Inference under Finite Mixture Models: Distributed Learning and Approximate Inference

by

Qiong Zhang

B.Sc., University of Science and Technology of China, 2015

M.Sc., University of British Columbia, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Statistics)

The University of British Columbia

(Vancouver)

April 2022

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Inference under Finite Mixture Models: Distributed Learning and Approximate Inference

submitted by **Qiong Zhang** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Statistics**.

Examining Committee:

Jiahua Chen, Professor, Department of Statistics, UBC
Supervisor

Trevor Campbell, Assistant Professor, Departmnet of Statistics, UBC
Supervisory Committee Member

Harry Joe, Professor, Department of Statistics, UBC
Supervisory Committee Member

Daniel J. McDonald, Associate Professor, Department of Statistics, UBC
University Examiner

Jun-cheng Wei, Professor, Department of Mathematics, UBC
University Examiner

Geoffrey McLachlan, Professor, School of Mathematics and Physics, The University of Queensland
External Examiner

Abstract

Finite mixture models are widely used to model data that exhibit heterogeneity. In machine learning, they are often used as probabilistic models for clustering analysis. In the application of finite mixtures to real datasets, the most fundamental task is to learn model parameters. In modern applications, datasets are often too large to be stored in a single facility and are distributed across data centres. To learn models on these distributed datasets, split-and-conquer (**SC!**) approaches are often used. SC approaches consist of two steps: they first locally learn one model per data centre and then send these local results to a central machine to be aggregated. Since the parameter spaces of mixtures are non-Euclidean, existing aggregation methods are not appropriate. We develop a novel computationally efficient aggregation approach for SC learning of finite mixtures. We show that the resulting estimator is root-n-consistent under some general conditions. Experiments show the proposed approach has comparable statistical performance with the global estimator based on the full dataset, if the latter is feasible. It also has better statistical and computational performance than some existing methods for learning mixtures on large datasets.

Finite mixtures are also widely used to approximate density functions of various shapes. When mixtures are used in graphical models to approximate density functions, the order of the mixture increases exponentially due to recursive procedures and the inference becomes intractable. One way to make the inference tractable is to use mixture reduction, that is, to approximate the mixture by one with a lower order. We propose a novel reduction approach by minimizing the Composite Transportation Divergence (CTD) between two mixtures. The optimization problem can be solved by a majorization minimization algorithm. We show

that many existing algorithms for reduction are special cases of our approach. This finding allows us to provide theoretical support for these existing algorithms. Our approach also allows flexible choices for cost functions in CTD. This flexibility allows our approach to have better performance than existing approaches.

We also discuss other related learning issues under finite mixtures.

Lay Summary

Consider a hypothetical example in marketing. Consumers at a supermarket chain may have different consumption patterns. To maximize the profit of the company, the chain may be interested in grouping consumers by common characteristics and priorities and designing more effective promotion strategies for each group. Consumer spending data collected from all consumers is a mixture of group-specific consumption styles, and statisticians can help the company to distinguish between different groups, identifying their respective proportions and specific consumption patterns.

We consider a scenario where consumer data are stored in multiple data centres. Each centre can only report its summary findings to the headquarters, and the headquarters want to combine these summaries to form a unified picture of customer groups and their consumption styles. This thesis develops statistical techniques to combine summary findings at headquarters and addresses broader issues.

Preface

This thesis was completed under the supervision of Professor Jiahua Chen. Chapter 3 is based on the manuscript Zhang and Chen (2021a) which is in press at "Advances and Innovations in Statistics and Data Science". Chapter 4 is based on the paper Zhang and Chen (2022) which is accepted by the *Journal of Machine Learning Research* and the conference paper Zhang and Chen (2021b) which is published at the *3rd International Conference on Statistics: Theory and Applications*. Chapter 5 is based on the manuscript Zhang et al. (2020). The ideas for Chapter 3 and Chapter 4 have been jointly developed by Qiong Zhang and Professor Jiahua Chen with all computational work conducted by Qiong Zhang as lead author. The ideas for Chapter 5 have been jointly developed by Qiong Zhang and coauthor Archer Gong Zhang with the majority of computational work and manuscript writing conducted by Qiong Zhang as lead author.

Table of Contents

List of Tables

List of Figures

List of Symbols and Notations

$:=$	Denoted as
$\odot$	The elementwise product between two vectors
$\otimes$	The Kronecker product between matrices
$\mathbb{1}(\cdot)$	Indicator function
$\mathbf{1}_n$	The vector of 1s with length n
δ_x	The dirac measure on a set A, $\delta_x(\mathbb{A}) = \mathbb{1}(x \in \mathbb{A})$
δ_{ij}	Kronecker delta function, the function is 1 if $i = j$, and 0 otherwise
Δ_K	The K simplex $\{(x_1, x_2, \ldots, x_{K+1}) : x_k \in [0, 1], \sum_{k=1}^{K+1} x_k = 1\}$
Θ^K	The Cartesian product $\Theta \times \Theta \times \cdots \times \Theta$ over K sets of Θ
$\phi(x; \mu, \Sigma)$	Probability density function of Gaussian distribution with mean μ and covariance Σ, the parameters are omitted when mean is zero and covariance is identity matrix
$\Phi(x; \mu, \Sigma)$	The cumulative distribution function of $\phi(x; \mu, \Sigma)$
$\|A\|_F$	The Frobenius norm $\|A\|_F = \sqrt{\text{tr}(A^\top A)}$ of matrix A
$A > 0$	The matrix A is positive definite
$A \geq B$	The matrix $A - B$ is semi-positive definite
$\mathbb{A} \backslash \mathbb{B}$	The set difference $\mathbb{A} \backslash \mathbb{B} = \{x : x \in \mathbb{A} \text{ and } x \notin \mathbb{B}\}$
$\text{card}(\mathbb{A})$	The cardinality of a set $\mathbb{A}$
$\det(A)$	The determinant of square matrix A
$\mathbb{E}$	Expectation of a random variable
$\langle f, g \rangle$	The function inner product $\langle f, g \rangle = \int f(x)g(x)\, dx$
$\mathbb{G}_K$	Space of mixing distributions with order up to K
$\mathbf{I}_d$	Identity matrix with dimension d

$[N]$	The index set $\{1, 2, \ldots, N\}$
$\binom{n}{k}$	The number of k combinations from a given set of n elements
$o(1)$	A quantity that converges to 0 almost surely
$\mathcal{O}(n)$	A quantity that denotes the computational cost bounded by Cn
$O_p(a_n)$	A quantity that is bounded by Ca_n for sufficiently large C with probability arbitrarily close to 1
$\mathbb{P}$	Probability of an event
$\mathcal{P}(\Theta)$	Space of probability measures on Θ equipped with some compatible σ-algebras
$\mathcal{P}_r(\Theta)$	Space of probability measures on Θ with finite rth moments
$\mathbb{R}$	Set of all real numbers
$\mathbb{R}_{<0}$	Set of all negative real numbers
$\mathbb{R}_{>0}$	Set of all positive real numbers
$\mathbb{R}_{+}$	Set of all non-negative real numbers
$\mathbb{S}_{+}^{d}$	Space of all $d \times d$ positive definite matrices
$\mathrm{tr}(A)$	The trace of matrix A
$\mathrm{Vec}(A)$	A vector formed by the entries of matrix A column-wise
$\|\boldsymbol{x}\|$	The Euclidean norm of vector $\boldsymbol{x}$

Glossary

ARI Adjusted Rand Index

BFGS Broyden-Fletcher-Goldfarb-Shanno

BP Belief Propagation

CDF Cumulative Distribution Function

CS Cauchy-Schwarz

CTD Composite Transportation Divergence

EM Expectation Maximization

GMM Gaussian Mixture Model

GMR Gaussian Mixture Reduction

HMM Hidden Markov Model

IID Independent and Identically Distributed

ISE Integrated Squared Error

KL Kullback-Leibler

KLA KL Averaging

LHS Left Hand Side

LL Log-likelihood

MAP Maximize a Posterior

MLE Maximum Likelihood Estimate

MM Majorization Maximization

MSE Mean Squared Error

MWDE Minimum Wasserstein Distance Estimator

OT Optimal Transport

PMLE penalized Maximum Likelihood Estimate

RHS Right Hand Side

Acknowledgments

First and foremost, I am grateful to my advisor, Jiahua Chen, for his guidance and support in the past seven years. He taught me to think critically and accurately. He also gave me the freedom to explore various research ideas and guidance when needed.

I would also like to thank my committees, Prof. Trevor Campbell and Prof. Harry Joe, for their detailed comments, insightful advice, and valuable discussions. I would also like to thank Prof. Geoffrey McLachlan who provided extensive feedback as the external examiner, Prof. Daniel J. McDonald and Prof. Jun-cheng Wei for serving as the university examiners. My sincere thanks also go to Dr. Peng Dai and Dr. Juwei Lu for offering me the summer internship opportunity. I thank the International Doctoral Fellowship program at UBC for funding my Ph.D. studies.

