariate Gaussian with mean 0 and covariance matrix \( I_k - \mu\mu^T \), which is supported on the hyperplane perpendicular to \( \mu \).

Using Lemma 25, we are able to show that as \( \tau \) goes to infinity, (3.2) reduces to a Gaussian linear factor model with \((k - 1)\)-dimensional latent factors in \( \mathbb{R}^{k-1} \).

Proposition 9. Suppose we have a hyper-ellipsoid with axes represented by unit vectors \( u_1, \ldots, u_k \) and semi-axis lengths \( s_1 \geq s_2 \cdots \geq s_k \geq 0 \), with \( U = [u_1 \ldots u_k] \) and \( S = \text{diag}(s_1, \ldots, s_k) \). Suppose \( x_\tau \) follows (3.2), with \( \Lambda(\tau) = US(\tau) \), where \( S(\tau) = \sqrt{\tau}\text{diag}(s_1, \ldots, s_{k-1}, 0) \) and latent factors \( \eta_\tau \sim vMF(e_k, \tau) \) around \( \mu = e_k \). Then \( x_\tau \) converges in distribution as \( \tau \to \infty \) to a Gaussian latent factor model with \( k - 1 \) latent factors and loadings matrix \( U_{-k}S_{-k} \) where \( U_{-k} = [u_1 \ldots u_{k-1}] \) and \( S_{-k} = \text{diag}(s_1, \ldots, s_{k-1}) \), namely \( x_\infty = c + \tilde{\Lambda}\tilde{\eta} + \epsilon \), where \( \tilde{\eta} \sim N(0, I_{k-1}) \), \( \tilde{\Lambda} = U_{-k}S_{-k} \).

### 3.4 Posterior computation

Our extensive experiments have shown that standard Markov chain Monte Carlo algorithms, such as the Gibbs sampler and Hamiltonian Monte Carlo, fail to perform adequately for posterior sampling in models involving Ellipsoid-Gaussian likelihoods. In this section, we describe the computational challenges and motivate the algorithm we choose—a hybrid of geodesic stochastic Nosé-Hoover thermostat (Liu et al., 2016) and adaptive Metropolis (Vihola, 2012). The detailed sampling procedures, including an extensive discussion of prior specification, gradient computation and an outline of the algorithm, can be found in the Supplementary Material.

A brief comment is that samplers which rely on instantiating the latent variables \( \{\eta_i\} \) tend to be subject to poor mixing; this includes Gibbs samplers. This is due to well known problems with Gibbs sampling in latent factor models, motivating pseudo marginal algorithms (Andrieu and
Roberts, 2009; Beaumont, 2003; Vihola, 2012). We can bypass the need for pseudo marginal algorithms, which have also had poor performance in our experiments, due to the availability of a closed form likelihood marginalizing out the latent variables.

Marginalizing out the latent factors and using the resulting Ellipsoid-Gaussian likelihood in posterior sampling brings challenges. First, the term $\xi\{\tau\mu + \Lambda^T\Sigma^{-1}(x - c), \Lambda^T\Sigma^{-1}\Lambda/2\}$ needs to be approximated (Kume and Wood, 2005), but repeatedly applying such approximations becomes slow. Second, the gradient tends to change drastically with small changes in parameters, making it challenging to define efficient proposals. One failed example is the Barker proposal (Livingstone and Zanella, 2020). Even when proposal values are somewhat close to the current samples, the drastic changes in the gradient result in acceptance probability close to zero; to improve acceptance the algorithm adapts to propose tiny changes. The recently proposed transport Markov chain Monte Carlo (Duan, 2021) failed for similar reasons. Third, Gibbs-type updates lead to poor mixing, even with the latent factors marginalized out. For example, when we condition on the center $c$ and the shape of the ellipsoid as determined by $\Lambda$, there is often not much uncertainty in $\mu$, a parameter related to the mean direction of the data. As a result the chain tends to get stuck in a local mode. Fourth, $\mu$ is constrained to be on a sphere. A typical algorithm would require transforming $\mu$ to an unconstrained space $\mathbb{R}^k$. Specifically, to update $\mu$, we update an unconstrained vector $\tilde{\mu}$ and map it back to $\mu$ with $\tilde{\mu}/\|\tilde{\mu}\|_2$. $\tilde{\mu}$, unfortunately, tends to move in the direction close to being perpendicular to the unit sphere centered at the origin, which translates into minimal movement in $\mu$, as our experiments show. In addition, the update of the loadings matrix is also not very efficient, even with the use of popular shrinkage priors, such as the Dirichlet-Laplace (Bhattacharya et al., 2015). This makes us wonder whether directly updating the axes directions and lengths can lead to faster convergence since the dimension of the parameter space would be drastically reduced.

A possible remedy to the first issue is to use stochastic gradient algorithms; a small subset of the data are involved in a given iteration, leading to dramatic speedup. By relying on a small step size, one can avoid the need for a Metropolis-Hastings correction (Ma et al., 2015), which circumvents the second issue. Also, by updating all the parameters in a single block, we address the third challenge. Our solution to the fourth problem is inspired by sampling schemes designed for
distributions on manifolds embedded in Euclidean space with a known geodesic flow; see Byrne and Girolami (2013) and Liu et al. (2016) for more details.

Based on the above considerations, and on thorough experiments trying different algorithms, we use geodesic stochastic gradient Nosé-Hoover thermostats (Liu et al., 2016); we choose this dynamics because of the ease of tuning and robust performance. One weakness, however, is that all parameters share a single step size but the variances of their associated gradient are on vastly different scales. A discussion of our failed attempts to resolve this issue is included in the Supplementary Material. Our solution is to add an additional transition kernel for the parameters that would benefit from a larger step size, such as \( \Sigma \) and \( \tau \); robust adaptive Metropolis (Vihola, 2012) provides an effective way to define this transition kernel.

### 3.5 Simulation studies

#### 3.5.1 Experiment setups

We compare to infinite mixtures of infinite factor analyzers (Murphy et al., 2020), abbreviated as mixtures of factor analyzers, and infinite Gaussian linear factor models to illustrate our ability to characterize multivariate distributions with curved dependence. We use the R package IMIFA (Murphy et al., 2021) for mixtures of factor analyzers and infinitefactor (Poworoznek, 2020) for Gaussian linear factor models. Both packages use Markov chain Monte Carlo.

We generate 1,000 data samples from each of the models specified in Section 3.5.2 and then compare performance in two different ways. First, we evaluate the log posterior predictive density on the test sets, which are set to be 5\%, 10\%, 15\%, 20\%, 30\%, ..., 90\% of the simulated data samples. The estimated log posterior predictive density is

\[
\sum_{i \in \text{test set}} \frac{1}{M} \sum_{m=1}^{M} \log q(x_i; \theta^{(m)}),
\]

where \( q(x_i; \theta^{(m)}) \) is the likelihood of \( x_i \) evaluated at the \( m \)th draw \( \theta^{(m)} \) from the posterior. Second, we randomly hold out 5\% to 30\% of the data entries and compare performance in predicting the hold-out sets using predictive MSE. The test sets and the hold-out sets differ in that the former corresponds to entire entries of certain observations and the latter corresponds to partial entries only.
As Ellipsoid-Gaussian, Gaussian linear factor models and mixtures of factor analyzers are all factor models, we require specification of the number of latent factors. For Ellipsoid-Gaussian, the number of latent factors is determined by WAIC (Watanabe, 2013). For mixtures of factor analyzers and Gaussian linear factor models, the number of latent factors is automatically determined in the packages we use. We use default initialization of all three methods; the details of initialization for Ellipsoid-Gaussian can be found in the Supplementary Material. As Ellipsoid-Gaussian uses a stochastic gradient Monte Carlo algorithm, specification of a step size parameter is required. As is common practice for these algorithms, we choose a suitable step size from \{5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}\} in a tuning phase.

3.5.2 Simulated data sets

In the simulation experiments, we generate three data sets from Ellipsoid-Gaussian distributions, two with curvature and one without, and one data set from the Gaussian linear factor model. We also generate data from the hybrid Rosenbrock (Pagani et al., 2021), which was recently introduced as a benchmark to test the performance of Markov chain Monte Carlo on distributions with a curved and narrow shape. The density of a hybrid Rosenbrock random variable \( x = (x_1, \ldots, x_{n_2,n_1}) \) is

\[
\exp \left\{ -a_{ro}(x_1 - \nu)^2 - \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} b_{ji} (x_{ji} - x_{j,i-1})^2 \right\},
\]

(3.3)

where \( a_{ro}, b_{ji} \in \mathbb{R}^+ \). The settings of the simulated data sets are as follows:

very curved in \( \mathbb{R}^{10} \): \( x_i = \Lambda \eta_i + \epsilon_i, \quad \eta_i \sim \text{vMF}(\mu, 3), \quad \epsilon_i \sim \text{N}(0, 0.01I_{10}), \quad k = 3 \).

shell shape in \( \mathbb{R}^{3} \): \( x_i = \Lambda \eta_i + \epsilon_i, \quad \eta_i \sim \text{vMF}(\mu, 5), \quad \epsilon_i \sim \text{N}(0, 0.01I_3), \quad k = 3 \).

approx. Gaussian in \( \mathbb{R}^{6} \): \( x_i = \Lambda \eta_i + \epsilon_i, \quad \eta_i \sim \text{vMF}(\mu, 30), \quad \epsilon_i \sim \text{N}(0, 0.4I_6), \quad k = 2 \).

Gaussian linear factor in \( \mathbb{R}^{6} \): \( x_i = \Lambda \eta_i + \epsilon_i, \quad \eta_i \sim \text{N}(0, I_3), \quad \epsilon \sim \text{N}(0, 0.01I_6), \quad k = 3 \).

hybrid Rosenbrock in \( \mathbb{R}^{3} \): \( a_{ro} = 1, \quad b = (30, 1)^T, \quad \nu = 0 \) in Equ. 3.3.
3.5.3 Simulation results

In these examples, mixtures of factor analyzers tend to perform comparably to Ellipsoid-Gaussian when the training set size is large, and the Gaussian linear factor model tends to perform the worst in the presence of curvature, as evidenced by the out-of-sample log posterior predictive density in Fig. 3.6. Other supporting figures can be found in the Supplementary Material.

![Log posterior predictive density vs training set size](image)

Figure 3.6: The log posterior predictive density of the test sets evaluated under each model as a function of the test set sizes, with red, blue and green corresponding to Ellipsoid-Gaussian, Gaussian linear factor models and mixtures of factor analyzers, respectively.

One interesting example is the very curved data in $\mathbb{R}^{10}$. Figure 3.6 shows that mixtures of factor analyzers are only able to perform comparably to the Gaussian linear factor model even when the training set size is large. For the examples Gaussian linear factor in $\mathbb{R}^6$ and approx. Gaussian in $\mathbb{R}^6$, where the data can be approximated by a Gaussian distribution, the three methods perform equally well, though mixtures of factor analyzers experience a terrible run when the training set size is 500; see Fig. 3.6. As the training sample size decreases, the performance of mixtures of factor analyzers tends to be worse than that of Ellipsoid-Gaussian.

While the Gaussian linear factor model fails to capture curvature in the data sets (see Fig. 3.7c), its performance is surprisingly good in terms of MSE, which highlights limitations of the MSE at capturing departures from Gaussianity. In addition, the model tends to use more latent factors than
the number of variables in the data, suggesting a lack of dimension reduction and interpretability. As expected, mixtures of factor analyzers also use too many components leading to spurious clusters, which is especially true when the model is fitted to small sample size data with missing entries; see the visualisation of the posterior predictive distribution in Fig. 3.7b.

Figure 3.7: The scatter plot matrices of the posterior predictive (green) juxtaposed with the original data (red) for each method, where the models were fitted to the very curved data set in $\mathbb{R}^{10}$ with 1000 observations and 20% random missingness (top row) and the hybrid Rosenbrock data in $\mathbb{R}^{3}$ with 300 observations (bottom row) respectively. Letters p and o stand for posterior predictive and original data respectively.
3.6 Real data applications

3.6.1 Horse mussel data

The horse mussel data can be found in the R package dr (Weisberg, 2002), and contain measurements on 201 horse mussels taken from five sites in New Zealand in December 1984 (Camden, 1989). The measurements include shell width $W$, length $L$ and height $H$, each in millimeters, and shell mass $S$ and muscle mass $M$, each in grams. An interest of the study lies in predicting the muscle mass, the edible portion of the mussel, using other variables. The scatter plot matrix in Fig. 3.2b shows that the variables have curved relationships. We standardize the data such that each variable has mean zero and variance one. We choose step size $\epsilon = 5 \times 10^{-4}$ and $k = 3$ for Ellipsoid-Gaussian.

Applying cross validation in which we hold out up to 70% of the data, we find that Gaussian linear factor models are not competitive while Ellipsoid-Gaussian and mixtures of factor analyzers have similar fit out of sample. However, mixtures of factor analyzers are complex to interpret, in using three mixture components with each having up to four latent factors. This also leads to spurious clusters or outliers in the posterior predictive distribution; see Fig. 3.9b. In contrast, our Ellipsoid-Gaussian model has a linear factor structure simplifying interpretation. To illustrate this, we estimate the factor loadings structure by applying the MatchAlign algorithm (Poworoznek et al., 2021) to resolve rotational ambiguity and column label switching in $\Lambda$. The posterior mean of the resulting samples is visualized in Fig. 3.8a, where we see the muscle mass largely depends on a latent factor representing overall shell size. Figure 3.8b shows the relationship between the posterior mean of the muscle mass and the shell width holding the other covariates at their sample mean level: the muscle mass appears to be roughly constant when the shell length is less than 250mm and then increases linearly as shell length increases.

3.6.2 Air quality data

Urban atmospheric pollutants are believed to have contributed to an increasing incidence of respiratory illnesses. To monitor urban air quality, a multi-sensor device was placed in a polluted area in
Figure 3.9a illustrates the posterior predictive distribution produced by the Ellipsoid-Gaussian,
Figure 3.9: The scatter plot matrices of the posterior predictive (green) juxtaposed with the original data (red); the model was fitted to data sets with 20% in the test sets. Letters p and o stand for posterior predictive and original data respectively.

with the fitted marginal distributions matching those of the original data and capturing the curvature. The posterior predictive distribution produced by Gaussian linear factor models does not match the curvature in the data while that by mixtures of factor analyzers contain outliers. Gaussian linear factor model underfit, while mixtures of factor analyzers produce an overly complex fit with twenty-one components, each with up to four latent factors. Hence, we focus on the Ellipsoid-Gaussian as a good compromise between fit and interpretability.

Figure 3.8a visualizes the posterior mean of the post-processed loadings matrix, suggesting the weighted average of $O_3$ and CO concentration as one of the important latent factors. Figure 3.8b shows the relationship between the expected benzene concentration and the CO concentration hold-
ing all other variables at their sample mean level; specifically, the benzene level tends to decrease as CO increases when the concentration of CO is less than $1000 \text{ mg/m}^3$, but largely remains stable as the CO level continues to increase.

### 3.7 Discussion

In this article, we propose a simple and flexible von-Mises Fisher linear factor model to capture curved dependence in data, leading to a new class of Ellipsoid-Gaussian multivariate distributions. Simulating from the model results in points distributed about an ellipsoid, which is flexible enough to match curvature in many datasets. The use of a single factor loadings matrix facilitates dimension reduction and simple interpretation. In contrast, Gaussian linear factor models fail to characterize curvature and mixtures of factor analyzers are highly complex and can overfit the data, leading to spurious clusters and outliers in the posterior predictive.

In the absence of curved dependence in data, the model behaves like the routinely used Gaussian linear factor model, as we demonstrate both theoretically and empirically. We show how to marginalize out the latent factors and derive the density with respect to the Lebesgue measure. We also characterize various appealing properties of the distribution, such as that the marginal distribution of any sub-vector still follows an Ellipsoid-Gaussian.

There are a number of directions that would be interesting to pursue in future work. First, it is conceptually appealing to place shrinkage priors on the loadings, such as in Kowal (2021). However, this leads to challenges in computation. Currently, we sample the axes directions $U$ (i.e. the left singular vectors of $\Lambda$) and the axes lengths (i.e. the singular values of $\Lambda$). It is appealing to develop shrinkage priors for the axis directions to facilitate high-dimensional inferences, but the directions are constrained to the Stiefel manifold, making it unclear how to define appropriate priors. Second, efficiency of fitting Ellipsoid-Gaussian models is constrained by computation for approximating the pseudo-normalising constant. A faster but still reliable method for approximating this term would be helpful.
In this section, we provide proofs for Propositions 6–9 and Lemma 25. We discuss prior specification for the Ellipsoid-Gaussian distribution, followed by gradient computation, sampling procedures, initialization of the sampler and imputation of latent factors. In the last part, we include additional results for the simulation study and real data applications.