I want to thank all my collaborators, especially Dr. Bo Chang, Dr. Xin Ding, Hanwen Liang, and Dr. Archer Gong Zhang. They have provided me with much help, and their perspectives are a valuable source of scientific inspiration. I have learned a lot from them.

I want to thank other faculty members for their advice and support, especially Prof. Xiaoxiao Li, Prof. Melissa Lee, Prof. John Petkau, and Prof. Lang Wu. I also want to thank all staff members in our department, especially Binh Cong Dang, The Ha, Ali Hauschildt, Peggy Ng, and Andrea Sollberger, for assisting me and making the department a welcoming and friendly place.

I want to thank all my past and current fellow students, especially Jonathan Agyeman, Hung-li Chen, Derek Cho, Daniel Dinsdale, Xinzhe Dong, Xinyao Fan, Yidie Feng, Eric Fu, Julian Ho, Boyi Hu, Weining Hu, Xixi Hu, David Kepllinger, Xiaomeng Ju, Xinglong Li, Shenyi Pan, Elena Shchurenkova, Ning Shen, Sonja

Isberg Surjanovic, Joe Watson, Wayne Wang, Xinmiao Wang, Yijun Xie, David Xu, Qian Ye, Xiaoli Yu, Tingting Zhao, Yichen Zhao, and Mengling Zhou, for being around.

I thank my other friends, who have provided emotional support for me throughout the years, especially Yiyan He, Yiwei Hou, Mengqiao Li, Richard Schonberg, Bin Wang, Kuan Wang, Xiaotong Wang, Yunmiao Wang, and Chuanpin Yu.

Finally, I would like to thank my parents and brother for their love. They have sacrificed a lot raising me and supporting my career.

Dedication

This dissertation is dedicated to my beloved ones.
谨以此文献给所有我爱的人。

Chapter 1

Introduction

1.1 Finite Mixture Model

Let $\mathbb{R}^d$ be the standard Euclidean space of dimension d and Θ be some space. A parametric distribution family with density function $f(x;\theta)$ with respect to some σ-finite measure is $\mathcal{F} = \{f(x;\theta) : x \in \mathbb{R}^d, \theta \in \Theta\}$. Let $\delta_\theta(\cdot)$ be the Dirac measure such that $\delta_\theta(\mathbb{A}) = 1$ if θ is in set $\mathbb{A}$ and 0 otherwise. Let $G = \sum_{k=1}^{K} w_k \delta_{\theta_k}$ be a discrete probability measure, assigning probability w_k to parameter value $\theta_k \in \Theta$ for some integer $K > 0$. A distribution with the following density function

$$f(x;G) = \int f(x;\theta)\, dG(\theta) = \sum_{k=1}^{K} w_k f(x;\theta_k) \tag{1.1}$$

is called a finite mixture of $\mathcal{F}$. We call $f(x;\theta)$ the subpopulation density function. The elements of $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)^\top$ and that of $\boldsymbol{w} = (w_1, w_2, \ldots, w_K)^\top$ are respectively called the subpopulation parameter and the mixing weight. We use $F(x;\theta)$ and $F(x;G)$ for the Cumulative Distribution Function (CDF) of $f(x;\theta)$ and $f(x;G)$ respectively. Let $\Theta^K = \Theta \times \Theta \times \cdots \times \Theta$ be the Cartesian product over K sets of Θ and Δ_{K-1} be the $(K-1)$-dimensional simplex $\{(w_1, w_2, \ldots, w_K) : w_k \in [0,1], \sum_{k=1}^{K} w_k = 1\}$, we denote the space of mixing distributions G of

order up to K as

$$\mathbb{G}_K = \left\{ G : G = \sum_{k=1}^{K} w_k \delta_{\theta_k}, \boldsymbol{w} \in \Delta_{K-1}, \boldsymbol{\theta} \in \Theta^K \right\}. \tag{1.2}$$

A mixture of (exactly) order K has its mixing distribution G being a member of $\mathbb{G}_K \backslash \mathbb{G}_{K-1} = \{G : G \in \mathbb{G}_K \text{ and } G \notin \mathbb{G}_{K-1}\}$. The order K is also referred to as the number of components of a mixture. A mixture model is a collection of mixture distributions of $\mathcal{F}$.

The finite mixtures are commonly used to model the distribution of population that exhibits heterogeneity. In many applications, the population can be decomposed into several different but homogeneous subpopulations, whose distributions can be modelled by a classical parametric distribution. As early as in 1894, Pearson (1894) applies a Gaussian mixture to analyze crabs' ratio of the forehead to body length data. The histogram of the ratio of forehead to body length of 1000 crabs that Pearson analyzed is shown in Figure 1.1. In this figure, the dashed line is the density function of the single Gaussian distribution fitted to the data. The Gaussian distribution is clearly not a good fit. Based on the general understanding that a well-developed biological species should have its biometrics normally (Gaussian) distributed, Pearson suggests the 1000 crabs are composed of 2 unidentified subspecies. He subsequently fits a 2-component Gaussian mixture to the data and the density function of the fitted mixture is given by the solid line in Figure 1.1. The well-fitted mixture supports the hypothesis of two subspecies of crabs in the collected sample. Finite mixtures are also widely used in other disciplines. In finance, people believe the stock prices in the stock market are either in a "normal" state or a "extreme" state (Liesenfeld, 2001). Hence, the distribution of stock prices often resembles a Gaussian mixture. In the study of the evolution of galaxies, Baldry et al. (2004) suggests the existence of 2 galaxy subpopulations: a passively evolving red galaxy subpopulation and a blue star-forming galaxy subpopulation. A 2-component mixture fitted on the data suggests that there cannot be a continuous evolution, and the rapid change of galaxies in these 2 subpopulations is due to galaxy merger.

In machine learning, finite mixtures are often used as probabilistic models for

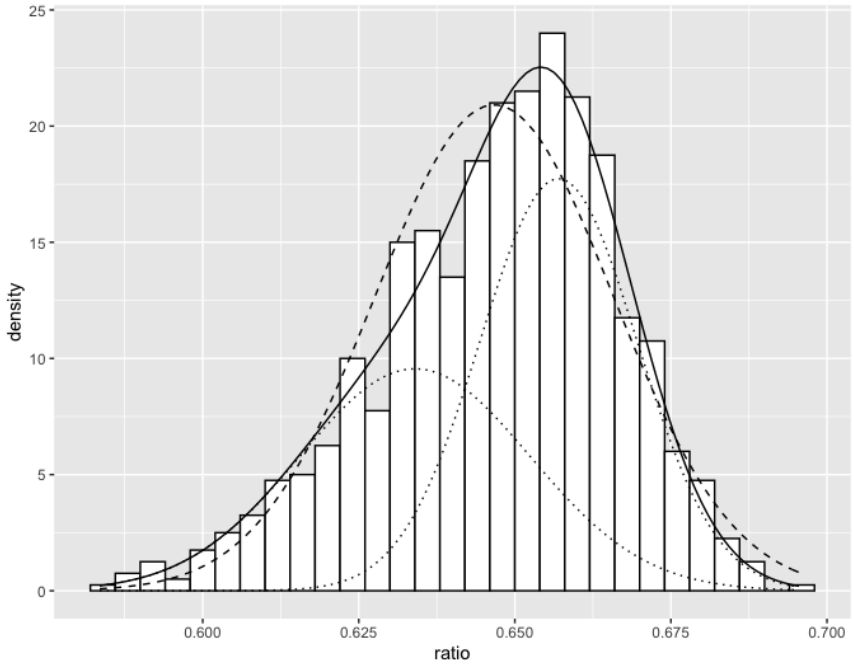


Figure 1.1: Plot of the histogram of the ratio of forehead to body length data on 1000 crabs and of the fitted Gaussian density (dashed line) and two-component Gaussian mixture density (solid line). The two-component Gaussian mixture suggests the crabs may be from two unidentified subspecies (dotted lines).

clustering analysis (Bishop, 2006). The finite mixture model is used in Fraley and Raftery (2002) to cluster breast cancer patients into different groups. Clinically, doctors generally divide the tumours into either malignant or benign types. Their analysis suggests that there may be 3 groups suggesting that the malignant tumour may be in different stages. The finding based on the mixture model is clinically important to determine an appropriate course of action for malignancy. In a clinical example in Baudry et al. (2010), the Gaussian mixture is used to study the development of the graft-versus-host disease (GvHD). GvHD occurs in allogeneic hematopoietic stem cell transplant recipients when donor-immune cells in the graft attack the skin, gut, liver, and other tissues of the recipient. GvHD is diagnosed by clinical and histologic criteria that are often nonspecific and it is typically apparent

only after the disease is well established. In their study, a mixture model is fitted to the bio-marker of GvHD positive patient data and the result suggests the existence of 4 cell subpopulations. These cell subpopulations correspond to colour combinations of lymphocyte phenotypic and activation markers at progressive time points post-transplant.