.1 Proofs of propositions and lemmas

.1.1 Proof of Proposition 6

Proof. 1. Let \( w = \Lambda \eta \) and \( \phi \) denote the pdf of \( \epsilon \). So \( x = c + w + \epsilon \). We claim the density of \( x \) is

\[
\text{f}_{\text{ellipsoid Gaussian}}(x) = \mathbb{E}_w [\phi(x - c - w)],
\]

(4)

with respect to the Lebesgue measure in \( \mathbb{R}^p \). Equation (4) can be simplified to be

\[
\mathbb{E}_\eta [\phi(x - c - \Lambda \eta)]
\]

\[
= \int_{S_{k-1}} \phi(x - c - \Lambda \eta) f_{\text{vMF}}(\eta) S^{k-1}(d\eta)
\]

\[
= \int_{S_{k-1}} \left[ (2\pi)^{-\frac{p}{2}} \prod_{i=1}^{p} \frac{1}{\sigma_i} \exp \left\{ -\frac{1}{2} (x - c - \Lambda \eta)^T \Sigma^{-1} (x - c - \Lambda \eta) \right\} \right] \times
\]

\[
\{C_k(\tau) \exp(\tau \eta^T \mu)\} S^{k-1}(d\eta)
\]

(5)

\[
= \frac{C_k(\tau)}{(2\pi)^{\frac{p}{2}} \prod_{i=1}^{p} \sigma_i} \exp \left\{ -\frac{1}{2} (x - c)^T \Sigma^{-1} (x - c) \right\} \times
\]

\[
\left[ \int_{S_{k-1}} \exp \left\{ -\frac{1}{2} \eta^T (\Lambda^T \Sigma^{-1} \Lambda) \eta + (\tau \mu^T + (x - c)^T \Sigma^{-1} \Lambda) \eta \right\} S^{k-1}(d\eta) \right] \right)^{\kappa/2}
\]

(6)

\[
= \frac{C_k(\tau)}{(2\pi)^{\frac{p}{2}} \prod_{i=1}^{p} \sigma_i} \exp \left\{ -\frac{1}{2} (x - c)^T \Sigma^{-1} (x - c) \right\}
\]

\[
\zeta \left\{ \tau \mu + \Lambda^T \Sigma^{-1} (x - c), \frac{\Lambda^T \Sigma^{-1} \Lambda}{2} \right\},
\]

where * is the kernel of the Fisher-Bingham distribution with parameter \( \kappa = \|\tau \mu + \Lambda^T \Sigma^{-1} (x - c)\|, \psi = \{\tau \mu + \Lambda^T \Sigma^{-1} (x - c)\}/\kappa \) and \( A = \Lambda^T \Sigma^{-1} \Lambda/2 \). Now we show that the density
of $x$ is

$$f_{\text{ellipsoid Gaussian}}(x) = \mathbb{E}_w \{ \phi(x - c - w) \}.$$  

Recall that if $\mathcal{U}, \mathcal{V}$ are finite Borel measures such that $\mathcal{V}$ has a density $g$ with respect to Lebesgue measure, then the convolution $\mathcal{U} * \mathcal{V}$ has density $g * \mathcal{U}$, as defined by

$$g * \mathcal{U}(x) = \int g(x - z)d\mathcal{U}(z). \quad (7)$$

In the special case where $\mathcal{U}$ and $\mathcal{V}$ are both probability measures, the convolution $\mathcal{U} * \mathcal{V}$ is the distribution of $Y + Z$, where $Y, Z$ are independent random variables with marginal distributions $\mathcal{U}$ and $\mathcal{V}$. Equation (7) can be rewritten as:

$$g * \mathcal{U}(x) = \int g(x - z)d\mathcal{U}(z) = \mathbb{E}g(x - Y),$$

where $x = c + w + \epsilon$ and $\epsilon$ has a density $\phi$ with respect to Lebesgue measure on $\mathbb{R}^p$. Hence the density of $x$ is

$$f_{\text{ellipsoid Gaussian}}(x) = \mathbb{E}_w \{ \phi(x - c - w) \}.$$  

2. With $\Sigma = \sigma^2 I_p$, $\Lambda = rI_p$, and $k = p$, equation (6) can be simplified to

$$\int_{S_{k-1}} \exp \left[ -\frac{1}{2} \eta^T \left( \Lambda^T \Sigma^{-1} \Lambda \right) \eta + \left\{ \tau \mu^T + (x - c)^T \Sigma^{-1} \Lambda \right\} \eta \right] \delta^{k-1}(d\eta)$$

$$= \int_{S_{k-1}} \exp \left[ -\frac{r^2}{2\sigma^2} \eta^T \eta + \left\{ \tau \mu^T + \frac{r}{\sigma^2} (x - c)^T \right\} \eta \right] \delta^{k-1}(d\eta)$$

$$= \exp \left( -\frac{r^2}{2\sigma^2} \right) \int_{S_{k-1}} \left\{ \tau \mu^T + \frac{r}{\sigma^2} (x - c)^T \right\} \eta \delta^{k-1}(d\eta)$$

$$= \exp \left( -\frac{r^2}{2\sigma^2} \right) \frac{1}{C_p(\|\tau \mu + \frac{r}{\sigma^2} (x - c)\|)},$$

where $**$ corresponds to a von-Mises Fisher distribution with concentration parameter $\tau^* = \|\tau \mu + \frac{r}{\sigma^2} (x - c)\|$ and mean direction $\mu^* = \tau \mu + \frac{r}{\sigma^2} (x - c)/\tau^*$. Hence, the part multiplied by $*$ in equation (6) can also simplified to be:

$$\frac{C_k(\tau)}{(2\pi\sigma^2)^{\frac{k}{2}}} \exp \left\{ -\frac{1}{2\sigma^2} (x - c)^T (x - c) \right\}.$$
The density of $x$ is thus

$$
\frac{C_k(\tau)}{(2\pi\sigma^2)^\frac{p}{2}} \exp \left[ -\frac{1}{2\sigma^2} \left\{ (x - c)^T (x - c) + r^2 \right\} \right],
$$

with respect to the Lebesgue measure on $\mathbb{R}^p$, which coincides with the density of the Fisher-Gaussian kernel.

3. Let $I$ denote a subset of the indices $\{1, \ldots, p\}$ of a vector $x$, and $x_I$ to be the vector of length $|I|$ containing the entries of $x$ corresponding to the elements of $I$; the matrix $\Lambda_I$ represents the matrix with $|I|$ rows which are equal to the rows of $\Lambda$ with indices in $I$. With this notation, we have $x_I = c_I + \Lambda_I \eta + \epsilon_I$, where $\epsilon_I \sim \mathcal{N}(0, \Sigma_I)$, $\Sigma_I = \text{diag}(\sigma_j, j \in I)$, $\eta \sim \mathcal{N}(\mu, \tau)$. This corresponds to the Ellipsoid-Gaussian distribution with density

$$
f_{EG}(x_I) = \frac{C_k(\tau)}{(2\pi)^\frac{|I|}{2} \prod_{i \in I} \sigma_i} \exp \left\{ -\frac{1}{2} (x_I - c_I)^T \Sigma_I^{-1} (x_I - c_I) \right\} \\
\varsigma \left\{ \tau \mu + \Lambda_I^T \Sigma_I^{-1} (x_I - c_I), \frac{1}{2} \Lambda_I^T \Sigma_I^{-1} \Lambda_I \right\},
$$

4.

$$
\mathbb{E}(x) = \mathbb{E}(c + w + \epsilon) \\
= c + \mathbb{E}(\Lambda \eta) + \mathbb{E}\epsilon \\
= c + \Lambda \mathbb{E}(\eta) \\
= c + \Lambda \rho_k(\tau) \mu,
$$

where $\rho_k(\tau) = I_{k/2}(\tau)/I_{k/2-1}(\tau)$ and $I_r(\cdot)$ is the modified Bessel function of the first kind and order $r$.

5. The covariance of $\eta$ is

$$
cov(\eta) = \rho_k(\tau) I_k + \left\{ 1 - \frac{k}{\tau} \rho_k(\tau) - \rho_k^2(\tau) \right\} \mu \mu^T,
$$
where \( \rho_k(\tau) \) has the same definition as in 4. Hence,

\[
\text{cov}(x) = \text{cov}(w + \epsilon) \\
= \text{cov}(w) + \text{cov}(\epsilon) \\
= \Lambda \text{cov}(\eta)\Lambda^T + \Sigma \\
= \frac{\rho_k(\tau)}{\tau} \Lambda\Lambda^T + \left\{1 - \frac{k}{\tau}\rho_k(\tau) - \rho_k^2(\tau)\right\} \Lambda\mu(\Lambda\mu)^T + \Sigma.
\]

\[
.1.2 \quad \text{Proof of Proposition 7}
\]

\textbf{Proof.} Let \( \Gamma \) be an arbitrary \( k \times k \) orthogonal rotation matrix satisfying \( \Gamma\Gamma^T = I_k \). Define \( \tilde{\eta} = \Gamma\eta \), \( \tilde{\Lambda} = \Lambda\Gamma^T \) and \( \tilde{\mu} = \Gamma\mu \). The von-Mises Fisher density satisfies

\[
f_{\text{vMF}}(\tilde{\eta}; \tilde{\mu}, \tau) = f_{\text{vMF}}(\eta; \mu, \tau).
\]

Hence \( x = c + \Lambda\eta + \epsilon = c + \tilde{\Lambda}\tilde{\eta} + \epsilon \), where \( \tilde{\eta} \sim \text{vMF}(\tilde{\mu}, \tau) \).

\[
.1.3 \quad \text{Proof of Proposition 8}
\]

\textbf{Lemma 26.} The moment generating function of the von-Mises Fisher distribution is

\[
M_{\text{vMF}}(t) = \frac{C_k(\tau)}{C_k(\|t + \tau\mu\|)}.
\]

\textbf{Proof.} The moment generating function is given by

\[
M_{\text{vMF}}(t) = \mathbb{E} \left( e^{t^T\eta} \right) \\
= \int C_k(\tau) \exp(t^T\eta + \tau\mu^T\eta) \, d\eta \\
= C_k(\tau) \int \exp(\tilde{\tau}\tilde{\mu}^T\eta) \, d\eta \\
= \frac{C_k(\tau)}{C_k(\tilde{\tau})},
\]

where \( \tilde{\tau} = \|t + \tau\mu\| \) and \( \tilde{\mu} = (t + \tau\mu)/\tilde{\tau} \).

145
Figure .10: As the concentration of a von-Mises Fisher distribution increases, the distribution approaches a degenerate Gaussian, supported on the hyperplane perpendicular to $\mu$.

of Proposition 8. An Ellipsoid-Gaussian random variable can be expressed in the form $x = c + \Lambda \eta + \epsilon$. Since the moment generating function of a sum of independent random variables is given by the product of moment generating functions, by Lemma 26

$$M_{\text{Ellipsoid-Gaussian}}(t) = M_{c + \Lambda \eta}(t) M_{\epsilon}(t)$$

$$= \exp(t^T c) M_{\text{vMF}}(\Lambda^T t) \exp \left( \frac{1}{2} t^T \Sigma t \right)$$

$$= \exp \left( t^T c + \frac{1}{2} t^T \Sigma t \right) \frac{C_k(\tau)}{C_k(\|\Lambda^T t + \tau \mu\|)}.$$

.1.4 Proof of Lemma 25

Kent (1978) provided a partial proof to this Lemma, and here we provide a complete proof, along with an intuitive explanation in Fig. .10.
**Proof.** The characteristic function of vMF \(k(\mu, \tau)\), denoted by \(\psi(t; \mu, \tau)\), is

\[
\psi(t; \mu, \tau) = C_k(\tau) \int_{S_k} \exp \left( it^T \eta + \tau \mu^T \eta \right) \nu_k(d\eta)
\]

\[
= \frac{C_k(\tau)}{C_k(||it + \tau \mu||)}.
\]

Based on Henkel's expansion of the modified Bessel function of the first kind,

\[
I_v(z) \sim (2\pi z)^{-1/2} e^z \{ 1 + O(|z|^{-1}) \}, \quad |\arg z| < \frac{1}{2} \pi - \delta
\]

as \( z \to \infty \) with fixed \( v \), where \( \delta \) is an arbitrary fixed positive number. Hence,

\[
\frac{C_k(z_1)}{C_k(z_2)} = \frac{I_{k/2-1}(z_2)}{I_{k/2-1}(z_1)} \left( \frac{z_1}{z_2} \right)^{k/2-1}
\]

\[
\sim \left( \frac{z_1}{z_2} \right)^{k/2} \exp(z_2 - z_1),
\]

as \( z_1, z_2 \to \infty \) with both \( |\arg z_1| \) and \( |\arg z_2| \) smaller than \( \frac{1}{2} \pi - \delta \). Letting \( z_1 = \tau \) and \( z_2 = ||it^{\frac{1}{2}} + \tau \mu|| \), \( z_2 \) can be rewritten as \( \left( \tau^2 + 2\tau^{3/2}i \sum_{j=1}^{k} t_j \mu_j - \tau \sum_{j=1}^{k} t_j^2 \right)^{\frac{1}{2}} \). As \( \tau \to \infty \),

\[
\left( \frac{z_1}{z_2} \right)^{k/2} \to 1,
\]

and by a Taylor series expansion,

\[
\left( \tau^2 + 2\tau^{3/2}i \sum_{j=1}^{k} t_j \mu_j - \tau \sum_{j=1}^{k} t_j^2 \right)^{\frac{1}{2}}
\]

\[
= \tau \left\{ 1 + i\tau^{-\frac{1}{2}} \sum_{j} t_j \mu_j - \frac{1}{2} \tau^{-1} \sum_{j} t_j^2 + \frac{1}{2} \tau^{-1} \left( \sum_{j} t_j \mu_j \right)^2 + O(\tau^{-3/2}) \right\},
\]

we have

\[
z_2 - z_1 = \tau \left\{ i\tau^{-\frac{1}{2}} \sum_{j} t_j \mu_j - \frac{1}{2} \tau^{-1} \sum_{j} t_j^2 + \frac{1}{2} \tau^{-1} \left( \sum_{j} t_j \mu_j \right)^2 + O(\tau^{-3/2}) \right\}.
\]

Hence,

\[
\exp(-i\tau^{\frac{1}{2}} t^T \mu)\psi(\tau^{\frac{1}{2}} t; \mu, \tau) = \frac{C_k(\tau)}{C_k(||i\tau^{\frac{1}{2}} t + \tau \mu||)}
\]

\[
\to \exp \left[ -\frac{1}{2} \left( \sum_{j} t_j^2 - \left( \sum_{j} t_j \mu_j \right)^2 \right) \right]
\]

\[
= \exp \left\{ -\frac{1}{2} t^T (I_k - \mu \mu^T) t \right\}.
\]
Hence, as $\tau \to \infty$, we have

$$\tau^{1/2}(\eta - \mu) \to N(0_k, I_k - \mu\mu^T).$$

To see this is a distribution on the hyperplane perpendicular to $\mu$, consider random variable $Z \sim N(k)(0_k, I_k)$, and note that

$$(I_k - \mu\mu^T)Z \sim N(k)(0_k, I_k - \mu\mu^T),$$

since $I_k - \mu\mu^T$ is a perpendicular projection operator onto the space perpendicular to $\mu$ and is thus idempotent. $\tau^{1/2}(\eta - \mu)$ converges in distribution to $Z$ as $\tau \to \infty$; by the continuous mapping theorem,

$$\{\tau^{1/2}(\eta - \mu)\}^T \mu \Rightarrow \{(I_k - \mu\mu^T)\}^T \mu = 0,$$

or equivalently,

$$\{\tau^{1/2}(\eta - \mu)\}^T \mu \Rightarrow 0,$$  \hspace{1cm} (8)

where symbol $\Rightarrow$ denotes convergence in distribution. \hfill \Box

### 1.5 Proof of Proposition 9

**Proof.** If $s_k = 0$ and $\mu = e_k$, then we have $S = S(I_k - \mu\mu^T)$. $$(I_k - \mu\mu^T)(\eta - \mu) = (I_k - \mu\mu^T)\eta$$

and from Lemma 25 we know that $\sqrt{\tau}(I_k - \mu\mu^T)\eta \Rightarrow N(0, I_k - \mu\mu^T)$ as $\tau \to \infty$. Let $z \sim N(0, I_k)$ be a vector of standard normal latent factors and $z_{-k} \sim N(0, I_{k-1})$ be the first $k - 1$ components. We can see that

$$\sqrt{\tau}S\eta = S\sqrt{\tau}(I_k - \mu\mu^T)\eta \Rightarrow S(I_k - \mu\mu^T)z = Sz$$

as $\tau \to \infty$, since $(I_k - \mu\mu^T)^2 = (I_k - \mu\mu^T)$. Be decomposing into block matrices, we find

$$\begin{bmatrix} U_{-k}u_k & S_{-k} & 0 & z_{-k} \\ 0^T & 0 & z_k & 0 \end{bmatrix} = \begin{bmatrix} U_{-k}u_k & S_{-k}z_{-k} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = U_{-k}S_{-k}z_{-k}. $$