It is often cited (Nguyen et al., 2020; Titterington et al., 1985) that there always exists a Gaussian mixture whose density function is arbitrarily close to any density function. For example, the kernel density estimate with Gaussian kernel and proper bandwidth is consistent for any continuous density function that vanishes at infinity (Wied and Weißbach, 2012). Finite mixtures are therefore also broadly used as a parametric model to approximate distributions with unknown shapes. Figure 1.2 gives density functions of Gaussian mixtures with various shapes, demonstrating their ability to approximate an arbitrary density. In system design in engineering,

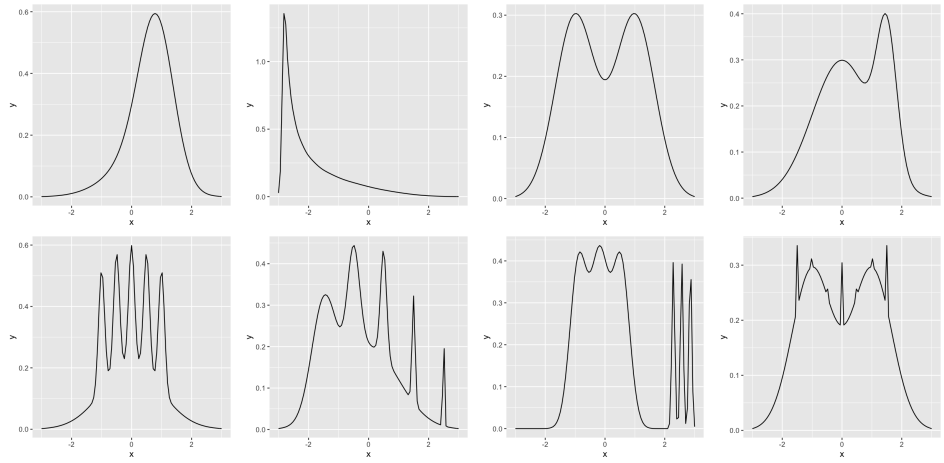


Figure 1.2: Density function of Gaussian mixtures with various shapes in McLachlan and Peel (2004, Section 1.5).

the shape of the distribution of the design life of systems can vary considerably. Bučar et al. (2004) proposes to approximate the density functions of these distributions by finite Weibull mixtures. In Santosh et al. (2013), Brubaker et al. (2015), and Yu et al. (2018), the Gaussian mixtures are used to approximate density functions in Bayesian inference procedures under hidden Markov models for the task of object tracking in video sequences.

Commonly Used Models for Subpopulation

There are a lot of choices for the subpopulation distribution family $\mathcal{F}$. We give several examples of the most commonly used models below.

The finite Gaussian mixtures are by far the most studied finite mixture model. For example, Lo et al. (2001), Chen and Li (2009), and Chen et al. (2012) study the problem of testing the order of a Gaussian mixture. The `mclust` package in `R` (Scrucca et al., 2016) is developed for using finite Gaussian mixtures for model-based clustering, classification, and density estimation in applications. Xu and Jordan (1996) studies the convergence of the Expectation Maximization (EM) algorithm under finite Gaussian mixtures. Various learning approaches under finite Gaussian mixtures (Constantinopoulos and Likas, 2007; Pernkopf and Bouchaffra, 2005; Vlassis and Likas, 2002) are also studied. We give the density function of finite Gaussian mixtures in the following example.

Example 1.1 (Finite Gaussian Mixture)**.** *As the name suggests, a finite Gaussian mixture is a mixture of Gaussian distributions. A d-dimensional Gaussian distribution with mean vector μ and covariance matrix Σ has density function given by*

$$\phi(x;\mu,\Sigma) = \det(2\pi\Sigma)^{-1/2}\exp\left\{-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right\}$$

where det$(\cdot)$ is the determinant of a square matrix. We denote by $\Phi(x;\mu,\Sigma)$ its CDF*. We denote the density of a finite Gaussian Mixture Model (*GMM*) of order K and its* CDF *by*

$$\phi(x;G) = \sum_{k=1}^{K} w_k\phi(x;\mu_k,\Sigma_k); \quad \Phi(x;G) = \sum_{k=1}^{K} w_k\Phi(x;\mu_k,\Sigma_k).$$

Under the finite GMM*, the subpopulation parameter is $\theta = (\mu,\Sigma)$ with its parameter space $\Theta = \mathbb{R}^d \times \mathbb{S}_+^d$ where $\mathbb{S}_+^d$ is the space of all $d \times d$ positive definite matrices.*

Binomial and Poisson mixtures are also broadly investigated in the literature and used in applications. In genetics, the number of recombinants of a family with K offspring has binomial mixture distribution in the presence of genetic mutations (Chernoff and Lander, 1995). The Poisson mixture model is well motivated

for count data such as the number of patents (Wang et al., 1998) and the spinal tumour counts for patients with neurofibromatosis 2 (Joe and Zhu, 2005). The Gamma mixtures are often used to model household income distribution (He and Chen, 2021). These mixtures may be regarded as special cases where the subpopulation distribution family of the mixture, $\mathcal{F}$, is an exponential family. The exponential family is defined as follows.

Definition 1.1 (Exponential Family)**.** *An exponential family is defined as a distribution family whose densities can be represented as*

$$f(x;\theta) = \exp\{\theta^\top T(x) - A(\theta)\}h(x)$$

with respect to some reference measure $\nu(\cdot)$. In this definition, the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_m)^\top$ is called the natural parameter. The natural sufficient statistics is the vector $T(x) = (T_1(x), T_2(x), \ldots, T_m(x))^\top$. The function $h(x)$ modifies the reference measure $\nu(\cdot)$ and the log-partition function $A(\theta)$ is a normalization constant that does not depend on x. The parameter space of θ is usually expanded to be

$$\Theta = \left\{\theta \in \mathbb{R}^m : \int \exp\{\theta^\top T(x)\}\, \nu(dx) < \infty\right\}.$$

In Table 1.1, we list widely used exponential families with their sufficient statistics and parameter space. We do not include the reference measure $\nu(\cdot)$ and $h(x)$ as they are not relevant in statistical inferences.

Another well-investigated class of mixture models has location-scale families as their subpopulation model $\mathcal{F}$. Let $f_0(x)$ be the distribution of a univariate random variable with support $x \in \mathbb{R}$. A location-scale distribution family is formed by all distributions with density function

$$f(x;\theta) = \frac{1}{\sigma} f_0\left(\frac{x-\mu}{\sigma}\right)$$

for $\theta = (\mu, \sigma)^\top$ and parameter space $\Theta = \mathbb{R} \times (0, \infty)$. Some examples of $f_0(x)$ are:

- Univariate Gaussian distribution: $f_0(x) = \phi(x; 0, 1)$;
- Logistic distribution: $f_0(x) = \exp(-x)/\{1 + \exp(-x)\}^2$;

Table 1.1: The natural sufficient statistics, natural parameter, and parameter space of some widely used exponential distribution families.

Name of $\mathcal{F}$	$T(x)$	$A(\theta)$	Θ
		Univariate discrete distributions	
Binomial	x	$\log\{1+\exp(\theta)\}$	$\mathbb{R}$
Poisson	x	$\exp(\theta)$	$\mathbb{R}$
		Univariate continuous distributions	
Exponential	x	$-\log(-\theta)$	$(-\infty, 0)$
Weibull (known k)	x^k	$-\log(-\theta)$	$(-\infty, 0)$
Laplace (known μ)	$\lvert x-\mu\rvert$	$\log(-2/\theta)$	$(-\infty, 0)$
Rayleigh	x^2	$-\log(-2\theta)$	$(-\infty, 0)$
Log-normal	$(\log x, \log^2 x)^\top$	$-\theta_1^2/\theta_2 - 1/\sqrt{2\theta_2}$	$\mathbb{R}\times(-\infty, 0)$
Gamma	$(\log x, x)^\top$	$\log\Gamma(\theta_1+1) - (\theta_1+1)\log(-\theta_2)$	$(-1, \infty)\times(-\infty, 0)$
Inverse Gamma	$(\log x, 1/x)^\top$	$\log\Gamma(-\theta_1-1)) + (\theta_1+1)\log(-\theta_2)$	$(-\infty, -1)\times(-\infty, 0)$

- Gumbel distribution (type I extreme-value distribution): $f_0(x) = \exp\{-x - \exp(-x)\}$.