148
Thus as $\tau \to \infty$ we have

$$x_{\tau} \sim c + U_{-k}S_{-k}z_{-k} + \epsilon,$$

which is a Gaussian linear factor model with $k-1$ latent factors and factor loadings matrix $U_{-k}S_{-k}$, as desired.

## 2 Prior specification

### 2.1 Inference on the concentration parameter

The Ellipsoid-Gaussian distribution has mass centered around a hyper-ellipsoid, but real data are often concentrated around a small portion of the hyper-ellipsoid. Hence, there can be multiple sets of parameters corresponding to different hyper-ellipsoids that match the curvature of the data equally well. For example, to obtain a different fit to the points in Fig. 3.3b, we can move the center of an ellipsoid farther from the data cloud, while increasing the axis lengths $s_j$s and the concentration parameter $\tau$ without significantly changing the distribution. Our strategy is to constrain $\tau$, so that if we generate data with this $\tau$ the resulting data will be neither too concentrated nor too diffuse around an ellipsoid. We do not want the resulting data to be too diffuse because real data almost never wrap entirely around an ellipsoid. To this end, we put a lower bound on $\tau$ in the prior to eliminate almost-uniform distributions on the ellipsoid. We can also put an upper bound on $\tau$ because while real data can be very concentrated around an ellipsoid, there are other parameters we can adjust accordingly to fit the data well in the presence of a constraint on $\tau$. In addition, for large $\tau_1$ and $\tau_2$ (greater than 100, for example), the distributions $\text{vMF}(\mu, \tau_1)$ and $\text{vMF}(\mu, \tau_2)$ are, for practical purposes, very similar. This can be formalized by examining the distances between the distributions for large $\tau$.

To determine an appropriate set of bounds, we make use of the fact that the level curves of the von-Mises Fisher distribution on the surface of a unit sphere are rings radiating out from the mean direction. Due to the rotational property of the von-Mises Fisher distribution, these rings, which correspond to the angle between some unit vectors and the mean, are determined only by $\tau$. To
Table 1: Reference bounds for $\tau$ under different latent dimensions $k$

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower bound $l_k$</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>upper bound $u_k$</td>
<td>14</td>
<td>22</td>
<td>30</td>
<td>35</td>
<td>39</td>
<td>47</td>
<td>51</td>
<td>56</td>
<td>62</td>
</tr>
</tbody>
</table>

find a practical constraint on $\tau$, we find the $\tau$ such that a certain percentage of the samples from the distribution are within a certain angle of $\mu$. For example, to find a reasonable lower bound, we require that at least 95% of the generated points have an angle of at most $\pi/2$ with $\mu$ so that they do not wrap around the sphere; to find an upper bound, we require that at most 95% of the points lie within $\pi/7$ of $\mu$ so that they are not overly concentrated on the sphere. A side benefit of this approach is that the resulting bounds automatically adjust for the latent dimension. The resulting bounds can be found in Table 1.

### 2.2 Inference on the factor loadings matrix

The non-identifiability of $\Lambda$, which encodes the axes directions and lengths of the underlying hyper-ellipsoid, poses challenges for computation. One natural idea is to put a strong shrinkage prior so that a narrower set of hyper-ellipsoids is favored. In fact, this kind of prior is frequently used in a variety of factor models, particularly for high-dimensional inference. Some examples of these shrinkage priors are the Dirichlet-Laplace prior of Bhattacharya et al. (2015) or the ordered spike-and-slab prior of Kowal (2021). Unfortunately, these priors do not assist with the inference for $\Lambda$ in our model. In fact, we notice slower convergence when $\Lambda$ is directly updated in comparison to when the axes directions (i.e. the left singular vectors of $\Lambda$) $U$ and the axes lengths (i.e. the singular values of $\Lambda$) $s_j$’s are directly updated in many cases. These priors may also come with some unintended side effects. Take the Dirichlet-Laplace prior as an example. When the penalization strength is too high, the prior strongly favors flat ellipsoids with axes parallel to the coordinate axes, which is often not consistent with the data.

One reason why directly updating the axes directions and lengths can lead to faster convergence
is that the dimension of the parameter space is drastically reduced; as the $k$ by $k$ right singular vectors $V$ of $\Lambda$ are not identifiable, we can fix this parameter at the identity matrix and only update the left singular vectors and singular values. As a default choice, we choose a uniform prior on the Stiefel manifold for $U$.

On the other hand, we favor smaller ellipsoids through an appropriate prior on the axes lengths $s_j$'s, consistent with our previous choice of constraints on $\tau$. As we constrain $\tau$ to be relatively small on the positive real line, we generally want small axis lengths, given that there is approximate non-identifiability when $\tau$ increases along with the axis lengths. As a prior for $s_j$ that maintains this behavior, we choose a truncated Gaussian that is constrained to be positive and has small variance.

### 2.3 Prior specification

The discussion in Sections 2.1 and 2.2 leads to the following prior:

\[
\begin{align*}
    c &\sim N(c_0, \sigma_c^2 I_p), \\
    \tau &\sim \text{Gamma}(a, b) \mathbb{1}(I_k, u_k), \\
    \mu &\sim \text{vMF}({\mu}_0, \tau_0), \\
    \sigma_j^2 &\sim N(0, \sigma_0^2) \mathbb{1}(0, \infty), \quad j = 1, \ldots, p \\
    U &\sim \text{Unif}(V_k(\mathbb{R}^p)), \\
    s_l &\sim N(0, \sigma_s^2), \quad l = 1, \ldots, k
\end{align*}
\]

where $V_k(\mathbb{R}^p) = \{ A \in \mathbb{R}^{p \times k} : A^T A = I_k \}$, and $I_k$ and $u_k$ are the lower and upper bounds for $\tau$ at latent dimension $k$. A set of reference bounds is provided in Table 1 and code to determine them is also provided in case a different constraint is preferred. We place a truncated Gaussian prior on the noise variance $\sigma_j$ because we favor a fit with small noise. For $\mu$, if a uniform prior on the unit sphere is desired, one can set $\tau_0 = 0$. For $\tau$, we set $a = 5, b = 1$ to favor a relatively small $\tau$ value, representing a fit with curvature. $c_0$ and $\mu_0$ are set to be the initialization of $c$ and $\mu$. Results are generally not sensitive to the choice of the prior parameters.


.3 Gradient computation

.3.1 Notation and parameter transformation

Let \( \theta = (c, \tau, \Lambda, \mu, \Sigma) \). Denote the prior for \( \theta \) by \( \pi_0(\theta) \) and the likelihood by \( \pi(x \mid \theta) \). Let precision \( 1/\sigma_j^2 \) be \( \omega_j^2 \). Given a data set \( \mathcal{D} = \{x_i\}_{i=1}^N \), a stochastic gradient Monte Carlo algorithm draws a random subset \( \mathcal{S} \) of \( \mathcal{D} \) to build the stochastic gradient

\[
\nabla_{\theta} G(\theta) := -\nabla_{\theta} \log \pi_0(\theta) - \frac{N}{{|\mathcal{S}|}} \sum_{x \in \mathcal{S}} \nabla_{\theta} \log \pi(x \mid \theta).
\]

Using a gradient-based method, parameters must be unconstrained on the embedded manifold for ease of updates; this means we must transform several of our parameters. We update \( \logit \{ (\tau - l_k)/(u_k - l_k) \} \), \( \log(1/\sigma_j^2) \) and \( \log(s_l) \) instead of \( \tau, \sigma_j^2 \) and \( s_l \) because the former parameters lie on the real line. That is, the set of parameters we are directly updating is

\[
\theta' := \{c, \logit \{ (\tau - l_k)/(u_k - l_k) \}, U, \{\log(s_l) : l = 1, \ldots, k\}, \mu, \{\log(\omega_j^2) : j = 1, \ldots, p\}\}.
\]

Refer to Table .2 for a summary.

In the next sections, we present the gradient associated with these parameters. We use subscripts \( j \cdot \) and \( \cdot j \) to represent the \( j \)th row and column of a matrix respectively. For ease of representation, we denote \( \tau \mu + \Lambda^T \Sigma^{-1} (x - c) \) and \( \text{vec}(\Lambda^T \Sigma^{-1} \Lambda / 2) \) by \( g_1(x) \) and \( g_2 \) respectively. In addition, any vector (including gradient) is represented as a column vector.

.3.2 Gradient from the likelihood

The gradient includes contributions from both the likelihood and the prior. For both parts, we need to first compute the gradient w.r.t \( \theta \) and then apply the chain rule of differentiation. To this end, we need the following gradient:

\[
\nabla_{\tau} = (u_k - l_k) \frac{\exp(\tau)}{\{1 + \exp(\tau)\}^2},
\]

\[
\nabla_{\sigma_j^2} \omega_j^2 = \omega_j^2,
\]

\[
\nabla_{\log(s_l)} \Lambda_j = s_l U_j,
\]

\[
\nabla_{U_j} \Lambda_j = (s_j, \ldots, s_j)^T.
\]
Table 2: The parameters involved in the sampler

<table>
<thead>
<tr>
<th>Original parameter $\theta$</th>
<th>Parameter being updated $\theta'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>$c$</td>
</tr>
<tr>
<td>$\tau$</td>
<td>$\tilde{\tau} := \text{logit} \left( \frac{\tau - l_k}{u_k - l_k} \right)$</td>
</tr>
<tr>
<td>$U$</td>
<td>$U$</td>
</tr>
<tr>
<td>$s_l$</td>
<td>$\tilde{s}_l := \log(s_l)$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>$\sigma_j^2$</td>
<td>$\tilde{\sigma}_j := \log(1/\sigma_j^2)$</td>
</tr>
</tbody>
</table>

Next, we discuss the gradient w.r.t $\theta$ from the likelihood. The log-likelihood of $x$ can be written as:

$$
\log \pi(x \mid \theta) = \log \pi(x \mid \theta')
= \log C_k(\tau) - \sum_{j=1}^{p} \log \sigma_j - \frac{1}{2} (x - c)^T \Sigma^{-1} (x - c) + \\
\log \xi \left\{ \tau \mu + \Lambda^T \Sigma^{-1} (x - c), \frac{\Lambda^T \Sigma^{-1} \Lambda}{2} \right\} + \text{const.}
$$

Since the pseudo-normalising constant $\xi$ can only be approximated, the associated gradient must also be approximated, for which we use Richardson’s extrapolation, an accurate numerical differentiation method, implemented by the R package numDeriv (Gilbert and Varadhan, 2019). We
first present gradient w.r.t a single observation $x$:

$$
\nabla_\tau \log \pi(x \mid \theta) = \nabla_\tau \log C_k(\tau) - \nabla_{g_1} \log \xi(g_1, g_2)^T \mu.
$$

$$
\nabla_\mu \log \pi(x \mid \theta) = \tau \nabla_{g_1} \log \xi(g_1, g_2).
$$

$$
\nabla_c \log \pi(x \mid \theta) = \Sigma^{-1} (x - c) - \Sigma^{-1} \nabla_{g_1} \log \xi(g_1, g_2).
$$

$$
\nabla_{\omega_j^2} \log \pi(x \mid \theta) = 0.5/\omega_j^2 - 0.5(x_j - c_j)^2 +

(x_j - c_j) \nabla_{g_1} \log \xi(g_1, g_2)^T \Lambda_j. + \nabla_{g_2} \log \xi(g_1, g_2)^T \mu.
$$

$$
\nabla_{\text{vec}(\Lambda)} \log \pi(x \mid \theta) = \{I_k \otimes \Sigma^{-1} (x - c)\} \nabla_{g_1} \log \xi(g_1, g_2) +

\{K(p, k)^T (\Sigma^{-1} \otimes I_k) + (I_k \otimes \Sigma^{-1} \Lambda)\} \nabla_{g_2} \log \xi(g_1, g_2),
$$

where $g_1 = g_1(x)$, $x_j$ is the $j$th variable of $x$ and $K(p, k)$ is a $pk$ by $pk$ commutative matrix such that $K(p, k) \text{vec}(\Lambda) = \text{vec}(\Lambda^T)$.

Updating with the subset $S$, the gradient becomes

$$
\nabla_\tau \sum_{x \in S} \log \pi(x \mid \theta) = |S| \nabla_\tau \log C_k(\tau) - \sum_{x \in S} \nabla_{g_1(x)} \log \xi(g_1(x), g_2)^T \mu.
$$

$$
\nabla_\mu \sum_{x \in S} \log \pi(x \mid \theta) = \tau \sum_{x \in S} \nabla_{g_1(x)} \log \xi(g_1(x), g_2).
$$

$$
\nabla_c \sum_{x \in S} \log \pi(x \mid \theta) = \sum_{x \in S} \Sigma^{-1} (x - c) - \Sigma^{-1} \nabla_{g_1(x)} \log \xi(g_1(x), g_2).
$$

$$
\nabla_{\omega_j^2} \sum_{x \in S} \log \pi(x \mid \theta) = 0.5|S|/\omega_j^2 - \sum_{x \in S} 0.5(x_j - c_j)^2 +

\sum_{x \in S} (x_j - c_j) \nabla_{g_1(x)} \log \xi(g_1(x), g_2)^T \Lambda_j. +

\sum_{x \in S} \nabla_{g_2} \log \xi(g_1(x), g_2)^T \mu.
$$

$$
\nabla_{\text{vec}(\Lambda)} \sum_{x \in S} \log \pi(x \mid \theta) = \sum_{x \in S} \{I_k \otimes \Sigma^{-1} (x - c)\} \nabla_{g_1(x)} \log \xi(g_1(x), g_2) +

\{K(p, k)^T (\Sigma^{-1} \otimes I_k) + (I_k \otimes \Sigma^{-1} \Lambda)\} \sum_{x \in S} \nabla_{g_2} \log \xi(g_1(x), g_2).
$$

Then apply the chain rule to find the gradient w.r.t $\theta'$.
.3.3 Gradient from the prior distribution

Next we discuss the gradient contributions from the prior distribution. The prior distribution is defined in Section .2.3 in terms of $\theta$. Since $\theta'$ is the parameter being updated, we need to find the prior distribution of $\theta'$, denoted by $\pi_0(\theta')$, from $\pi_0(\theta)$, which involves the Jacobian term $J(\theta') = |\partial \theta / \partial \theta'|$. Transformation of random variable gives $\log \pi_0(\theta') = \log \pi_0(\theta) + \log J(\theta')$. Therefore, we have the following relationship

$$\nabla_{\theta'} \log \pi_0(\theta') = (\partial \theta / \partial \theta') \nabla_{\theta} \{ \log \pi_0(\theta) + \log J(\theta') \}.$$  \hspace{1cm} (9)

To find (9), we first find $\nabla_{\theta} \log \pi_0(\theta)$:

$$\nabla_{c_j} \log \pi_0(\theta) = -(c_j - c_{0j}) / \sigma_c^2,$$

$$\nabla_{\mu} \log \pi_0(\theta) = \tau_0 \mu_0,$$

$$\nabla_{\tau} \log \pi_0(\theta) = (a - b) / \tau - b,$$

$$\nabla_{\sigma_j} \log \pi_0(\theta) = -\sigma_j^2 / \sigma_0^2,$$

$$\nabla_{\psi} \log \pi_0(\theta) = 0,$$

$$\nabla_{s_i} \log \pi_0(\theta) = -s_i / \sigma_s^2.$$  

Then we add a logged Jacobian term and apply the chain rule as defined in (9) to find the gradient w.r.t the parameter $\theta'$. Some details are as follows.