Naya et al. (2006) uses logistic mixture for thermogravimetric analysis and Salimans et al. (2017) uses this model for image analysis. The Weibull mixture is used by Hernández and Phillips (2006) to characterize end-to-end network delays and by Marín et al. (2005) to model the life time of patients with lupus nephritis. Zhang and Liu (2006) applies a mixture of Weibull distribution to model irregular diameter distribution of forest stands. The Weibull mixture is also used by Carta and Ramirez (2007) to model the distribution of wind speed.

1.2 Research Problems

In this thesis, we address the following research problems under finite mixture models.

1. The properties of the Minimum Wasserstein Distance Estimator (MWDE) under the finite location-scale mixtures.

2. Develop novel procedures for distributed learning of finite mixtures when the datasets are large and (or) stored on different machines. We consider the situation when the data in different machines cannot be directly shared due to privacy or other considerations.

3. Develop a general reduction approach that is widely used for approximate inference in graphical models, such as approximating distributions in recursive procedures and other applications.

We provide a simple description of the motivation for each problem below.

1.2.1 Minimum Distance Estimator

The most fundamental problem of using mixtures in applications is the learning of the mixing distribution G given a set of observations. In statistics, the Maximum Likelihood Estimate (MLE) is usually the first choice to learn the model parameters due to its various nice properties. Under finite mixture models, there is an easy-to-use EM algorithm for its numerical computation. However, under finite location-scale mixtures, the MLE of G is not well defined. The log-likelihood function of G based on a set of Independent and Identically Distributed (IID) observations $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ from a finite location-scale mixture model is given by

$$\ell_N(G|\mathcal{X}) = \sum_{n=1}^{N} \log f(x_n; G) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \frac{w_k}{\sigma_k} f_0\Big(\frac{x_n - \mu_k}{\sigma_k}\Big) \right\}.$$

For an arbitrary mixing distribution $G_\epsilon = 0.5\delta_{(x_1,\epsilon)} + 0.5\delta_{(0,1)}$ with a positive constant ϵ, it is seen that $\ell_N(G_\epsilon|\mathcal{X}) \to \infty$ as $\epsilon \to 0$. Hence, the MLE of G is not well defined or ill defined.

The minimum distance estimator is one of many alternatives to MLE (Blum and Susarla, 1977; Choi, 1969; Choi and Bulgren, 1968; Clarke and Heathcote, 1994; Cutler and Cordero-Brana, 1996; Macdonald, 1971). A minimum distance estimator resembles the MLE in a way as the MLE minimizes the Kullback-Leibler (KL) divergence between the empirical distribution and the assumed model. Let $\mathbb{1}(\cdot)$ be the indicator function and $F_N(x) = N^{-1} \sum_{n=1}^{N} \mathbb{1}(x_n \le x)$ be the empirical distribution. Given a distance $D(\cdot, \cdot)$ on the space of CDFs, a minimum distance estimator is defined to be

$$\widehat{G}_N := \underset{G \in \mathbb{G}_K}{\arg\min}\, D(F_N(\cdot), F(\cdot; G)).$$

Note in the above notation in the distance, we denote by dot the input of the CDFs

to address that the distance is defined between two functions $F_N(\cdot)$ and $F(\cdot; G)$, rather than two values $F_N(x)$ and $F(x; G)$. Table 1.2 lists the distances and the corresponding minimum distance estimators studied under finite mixture models that we are aware of.

Table 1.2: Minimum distance estimators under finite mixture models in the literature. In the table, f is the density function of a mixture, F is the corresponding CDF, F_N is the empirical distribution based on a set of IID sample $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ and $\hat{f}_N$ is a nonparametric density estimate for f based on $\mathcal{X}$, and $x_{(n)}$ is the nth order statistic.

Names of distances and their explicit forms
Wolfowitz distance (Choi, 1969; Choi and Bulgren, 1968) $D_{\mathrm{W}}(F, F_N) = \int (F(x; G) - F_N(x))^2 \, F_N(dx)$ $= N^{-1} \sum_{n=1}^{N} \left\{ \sum_{k=1}^{K} w_k F(x_{(n)}; \theta_k) - n/N \right\}^2$
Cramér-von Mises distance (Macdonald, 1971) $D_{\mathrm{CM}}(F, F_N) = \int (F(x) - F_N(x))^2 \, F(dx)$ $= \{12N^2\}^{-1} + N^{-1} \sum_{n=1}^{N} \left\{ \sum_{k=1}^{K} w_k F(x_{(n)}; \theta_k) - (n - 1/2)/N \right\}^2$
Squared L_2 norm (Clarke and Heathcote, 1994) $D_2(F, F_N) = \|F - F_N\|_2 := \left\{ \int (F(x; G) - F_N(x))^2 \, dx \right\}^{1/2}$
Kolmogorov distance (Blum and Susarla, 1977) $D_{\mathrm{K}}(F, F_N) = \sup_x |F_N(x) - F(x; G)|$
Hellinger distance (Cutler and Cordero-Brana, 1996) $D_{\mathrm{H}}(f, \hat{f}_N) = \|f^{1/2} - \hat{f}_N^{1/2}\|_2$ $= \left\{ \int \left(f^{1/2}(x; G) - \hat{f}_N^{1/2}(x) \right)^2 \, dx \right\}^{1/2}$

Noticeably, the Wasserstein distance is not in Table 1.2. The Wasserstein distance is a byproduct of the optimal transportation theory. It has drawn increased attention in the machine learning community recently due to its intuitive interpretation and good geometric properties (Arjovsky et al., 2017; Evans and Matsen, 2012). This leads to some interesting questions about the minimum distance estimator based on Wasserstein distance (MWDE). Is the MWDE well defined for location-scale mixtures? Is the MWDE as efficient as the PMLE? Is the MWDE robust to model misspecification as other minimum distance estimators? We address all of these questions in Chapter 3.

1.2.2 Distributed Learning of Mixtures

In the era of big data, there are various challenges for statistical inference when dealing with large-scale datasets. The sizes of the datasets for various applications may often be so large that they cannot be stored on a single machine. For example, Google distributes its huge database around the world (Corbett et al., 2013). Distributed data storage is also natural when the datasets are collected and managed by independent agencies. Examples include patient information collected from different hospitals and data collected by different government agencies (Agrawal et al., 2003). Privacy considerations may also make it difficult or even impossible to pool the separate collections of data into a single dataset stored in a single facility. Even if the dataset is stored on a single machine, it may not be possible to load all of it into the computer memory. Data analysis methods should therefore be designed so that they can work with subsets of the dataset, in parallel or sequentially. The information extracted from the subsets can then be combined to draw conclusions about the whole population. For distributed datasets, a commonly used communication efficient inference method to address the privacy concern and the big data concern is a two-step split-and-conquer procedure:

(i) Local inference: standard inference is carried out on local machines;

(ii) Aggregation: the local inference results are transmitted to a central machine to be aggregated.

The split-and-conquer approaches address privacy concerns by sharing only

summary statistics across machines. This also avoids a potentially high transmission cost since the summary statistics have small sizes and only need to be transmitted once. Many split-and-conquer approaches for distributed learning have been developed under many regular models. For example, the split-and-conquer learning of the generalized linear models (Chen and Xie, 2014), kernel ridge regression models (Zhang et al., 2015), ordinary linear models (Chang et al., 2017), and the split-and-conquer estimation of principal eigenspaces (Fan et al., 2019). See also the split-and-conquer version of the Wald-and-score tests (Battey et al., 2015). Most existing approaches first obtain one local estimate of the model parameters based on per local dataset. These local estimates are then pooled and aggregated through a linear averaging operation.

The split-and-conquer-based approaches have their unique challenges under finite mixtures. The parameter of the finite mixture, the mixing distribution, is a discrete distribution with a finite number of support points. A naïve linear averaging operation of the local estimates results in a mixing distribution with an inflated number of support points. The aggregated mixture hence has redundant and spurious subpopulations. Liu and Ihler (2014) develops a KL divergence-based aggregation approach that achieves the best efficiency under models from the exponential family, but its generalization to finite mixture models is not successful. Therefore, it is a completely new and important problem to develop an aggregation approach that is sensitive and computationally efficient with the least distortion. We design a novel aggregation approach in the split-and-conquer framework under the finite mixture model. We present this work in Chapter 4.

1.2.3 Mixture for Approximate Inference

The finite mixture models are also widely used to approximate densities of complex shapes. Recall that any density function can be approximated by a mixture with arbitrary precision as given in Titterington et al. (1985): "provided the number of component densities is not bounded above, certain forms of mixture can be used to provide arbitrarily close approximation to a given probability distribution". Nguyen et al. (2020) shows that any continuous density function that vanishes at infinity can be approximated arbitrarily close by a sequence of location-scale mix-

tures in the supremum norm.

When finite mixtures are used for density approximation, there is a trade-off between the accuracy of the approximation and computational efficiency. A higher-order mixture allows better approximation but leads to more expensive computational costs in downstream applications. For instance, the cost of evaluating the log-likelihood function increases with the order of the mixture. In some Bayesian inference procedures, distributions are mixtures whose orders increase exponentially with recursive operations (Manzar, 2017). A typical scenario is in belief propagation for finding the marginal or conditional distributions in probabilistic graphical models. Some reusable partial sums for the marginalization calculations are referred to as messages under graphical models. Gaussian mixtures are used to approximate the distribution of the messages (Sudderth et al., 2010). The order of this mixture increases exponentially and quickly becomes intractable with iterations. Similarly for the tracking problem under hidden Markov models (Santosh et al., 2013; Yu et al., 2018) when the transition and the likelihood are both approximated by finite Gaussian mixtures. The recursive procedure leads to Gaussian mixture posteriors whose orders increase exponentially. To overcome the computational difficulty, one way is to approximate a higher-order finite Gaussian mixture by one with a lower order. We refer to this problem as Gaussian Mixture Reduction (GMR). GMR is widely employed to control the order of the Gaussian mixture.

There has been a rich literature on mixture reduction, see Crouse et al. (2011) for a thorough review. The existing mixture reduction approaches can be classified into the following three categories. Some greedy approaches (Runnalls, 2007; Salmond, 1990; West, 1993) may identify two close subpopulations and merge them into a single Gaussian repeatedly until the order is reduced to some targeted value. One GMR approach is to group the subpopulations into clusters and have each cluster of subpopulations replaced by a Gaussian distribution (Schieferdecker and Huber, 2009). One may also directly search for a Gaussian mixture of a specific order to best approximate the target mixture (Williams, 2003). Yet more approaches are proposed and some can be regarded as variations of these approaches (Ardeshiri et al., 2013; Assa and Plataniotis, 2018; Yu et al., 2018).

A general understanding is that these greedy algorithms usually lead to suboptimal solutions. The direct search in Williams (2003) posts a challenging opti-

mization problem. The clustering-based approach leads to some respectable solutions but without a theory on algorithm convergence nor what the limits are. We develop an Majorization Maximization (MM) algorithm to link the clustering-based approaches with some optimality targets. We show such algorithms converge at least to some local optimal limits. We also show these algorithms can be further generalized to have better performance. We present the results in Chapter 5.

1.3 Summary of the Contributions and Organization of Thesis

We have provided some background on the research area of this thesis, explained the significance of the research problems, and the proposed approaches. We provide a summary of the organization of the rest of the thesis and the contributions of each research problem below.

1. Chapter 2 contains the preliminaries of this thesis. We introduce the latent structure of the mixture model and its induced model-based clustering method. We give some details about the MLE and the EM algorithm. They are needed for learning mixtures at local machines for the split-and-conquer approach. We then introduce the optimal transportation problem as the basis for divergence between probability measures. Some key notions such as Wasserstein distance and barycentre are given. We also specify some performance metrics to be used to compare the proposed approaches with some popular existing approaches.

2. Chapter 3 focuses on MWDE under the finite location-scale mixtures. It could be an alternative to likelihood approaches. We find it is well defined and consistent, derive a numerical solution, and carry out some simulations. Our moderate scaled simulation study shows this approach suffers some efficiency loss against a penalized version of MLE in general without a noticeable gain in robustness. The MWDE is computationally also more expensive than the penalized MLE. These reaffirm the general superiority of the likelihood-based learning strategies even for non-regular models.

3. In Chapter 4, we develop an effective split-and-conquer approach for the

learning of finite GMM under the distributed setting. We show the proposed estimator is root-n-consistent under some general conditions. Experiments based on simulated and real-world data show that the proposed split-and-conquer approach has comparable statistical performance with the global estimator that is based on the full dataset if the latter is feasible. It can even outperform the global estimator if the model assumption does not match the real-world data. It also has better statistical and computational performance than some existing methods.

4. In Chapter 5, we build a general theoretical framework and a useful algorithm for GMR to be used in approximate inference. We show that the clustering-based algorithms in the literature in fact solve an optimization problem regarding some composite transportation divergence between two mixtures. The optimization problem can be solved by an easy-to-implement MM algorithm. We show that the MM algorithm converges under general conditions. With many possible Composite Transportation Divergence (CTD) at our dispense, there is a great potential to find a GMR that is particularly effective in various situations. Numerical experiments are conducted to illustrate the effectiveness of a class of CTD approaches.

5. In Chapter 6, we point out that although the developments in Chapter 4 and Chapter 5 mainly focus on GMM, these results can be easily modified to apply to other mixtures.

Chapter 2

Preliminaries

As we already presented in the introduction in Chapter 1, there is rich literature on the theory and application of the mixture models. To address various learning problems under finite mixtures, we provide the fundamentals of finite mixtures in this chapter. A summary of the outline of this chapter is given as follows.

The finite mixture is widely used for clustering, which is a natural application based on its latent variable interpretation. Therefore, in Section 2.1, we provide the latent variable interpretation of the mixture model and the corresponding model-based clustering approach. For various learning problems we consider in this thesis, we evaluate their performance for clustering. The performance metric called the adjusted rand index is also introduced in this section.

In Section 2.2, we review some traditional learning approaches under finite mixtures. These approaches, such as the Maximum Likelihood Estimate (MLE) and the corresponding famous Expectation Maximization (EM) algorithm for its numerical computation, are reviewed when the dataset is stored on a single machine.

When a dataset is too large to be stored in a single facility or is stored in a distributed fashion, the cost of transmitting the data and privacy considerations raise challenges for these traditional learning methods. We develop a split-and-conquer procedure for the distributed learning of mixtures in this thesis. Our method learns the mixture via the principle of maximizing the likelihood on local machines independently and aggregates these locally estimated mixtures. Our proposed aggrega-

tion step is a minimum distance-based approach. It is therefore necessary to have a distance or divergence function on the space of either mixing distributions or mixtures. These divergences and distances are introduced in Section 2.3. Among all the divergences and distances we introduce, we devote a lot of effort to the transportation divergence. The transportation divergence, which is a byproduct of the Optimal Transport (OT) theory (Villani, 2003), plays an important role in this thesis. We therefore also introduce some key concepts in the optimal transportation theory instead of merely defining the divergence.

In Section 2.4, we introduce the barycenter of probability distributions, which is a generalization of the average in the space of probability distributions. The barycentres are used in split-and-conquer learning of finite Gaussian mixtures in Chapter 4 and the mixture reduction in Chapter 5.

In Section 2.5, we also include the metrics that are used in this thesis for the numerical evaluation of the performance of various estimators.

2.1 Finite Mixture Model and Clustering

The finite mixture models as defined in (1.1) are often used for clustering. The clustering with the finite mixture model is natural by introducing a latent variable to describe the finite mixture model.

Let $f(x;G) = \sum_{k=1}^{K} w_k f(x;\theta)$ be a mixture on $\mathcal{F} = \{f(x;\theta) : x \in \mathbb{R}^d, \theta \in \Theta\}$ as defined earlier. Let Z and X be two random variables such that the probability mass function of Z is given by

$$\mathbb{P}(Z = k) = w_k$$

for $k \in [K] = \{1, 2, \ldots, K\}$ and

$$\mathbb{P}(X \leq x | Z = k) = F(x;\theta_k)$$

where $F(x;\theta)$ is the CDF of $f(x;\theta)$. It is easily seen that

$$\mathbb{P}(X \leq x) = \sum_{k=1}^{K} \mathbb{P}(X \leq x | Z = k)\mathbb{P}(Z = k) = \sum_{k=1}^{K} w_k F(x;\theta_k).$$

The marginal distribution of X is a mixture of order K with mixing distribution $G = \sum_{k=1}^{K} w_k \delta_{\theta_k}$. The latent variable Z represents the subpopulation membership of X. It is called a latent variable since this variable is generally not observed in real applications.

The latent variable interpretation makes it straightforward to use mixture models for clustering. Given an observation X from a finite mixture $f(x; G)$, there exists a latent variable Z associated with X whose value is missing. By Bayes' theorem, we have

$$\mathbb{P}(Z = k|X = x) = \frac{w_k f(x; \theta_k)}{f(x; G)}.$$

Based on this theorem, the most probable membership of the unit with observed value x is given by

$$\kappa(x) = \kappa(x; G) = \arg\max_{j \in [K]} \{w_j f(x; \theta_j)\}. \tag{2.1}$$

When the mixing distribution G is known or is learned from a set of observations $\{x_1, \ldots, x_N\}$, one may divide the observed values to K clusters by their $\kappa(x_n; G)$ values. Even if the latent variable $Z = z_n$ is known, be aware that we do not necessarily have $z_n = \kappa(x_n; G^*)$ under true mixing distribution G^*.