1. $\tau$: Recall $\tilde{\tau} = \logit \left( \frac{\tau - l_k}{u_k - l_k} \right)$, $\frac{d\tau}{d\tilde{\tau}} = (u_k - l_k) \frac{\exp(\tilde{\tau})}{(1 + \exp(\tilde{\tau}))^2}$. $\log \pi_0(\tilde{\tau}) = \log \pi_0(\tau) + \log J(\tilde{\tau})$, where $J(\tilde{\tau}) = |\frac{d\tau}{d\tilde{\tau}}|$. It is easy to find that $J(\tilde{\tau}) = \frac{d\tau}{d\tilde{\tau}}$, and $\nabla_{\tilde{\tau}} \log J(\tilde{\tau}) = \frac{1 - \exp(\tilde{\tau})}{1 + \exp(\tilde{\tau})}$. In addition, by chain rule, $\nabla_{\tilde{\tau}} \log \pi_0(\tau) = \nabla_{\tau} \log \pi_0(\tau) \frac{d\tau}{d\tilde{\tau}}$. Hence, $\nabla_{\tilde{\tau}} \log \pi_0(\theta') = \nabla_{\tilde{\tau}} \log \pi_0(\theta) + \nabla_{\tilde{\tau}} \log J(\tilde{\tau}) = \nabla_{\tau} \log \pi_0(\theta)(u_k - l_k) \frac{\exp(\tilde{\tau})}{(1 + \exp(\tilde{\tau}))^2} + \frac{1 - \exp(\tilde{\tau})}{1 + \exp(\tilde{\tau})}$.

2. $\omega_j^2$: Recall $\tilde{\sigma}_j = \log \omega_j^2 = -\log \sigma_j^2$. For ease of demonstration, we drop index $j$ for now. $\frac{d\sigma^2}{d\tilde{\sigma}} = -\exp(-\tilde{\sigma})$. $\log \pi_0(\tilde{\sigma}) = \log \pi_0(\sigma^2) + \log J(\tilde{\sigma})$, where $J(\tilde{\sigma}) = |\frac{d\sigma^2}{d\tilde{\sigma}}| = \exp(-\tilde{\sigma})$. In addition, by chain rule, $\nabla_{\tilde{\sigma}} \log \pi_0(\sigma^2) = \nabla_{\sigma^2} \log \pi_0(\sigma^2) \frac{d\sigma^2}{d\tilde{\sigma}}$. Hence, $\nabla_{\tilde{\sigma}} \log \pi_0(\theta') = \nabla_{\tilde{\sigma}} \log \pi_0(\theta) + \nabla_{\tilde{\sigma}} \log J(\tilde{\sigma}) = -\nabla_{\sigma^2} \log \pi_0(\theta) \exp(-\tilde{\sigma}) - 1$.  

155
To summarise, the resulting prior gradient is as follows:

\[ \nabla_{c_j} \log \pi_0(\theta') = -\frac{(c_j - c_0)}{\sigma_c^2}, \]
\[ \nabla_{\mu} \log \pi_0(\theta') = \tau_0 \mu_0, \]
\[ \nabla_{\tilde{\tau}} \log \pi_0(\theta') = \left\{ \frac{(a - b) + b}{\tau} \right\} (u_k - l_k) \frac{\exp(\tilde{\tau})}{1 + \exp(\tilde{\tau})^2} + \frac{1 - \exp(\tilde{\tau})}{1 + \exp(\tilde{\tau})}, \]
\[ \nabla_{\tilde{\sigma}_j} \log \pi_0(\theta') = -\exp(-2\tilde{\sigma}_j)/\sigma_0^2 - 1, \]
\[ \nabla_{U} \log \pi_0(\theta') = 0, \]
\[ \nabla_{\log(s_l)} \log \pi_0(\theta') = -\frac{s_l^2}{\sigma_s^2} - 1. \]

For each parameter, the gradient is a sum of the gradient from the likelihood and its prior distribution.

### 4 Sampling algorithm

#### 4.1 Sampling procedures

In this section, we outline the sampling procedures, which involve geodesic stochastic gradient Nosé-Hoover thermostat (Liu et al., 2016) and robust adaptive Metropolis algorithm (Vihola, 2012) at every iteration, in Algorithm 2. For notation brevity, the algorithms are abbreviated as gSGNHT and ramcmc respectively in Algorithm 2.

The dynamics of the geodesic algorithm is defined in Liu et al. (2016). We use a symmetric splitting integrator scheme, which leads to more accurate sample draws and a solution that depends on the Riemann metric tensor only through the geodesic flow. The sampling procedure is provided in Algorithm 1.

Good default choices of the tuning parameters are \( A = 0.1, \epsilon = 5 \times 10^{-4} \) for the geodesic algorithm and \( L = 10, \gamma = 2/3 \) and the coerced acceptance rate \( \alpha^* = 23.4\% \) for robust adaptive Metropolis algorithm.
Output: \( \{ \theta^{(t)} : t = 1, \ldots, T \} \)

1 Randomly initialize \( \theta^{(0)} \)

2 Sample \( v^* \sim N(0, I) \), project \( v^{(0)} \leftarrow \mathcal{P}(\theta^{(0)})v^* \) and set \( \xi^{(0)} \leftarrow C \)

3 for \( t = 1, \ldots, T \) do

   4 Sample a subset \( S \) for computing \( \nabla_{\theta'} \mathcal{G}_H(\theta') \).

   \((x_0, v_0, \xi_0) \leftarrow (x^{(t-1)}, v^{(t-1)}, \xi^{(t-1)})\)

   5 for \( l = 1, \ldots, L \) do

      6 A: Update \((\theta^*, v^*) \leftarrow (x_{l-1}, v_{l-1})\) by the geodesic flow for time step \( \epsilon/2 \),

      \[ \xi^* \leftarrow \xi_{l-1} + (\frac{1}{m}v_{l-1}^T v_{l-1} - 1)\epsilon/2 \]

      7 B: \( v^* \leftarrow \exp\{-\xi^*\frac{\xi_n}{2}\}v^* \)

   8 O: \( v^* \leftarrow v^* + \mathcal{P}(\theta^*) \cdot [ -\nabla_{\theta'} \mathcal{G}_H(\theta^*)\epsilon + N(0, (2A - \epsilon_n V(\theta^*))\epsilon_n] \)

   9 B: \( v^* \leftarrow \exp\{-\xi^*\frac{\xi_n}{2}\}v^* \)

   10 A: Update \((\theta_t, v_t) \leftarrow (\theta^*, v^*)\) by the geodesic flow for time step \( \epsilon/2 \),

      \[ \xi_t \leftarrow \xi^* + (\frac{1}{m}v^* v^* - 1)\frac{\xi}{2} \]

11 \((\theta^{(t)}, v^{(t)}, \xi^{(t)}) \leftarrow (\theta_L, v_L, \xi_L), \) and no Metropolis-Hasting correction step

**Algorithm 1:** Sampling steps of geodesic Nosé-Hoover thermostat (Liu et al., 2016)
Output: samples from the posterior distribution \( \{ \theta(t) : t = 1, \ldots, T \} \)

1. Randomly initialize \( \theta(0) \)

2. Sample \( v^* \sim N(0, I) \) and project \( v(0) \leftarrow A(\theta(0))v^* \). \( \xi(0) \leftarrow C \)

3. for \( t = 1, \ldots, T \) do
   - \( \triangleright gSGNHT: \) joint update of \( \theta \)
   - Follow the steps 4, 5, 6, 7, 8, 9, 10 and 11 of Algorithm 1
   - \( \triangleright \text{ramcmc: } \) joint update of \( \iota := (\tilde{\tau}, \tilde{\sigma}_1, \ldots, \tilde{\sigma}_p) \)

4. Compute \( \iota^* = \iota(t) + M_{t-1}W_t \), where \( W_t \) is a random vector from a standard multivariate Gaussian distribution

5. With probability \( \alpha_t := \min\{1, \pi(\iota^* | D)/\pi(\iota(t) | D)\} \), the proposal is accepted and \( \iota(t) = \iota^* \); otherwise, the proposal is rejected

6. Compute the lower-diagonal matrix \( M_t \) with positive diagonal elements satisfying the equation

\[
M_tM_T = M_{t-1}(I + t^{-\gamma}(\alpha_t - \alpha)^* W_t W_T)M_{t-1}^T,
\]

where \( \gamma \in (1/2, 1] \), and \( \alpha^* \) is the coerced acceptance rate

Algorithm 2: The posterior sampler for Ellipsoid-Gaussian
.4.2 Initialization

We use subscript $\text{init}$ to represent the initial values of the parameters. We initialize the chain with $\tau_{\text{init}} = 5$ and the noise variance at a small value with $\Sigma_{\text{init}} = \text{diag}(0.1, \ldots, 0.1)$ to facilitate identification of the underlying ellipsoid.

Let the principal components of the (centered) data to be $\mathbf{U}$. We initialize the axes directions $\mathbf{U}_{\text{init}}$ to be the first $k$ principal components ordered by the associated standard deviation in decreasing order. To find the center $c_{\text{init}}$, we first rotate the data so that the new coordinate axes are parallel to the principal components $\mathbf{U}$, then use the mid points of the ranges in each dimension as an initial estimate of the center, which is then transformed back to the Cartesian coordinate system to be $c_{\text{init}}$. To initialize the axes lengths, we first center the data by using the estimated center $c_{\text{init}}$ so that the data are centered around the origin, then map the data by $\mathbf{U}_{\text{init}}^T$ to the coordinate space of the first $k$ principal components, and calculate the average distances to the origin along each coordinate axis as the initial estimates of the axes lengths. To initialize $\mu$, we estimate $S_{\text{init}}\mu_{\text{init}}$ based on the first moment of the Ellipsoid-Gaussian distribution and invert the quantity by the initial value of $S$, where $S = \text{diag}(s_1, \ldots, s_k)$. Specifically, $S_{\text{init}}\mu_{\text{init}} = \mathbf{U}_{\text{init}}^T (\bar{x} - c_{\text{init}}) / \rho_k(\tau_{\text{init}})$. This can be solved for $\mu_{\text{init}}$, so we then set the initial value $\mu_{\text{init}}$ to be the normalised version of $\mathbf{U}_{\text{init}}^T (\bar{x} - c_{\text{init}}) / \rho_k(\tau_{\text{init}}) \odot s_{\text{init}}$, where $s = (s_1, \ldots, s_k)^T$.

.4.3 Imputation of the latent factors

While our posterior sampler relies on marginalizing out the latent factors, sampling the latent factors can be of interest. To impute the latent factors for observation $i$, we simply add an additional step of drawing from $\eta_i \mid x_i, \theta_i$ at every Markov chain Monte Carlo iteration. Although $\eta_i \mid x_i, z_i$ follows a Fisher-Bingham distribution and there are dedicated algorithms to sample from this distribution in the literature (Hoff, 2009; Kent et al., 2018), our experiments show poor performance for these algorithms. Instead, we take advantage of the fact that in our setting, the Fisher-Bingham tends to be highly concentrated, motivating a von-Mises Fisher with high concentration (e.g. 30) centered at the current value as the proposal distribution for a Metropolis-Hasting update.

Specifically, the full conditional of latent factor $\eta_i$ is Fisher-Bingham($\bar{\tau}_i, \bar{\mu}_i, \bar{A}_i$), where $\bar{\tau}_i =$
\[
\| \tau \mu + \Lambda^T \Sigma^{-1} (x_i - c) \|, \quad \bar{\mu}_i = \{ \tau \mu + \Lambda^T \Sigma^{-1} (x_i - c) \} / \bar{\tau}_i \text{ and } \bar{A}_i = \Lambda^T \Sigma^{-1} \Lambda / 2.
\]
In our experiments, we choose the concentration parameter to be 30 and perform the transition for 20 times at every iteration for improved convergence.

### 4.4 Other comments on the sampling algorithm

As we discussed in Section 1.5, geodesic stochastic gradient Nosé-Hoover thermostat suffers from the problem of using a single step size for all parameters, whose associated gradients are on vastly different scales. One intuitive solution is to use a preconditioner which guarantees a larger step size in directions with less curvature and a smaller step size in directions with more curvature. This strategy has been widely used in stochastic gradient descent algorithms and some gradient-based sampling algorithms such as Hamiltonian Monte Carlo. Unfortunately, the position where a preconditioner assumes is taken by the Riemann metric tensor \( G \). We can see this by understanding the algorithm dynamics in Liu et al. (2016).

An alternative solution is to add an additional transition kernel for parameters that would benefit from taking a larger step. Such parameters include \( \tau \) and \( \Sigma \), and the robust adaptive Metropolis algorithm (Vihola, 2012) provides an elegant recipe for this additional transition kernel.

### 5 Experiment results

#### 5.1 Simulation results

In this section, we provide additional results of the simulation study. This includes the out-of-sample log posterior predictive density (Fig. .11) and scatter plot matrices of the posterior predictive (Fig. .13) for the data sets not displayed in the paper.

Imputing the hold-out entries by mixtures of factor analyzers is extremely slow to the extent of being computationally intractable for moderate dimensional (e.g. the very curved example in \( R^{10} \)) or large (e.g. the air quality data with 9,000 observations) datasets. Recall the settings of the
Figure 11: The log posterior predictive density of the test sets evaluated under each model as a function of the test set sizes, with red, blue and green corresponding to Ellipsoid-Gaussian, Gaussian linear factor models and mixtures of factor analyzers, respectively. In Fig. 11b the three curves are exactly overlapping, indicating the performance of the three methods is comparable.

Simulations are as follows:

- Very curved in $\mathbb{R}^{10}$: $x_i = \Lambda \eta_i + \epsilon_i$, $\eta_i \sim \text{vMF}(\mu, 3)$, $\epsilon_i \sim \text{N}(0, 0.01I_{10})$, $k = 3$.
- Shell shape in $\mathbb{R}^3$: $x_i = \Lambda \eta_i + \epsilon_i$, $\eta_i \sim \text{vMF}(\mu, 5)$, $\epsilon_i \sim \text{N}(0, 0.01I_3)$, $k = 3$.
- Approx. Gaussian in $\mathbb{R}^6$: $x_i = \Lambda \eta_i + \epsilon_i$, $\eta_i \sim \text{vMF}(\mu, 30)$, $\epsilon_i \sim \text{N}(0, 0.4I_6)$, $k = 2$.
- Gaussian linear factor in $\mathbb{R}^6$: $x_i = \Lambda \eta_i + \epsilon_i$, $\eta_i \sim \text{N}(0, I_3)$, $\epsilon \sim \text{N}(0, 0.01I_6)$, $k = 3$.
- Hybrid Rosenbrock in $\mathbb{R}^3$: $a_{ro} = 1$, $b = (30, 1)^T$, $\nu = 0$ in Eqn. 3.3.

.5.2 Application results

In Fig. 14, we visualize the relationships between the muscle mass (resp. benzene concentration) and the remaining covariates not displayed in the paper.
Figure 12: The trace plot (top row) and the auto correlation function plot (bottom row) of some precision parameters for the simulated and real data sets.
Figure 13: The scatter plot matrices of the posterior predictive (green) juxtaposed with the original data (red) for each method, where the models were fitted to the shell shape in \( \mathbb{R}^3 \) with 400 observations (top row), approx.Gaussian data in \( \mathbb{R}^6 \) with 1000 observations and 30% random missingness (middle row) and data from Gaussian linear factor model in \( \mathbb{R}^6 \) with 1000 observations and 20% random missingness (bottom row). Letters p and o stand for posterior predictive and original data respectively.
Figure 14: The posterior mean (black) and the 95\% credible band (blue ribbon) of muscle mass (top) and the benzene level (bottom) as a function of some covariates while holding other covariates at their sample mean level based on the Ellipsoid-Gaussian fit.
Bibliography


Devroye, L., Mehrabian, A., and Reddad, T. (2018). The total variation distance between high-dimensional gaussians with the same mean.


173


177


181


Biography

Hanyu graduated from University of Hong Kong in 2014 with a Bachelor’s degree in actuarial science. With an interest in statistics, she later pursued a Master’s degree in statistical science at Duke University and was subsequently admitted as a PhD student to the same department. She is the winner of “the best talk in theory and methods” at the 2021 Bayesian Young Statisticians Meeting. She was also a finalist for 2021 Facebook Fellowship and was awarded Duke TRIPDS Fellowship in 2020.