Given two mixing distributions G_1 and G_2, one can cluster observed values into different groups based on the predicted memberships $\{\kappa(x_n; G_1)\}_{n=1}^{N}$ and $\{\kappa(x_n; G_2)\}_{n=1}^{N}$. To evaluate the similarity of these two clustering outcomes, it is not wise to directly count the total number of matched labels $\sum_{n=1}^{N} \mathbb{1}(\kappa(x_n; G_1) = \kappa(x_n; G_2))$. This is because even if some observations are in the same cluster based on two clustering approaches, they may have different labels when the number of clusters is not the same or the labels of two clusters are switched. Instead, the Adjusted Rand Index (ARI) (Rand, 1971) as defined below is a good performance metric for clustering similarity.

Definition 2.1 (Adjusted Rand Index)**.** *Given two clustering approaches, suppose the observations in a dataset are divided into K clusters $A_1, A_2, \ldots, A_K$ by one method, and K' clusters $B_1, B_2, \ldots, B_{K'}$ by another method. Let $N_i = card(A_i)$, $M_j = card(B_j)$, and $N_{ij} = card(A_i \cap B_j)$ for $i \in [K]$ and $j \in [K']$, where $card(A)$ is cardinality of a set A. The* ARI *of these two clustering outcomes is*

defined to be

$$ARI = \frac{\sum_{i,j}\binom{N_{ij}}{2} - \binom{N}{2}^{-1}\sum_{i,j}\binom{N_i}{2}\binom{M_j}{2}}{\frac{1}{2}\sum_i\binom{N_i}{2} + \frac{1}{2}\sum_j\binom{M_j}{2} - \binom{N}{2}^{-1}\sum_{i,j}\binom{N_i}{2}\binom{M_j}{2}}.$$

where $\binom{n}{k}$ is the number of k combinations from a given set of n elements.

The value of ARI is in the range of $[-1, 1]$. A value closer to 1 indicates a higher degree of agreement of the two clustering approaches. When two clustering methods completely agree with each other, ARI takes the value of 1.

The latent structure interpretation also makes it simple to generate random samples from a mixture. We may draw a random index Z from $[K]$ with probability masses $\boldsymbol{w} = (w_1, w_2, \ldots, w_K)^\top$. Given the value of the latent variable $Z = k$, we then generate a random sample X from $f(x; \theta_k)$. One may repeat this procedure N times to obtain N pairs of (X, Z). Then drop the latent variable Z and $\{X_n\}_{n=1}^N$ is a set of IID samples of size N from the mixture $f(x; G)$. More efficiently, one may first generate random values $N_1, N_2, \ldots, N_K$ from the multinomial distribution with N trials and event probabilities $\boldsymbol{w}$. Then generate N_k IID observations from subpopulation distribution $f(x; \theta_k)$ for $k \in [K]$. The combined observations from subpopulations resemble a set of N IID observations from $f(x; G)$. By this sample generating procedure, observations from the same hidden subpopulation are glued together. One may randomly shuffle the simulated values when necessary.

2.2 Learning Finite Mixtures

The most fundamental statistical task in modelling the data with mixture models is the parameter estimation or the learning of the mixing distribution. In this section, we review some learning methods under finite mixture models.

2.2.1 Method of Moments Estimator

Given a set of observations, it is easy to compute the sample moments of various orders. At the same time, under many parametric models, we can find closed forms of corresponding population moments of its distributions as functions of the parameters. The method of moments for parameter estimation is to select param-

eter values so that the population moments of the specific distribution equal the sample moments. We usually need the number of moments equal to the number of free parameters in the model such that a solution exists and is unique.

Before the widespread of computers in the 1960s, the method of moments was usually the method of choice for parameter estimation under finite mixture models due to its ease of computation (Redner and Walker, 1984). In Pearson's crab example, Pearson (1894) uses the method of moments for estimating the parameters under the two-component univariate Gaussian mixture. A univariate Gaussian distribution has $p = 2$ parameters. Hence, a Gaussian mixture of order K in this case has $2K + (K - 1) = 3K - 1$ free parameters. Pearson must solve the equation system made of the first 5 moments, which is a difficult task back in the 1890s.

The method of moments is not widely used under finite mixture models in modern applications because of its inferior statistical efficiency. Furthermore, with the availability of the EM algorithm and modern computers, the method of moments loses its computational advantage.

2.2.2 Maximum Likelihood Estimator and the EM Algorithm

Given a set of observed data, the maximum likelihood approach estimates the parameters of an assumed model by maximizing its likelihood value. The corresponding estimator is called the MLE. The MLE is the most popular inference method in statistics. On top of the intuitive reasoning of MLE, it also has various good statistical and mathematical properties under regular models. For instance, MLE is asymptotically normal with the lowest possible asymptotic variance.

Under statistical models such as mixture, the MLE loses many of its nice properties. Yet, it is regarded as the most efficient based on empirical evidence albeit not always proved theoretically. Furthermore, the popularity of MLE is benefited from an easy-to-implement EM algorithm under finite mixture models. The MLE is the most popular choice under finite mixture models and the basis of the distributed learning in this thesis. We provide sufficient details for subsequent reference.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be a set of IID observations from the mixture $f(x; G)$ of order K. Under the finite mixture model, the log-likelihood function of

the mixing distribution G is given by

$$\ell_N(G|\mathcal{X}) = \sum_{n=1}^{N} \log f(x_n; G) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} w_k f(x_n; \theta_k) \right\}.$$

The MLE of the mixing distribution G is defined to be

$$\widehat{G}^{\text{MLE}} = \arg\sup_{G \in \mathbb{G}_K} \ell_N(G|\mathcal{X}).$$

As we shown in Section 1.2.1, the MLE can be nonsensical under some mixture models. However, this issue has little impact on the review of the EM algorithm for the numerical computation of MLE under finite mixtures.

The most popular choice for numerical computation of MLE is the EM algorithm. The description of the EM algorithm is most convenient via the latent variable interpretation of the mixture distribution given earlier. For the nth unit with observed value x_n from the finite mixture $f(x; G)$, there is a latent variable Z_n is associated with the observed value. We introduce a one-hot membership vector $\boldsymbol{z}_n = (z_{n1}, \ldots, z_{nK})$ where $z_{nk} = 1$ when $Z_n = k$ and $z_{nj} = 0$ for $j \neq k$.

Suppose the latent variables $\{Z_n, n \in [N]\}$ are known, we have the complete dataset $\{(\boldsymbol{z}_n, x_n), n \in [N]\}$. Then we would also have the complete data log-likelihood function

$$\ell_N^c(G) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log\{w_k f(x_n; \theta_k)\}.$$

Clearly, the complete data log-likelihood cannot be directly used to estimate G since the latent variables are not known in practice. In the EM algorithm, we replace elements in the vector $\boldsymbol{z}_i$ in the complete data log-likelihood by their conditional expectations in each iteration.

At the tth iteration, let $G^{(t)}$ be the current value of the mixing distribution. We can then compute the conditional expectation of z_{nk} given the dataset:

$$w_{nk}^{(t)} = \mathbb{E}(z_{nk}|G^{(t)}, \mathcal{X}) = \frac{w_k^{(t)} f(x_n; \theta_k^{(t)})}{\sum_{j=1}^{K} w_j^{(t)} f(x_n; \theta_j^{(t)})}. \tag{2.2}$$

The conditional expectation of $\ell_N^c(G)$ is then given by

$$Q(G|G^{(t)}) = \sum_{n=1}^{N}\sum_{k=1}^{K} w_{nk}^{(t)} \log\{w_k f(x_n;\theta_k)\}.$$

This is the so called E step of the algorithm.

Instead of seeking the maximizer of the complete data log-likelihood $\ell_N^c(G)$, the M step of the algorithm seeks the maximizer of $Q(G|G^{(t)})$ for $G \in \mathbb{G}_K$. This task is simpler since the subpopulation parameters θ_k are well separated in $Q(G|G^{(t)})$. More specifically, the mixing distribution $G^{(t+1)}$ that maximizes $Q(G|G^{(t)})$ is made of mixing weights

$$w_k^{(t+1)} = N^{-1}\sum_{n=1}^{N} w_{nk}^{(t)}$$

and subpopulation parameters

$$\theta_k^{(t+1)} = \arg\max_{\theta}\left\{\sum_{n=1}^{N} w_{nk}^{(t)} \log f(x_n;\theta)\right\}.$$

The above EM iteration leads to another mixing distribution $G^{(t+1)}$. For many parametric subpopulation models $\mathcal{F}$, there is an explicit solution to $\theta_k^{(t+1)}$. Hence, EM iteration is often very easy to execute. Repeating the iteration leads to a sequence of mixing distributions. Under some conditions, Wu (1983) shows that $\ell_N(G^{(t)})$ is an increasing sequence and the mixing distribution sequence $\{G^{(t)}, t = 1, 2, \ldots\}$ converges to a local maxima of the log-likelihood function. The EM algorithm suffers from a slow algorithmic convergence rate in general, and it can easily be trapped to local maximums. Various approaches (Balakrishnan et al., 2017; Liu and Rubin, 1994; Meilijson, 1989; Meng and Rubin, 1993) have been proposed to speed up the convergence. We do not provide a review on this issue.

The MLE is not well defined for the most widely used finite Gaussian mixtures and finite location-scale mixtures in general. Many variations are proposed to overcome this obstacle. For example, Hathaway (1985) proposes a constrained maximum likelihood formulation that seeks to avoid the unboundedness of the

likelihood function. The resulting estimator has the desired consistency properties. However, this approach alters the parameter space and is not perfect. Ridolfi and Idier (2001) proposes to overcome the issue of the unbounded likelihood function through a Bayesian approach by posing an inverse gamma prior on the scale parameter under the Gaussian mixture models. The posterior mode is recommended as the Maximize a Posterior (MAP) estimator. This MAP estimator, which can be regarded as a penalized Maximum Likelihood Estimate (PMLE) from the frequentist angle, is later shown to be consistent by Chen et al. (2008).

In Chen et al. (2008), a penalized log-likelihood function is defined to be

$$p\ell_N(G|\mathcal{X}) = \ell_N(G|\mathcal{X}) - a_N \sum_{k=1}^{K} p(\theta_k|\mathcal{X})$$

for some penalty function $p(\cdot)$ and a parameter $a_N > 0$ controls the strength of the penalty. The PMLE then becomes

$$\widehat{G}^{\text{pMLE}} = \sup_{G \in \mathbb{G}_K} p\ell_N(G|\mathcal{X}) \tag{2.3}$$

Note that the penalty function is a sum of penalties applied to individual subpopulation parameters. This property is important to ensure easy implementation of the subsequent altered EM algorithm.

There are many possible choices of the penalty function $p(\theta|\mathcal{X})$ under different mixtures of $\mathcal{F}$. We recommend three penalty functions as follows.

- Under finite Gaussian mixtures, let S_x be the sample covariance matrix. Chen and Tan (2009) recommends the penalty function to be

  $$p(\theta|\mathcal{X}) = \text{tr}(\Sigma^{-1} S_x) + \log \det(\Sigma)$$

  where $\text{tr}(\cdot)$ is the trace of a square matrix. The penalty function reduces to

  $$p(\theta|\mathcal{X}) = s_x^2/\sigma^2 + \log \sigma^2$$

  under univariate Gaussian mixtures (Chen et al., 2008).

- Under finite location-scale mixtures with location parameter μ and scale parameter σ, one may replace the sample variance s_x^2 in the previous penalty function with a scale-invariance statistic. One such candidate is the squared sample inter-quartile range. This is helpful if the variance of $f_0(\cdot)$ is not finite.

- Under the finite mixture of two-parameter Gamma distributions, the likelihood is also unbounded. Penalizing the likelihood is also an effective way to restore statistical consistency. Chen et al. (2016) recommends the penalty function to be

$$p(\theta|\mathcal{X}) = r - \log r$$

  where r is the shape parameter in the Gamma distribution.

The EM algorithm for computing the MLE for finite mixture can be easily adapted to compute the PMLE with the recommended penalty function (Chen and Tan, 2009). Using the same latent variable interpretation as given earlier, the penalized complete data log-likelihood is

$$p\ell_N^c(G) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log\{w_k f(x_n;\theta_k)\} - a_N\sum_{k=1}^{K} p(\theta_k|\mathcal{X})$$

The only random quantity in $p\ell_N^c(G)$ is $\{z_{nk}, n \in [N], k \in [K]\}$ when conditioning on $\mathcal{X}$. Therefore, the conditional expectation calculation in (2.2) remains valid. The conditional expectation of $\ell_N^c(G)$ is then given by

$$Q(G|G^{(t)}) = \sum_{n=1}^{N}\sum_{k=1}^{K} w_{nk}^{(t)}\log\{w_k f(x_n;\theta_k)\} - a_N\sum_{k=1}^{K} p(\theta_k|\mathcal{X}).$$

This completes the E step.

The M step is to maximize the above $Q(G|G^{(t)})$ that includes a penalty term. With the recommended penalty function, the subpopulation parameters remain well separated. Clearly, we still have

$$w_k^{(t+1)} = N^{-1}\sum_{n=1}^{N} w_{nk}^{(t)}$$

and the updated subpopulation parameters become

$$\theta_k^{(t+1)} = \arg\max_{\theta} \left\{ \sum_{n=1}^{N} w_{nk}^{(t)} \log f(x_n;\theta) - a_N p(\theta|\mathcal{X}) \right\}. \tag{2.4}$$

Under Gaussian mixtures (Chen and Tan, 2009), the solution to (2.4) has the following closed form

$$\mu_k^{(t+1)} = \{N w_k^{(t+1)}\}^{-1} \sum_{n=1}^{N} w_{nk}^{(t)} x_n,$$
$$\Sigma_k^{(t+1)} = \{2a_N + N w_k^{(t+1)}\}^{-1} \{2a_N S_x + S_k^{(t+1)}\},$$

where

$$S_k^{(t+1)} = \sum_{n=1}^{N} w_{nk}^{(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^\top.$$

For general location-scale mixture, the M step (2.4) does not always have a closed-form. However, one only needs to solve an optimization problem with two variables.

The EM algorithm for the PMLE, like its MLE counterpart, increases the value of the penalized likelihood after each iteration. For all t, we have $\Sigma_k^{(t)} \geq \{2a_N/(N + 2a_N)\}S_x > 0$. The covariance matrices in $G^{(t)}$ have a lower bound that does not dependent on the parameter values. This property ensures that the log-likelihood under Gaussian mixture at $G^{(t)}$ has a finite upper bound. Hence, the above iterative procedure is guaranteed to have $p\ell_N^c(G^{(t)})$ converge to at least a non-degenerate local maximum.

The consistency and asymptotic normality of the PMLE under the Gaussian Mixture Model (GMM) are established in Chen et al. (2008) under univariate case and in Chen and Tan (2009) under multivariate case under the standard IID condition. These results will be used in consistency proof in this thesis later. Hence, we include a simplified version here for reference. We use $(w_k^*, \mu_k^*, \Sigma_k^*)^\top$ to denote the true mixing weight and the true subpopulation parameters in the following Lemma.

Lemma 2.1 (Consistency of PMLE)**.** *Given N* IID *observations from a finite* GMM *with known order K, the* PMLE $\widehat{G}$ *as defined by* (2.3) *with $a_N = N^{-1/2}$ is asymp-*

totically normal with rate $N^{-1/2}$. Specifically, let $(\widehat{w}_k, \widehat{\mu}_k, \widehat{\Sigma}_k)^\top$ denotes the mixing weight and subpopulation parameters based on $\widehat{G}$. It is possible to line up the subpopulation parameters of the true mixing distribution G^* and of the PMLE $\widehat{G}$ *such that*

$$(\widehat{w}_k, \widehat{\mu}_k, \widehat{\Sigma}_k)^\top = (w_k^*, \mu_k^*, \Sigma_k^*)^\top + o(1)$$

and

$$(\widehat{w}_k, \widehat{\mu}_k, \widehat{\Sigma}_k)^\top = (w_k^*, \mu_k^*, \Sigma_k^*)^\top + O_p(N^{-1/2})$$

as $N \to \infty$ in obvious notation.

Chen and Tan (2009) proves the root-n-consistency of the PMLE for a non-random penalty term. The asymptotic result remains valid when the sample covariance matrix is part of the penalty. This is because the sample covariance matrix converges to a positive definite matrix. The assumption of known K is crucial for the claimed rate of convergence. If K is unknown, then the convergence rate of $\widehat{G}$ is far below $N^{-1/2}$. See Chen (1995) and the recent developments in Rousseau and Mengersen (2011), Nguyen (2013), Heinrich and Kahn (2018), and Dwivedi et al. (2020).

2.2.3 Minimum Distance Estimator

The MLE can also be interpreted as a minimum distance estimator (Eguchi and Copas, 2006). The minimum distance estimator interpretation of the MLE motivates other minimum divergence estimators. See Table 1.2 for a list of minimum distance estimators that have been studied in the literature under finite mixture models. Most of these works study the theoretical properties of the minimum distance estimator, the computation of these estimators is usually challenging even in one-dimensional space. Therefore, the minimum distance estimator is generally not used in practice.

2.3 Divergences Between Mixing Distributions or Mixtures

We need various divergences and distances in the distance-based approaches in this thesis. We also need them to assess the quality of various estimators. Regular

models in statistics are usually parameterized by a vector of real numbers. Hence, how well an estimator performs is often measured by the Euclidean distance between the estimated parameter vector and the true parameter vector. This leads to the commonly used Mean Squared Error (MSE). The finite mixture models are not regular. A finite mixture is parameterized by the mixing distribution, which is a probability measure itself. A mixture model can be artificially parameterized by a vector of real numbers when the order is pre-specified. However, it is difficult to line up subpopulations in the estimated mixture and those in the true mixture due to the well-known label switching issue. Therefore, we instead identify a mixture distribution under a mixture model by its mixing distribution and measure the performance of the estimator by the distance between estimated and true mixing distributions.

In this section, we provide several distances under finite mixtures and discuss their pros and cons. Let us first introduce the general notion of divergence and distance.

Definition 2.2 (Divergence and Distance)**.** *Let Θ be a space. A bi-variate function $\rho(\cdot,\cdot)$ defined on Θ is a divergence if $\rho(\theta_1,\theta_2) \geq 0$ for any $\theta_1,\theta_2 \in \Theta$, with equality holds if and only if $\theta_1 = \theta_2$.*

Suppose that $\rho(\cdot,\cdot)$ also satisfies

(i) symmetry: that is, $\rho(\theta_1,\theta_2) = \rho(\theta_2,\theta_1)$, and

(ii) triangle inequality: that is

$$\rho(\theta_1,\theta_2) \leq \rho(\theta_1,\theta_3) + \rho(\theta_3,\theta_2),$$

for all $\theta_1,\theta_2,\theta_3 \in \Theta$.

Then $\rho(\cdot,\cdot)$ is a distance on Θ.

When $\rho(\cdot,\cdot)$ is a distance, we call (Θ,ρ) a metric space.

2.3.1 Commonly Used Divergences Between Mixtures

In this section, we show some commonly used divergences between two mixtures. Let $F(x;G) = \sum_{n=1}^{N} w_n F(x;\theta_n)$ and $F(x;\widetilde{G}) = \sum_{m=1}^{M} \widetilde{w}_m F(x;\widetilde{\theta}_m)$ be the

CDF of two finite mixtures whose subpopulation density functions are from $\mathcal{F} = \{f(x;\theta) : x \in \mathbb{R}^d, \theta \in \Theta\}$. We first consider the divergence or distance between these two mixtures.

Kullback-Leibler (KL) Divergence One choice is the well-known Kullback-Leibler (KL) divergence between two distributions. Even under the most popular and simple Gaussian mixtures, the KL divergence

$$D_{\mathrm{KL}}(\Phi(\cdot;G)\|\Phi(\cdot;\widetilde{G})) = \int \phi(x;G)\log\{\phi(x;G)/\phi(x;\widetilde{G})\}\,dx \tag{2.5}$$

does not have a closed form. One must use numerical approximation to evaluate its value, see approaches in Jensen et al. (2007) and Hershey and Olsen (2007).

Integrated Squared Error (ISE) Another choice is the squared L_2 distance or the Integrated Squared Error (ISE). It is well-defined on the space of continuous measures with proper density functions. Let $F(\cdot)$ and $G(\cdot)$ be two CDFs on $\mathbb{R}^d$, let $f(\cdot)$ and $g(\cdot)$ respectively be the density functions with respect to some measure $\nu(\cdot)$. Then the ISE is defined to be

$$D_{\mathrm{ISE}}(F,G) = \int_{\mathbb{R}^d} |f(x)-g(x)|^2\,\nu(dx). \tag{2.6}$$

Under finite mixtures, the measure $\nu(\cdot)$ is usually chosen to be the Lebesgue measure. Let $S_{GG}, S_{G\widetilde{G}}$ and $S_{\widetilde{G}\widetilde{G}}$ be square matrices of sizes $N \times N$, $N \times M$, and $M \times M$, with their (n,m)th elements given by

$$\int f(x;\theta_n)f(x;\theta_m)\,dx, \quad \int f(x;\theta_n)f(x;\widetilde{\theta}_m)\,dx, \quad \int f(x;\widetilde{\theta}_n)f(x;\widetilde{\theta}_m)\,dx.$$

The ISE between $F(\cdot;G)$ and $F(\cdot;\widetilde{G})$ is is then given by

$$\begin{aligned} D_{\mathrm{ISE}}(F(\cdot;G),F(\cdot;\widetilde{G})) &= \int |f(x;G)-f(x;\widetilde{G})|^2\,dx \\ &= \boldsymbol{w}^\top S_{GG}\boldsymbol{w} - 2\boldsymbol{w}^\top S_{G\widetilde{G}}\widetilde{\boldsymbol{w}} + \widetilde{\boldsymbol{w}}^\top S_{\widetilde{G}\widetilde{G}}\widetilde{\boldsymbol{w}}. \end{aligned}$$

Under the special case of GMM and general notation (μ,Σ) and $(\widetilde{\mu},\widetilde{\Sigma})$ for the

parameters of two Gaussians, we have

$$\int \phi(x;\mu,\Sigma)\phi(x;\widetilde{\mu},\widetilde{\Sigma})\,dx = \phi(\mu;\widetilde{\mu},\Sigma+\widetilde{\Sigma}). \tag{2.7}$$

To see this, let X and E be two independent random variables such that $X \sim \phi(x;\tilde{\mu},\tilde{\Sigma})$ and $E \sim \phi(e;\mu,\Sigma)$. The Left Hand Side (LHS) of (2.7) is the marginal density of $X+E$, clearly $X+E \sim \phi(x;\mu+\tilde{\mu},\Sigma+\tilde{\Sigma})$. See also Williams (2003, Appendix A-1).

The property of Gaussian distribution in (2.7) leads to the ISE between two finite Gaussian mixtures given by

$$\begin{aligned} D_{\text{ISE}}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})) = &\sum_{n=1}^{N}\sum_{n'=1}^{N} w_n w_{n'}\phi(\mu_n;\mu_{n'},\Sigma_n+\Sigma_{n'}) \\ &- 2\sum_{n=1}^{N}\sum_{m=1}^{M} w_n \widetilde{w}_m \phi(\mu_n;\widetilde{\mu}_m,\Sigma_n+\widetilde{\Sigma}_m) \\ &+ \sum_{m=1}^{M}\sum_{m'=1}^{M} \widetilde{w}_m \widetilde{w}_{m'}\phi(\widetilde{\mu}_m;\widetilde{\mu}_{m'},\widetilde{\Sigma}_m+\widetilde{\Sigma}_{m'}). \end{aligned} \tag{2.8}$$

The square root of the ISE is the L_2 distance that satisfies the symmetry and triangle inequality as given in Definition 2.2.

Cauchy-Schwarz (CS) Divergence The third choice we recommend is called CS divergence (Jenssen et al., 2006). With the same notation as above, the CS divergence between F and G is defined to be

$$D_{\text{CS}}(F,G) = -\log \frac{\int f(x)g(x)\,\nu(dx)}{\sqrt{\int f^2(x)\,\nu(dx)\int g^2(x)\,\nu(dx)}}$$

Applying (2.7), the CS divergence between two Gaussian mixtures has the following closed expression

$$D_{\text{CS}}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})) = -\log \frac{\boldsymbol{w}^\top S_{G\widetilde{G}}\widetilde{\boldsymbol{w}}}{\sqrt{\boldsymbol{w}^\top S_{GG}\boldsymbol{w}}\sqrt{\widetilde{\boldsymbol{w}}^\top S_{\widetilde{G}\widetilde{G}}\widetilde{\boldsymbol{w}}}}.$$

Note that the CS divergence is symmetric in two arguments.

These divergences are defined between two mixtures. One may measure the similarity between two mixing distributions by their corresponding mixtures. Is there any distance defined directly between two discrete distributions that can be directly used to measure the similarity between two mixing distributions? A special distance of the transportation divergence, the Kantorovich distance, is widely used to measure the similarity between two discrete distributions (Deng and Du, 2009). We introduce the transportation divergence between two distributions in the next section.

2.3.2 Transportation Divergence and Wasserstein Distance under Mixtures

The transportation divergence and the Wasserstein distance are the byproducts of optimal transportation theory. They are part of the key ingredients of this thesis. In this section, we first briefly review the history of the optimal transportation theory and the two fundamental problems where the optimal transportation theory is originated. We refer the interested readers to Villani (2003) and Peyré and Cuturi (2019) for details where the former focuses on the theoretical aspect and the latter focuses on the computational aspect of the optimal transport.

A Brief History Optimal Transport

Optimal Transport (OT) problem has a long history starting from 1781 when the French mathematician and physicist *Gaspard Monge (1746-1818)* formulated the original problem. He considered the problem of how to move a pile of sand to fill up a hole at a minimal cost. The assignment of the sand from the original location to the destination location is called "transport plan" or "transportation plan". The terminology "optimal transport" is used since the problem searches for the transportation plan that gives the minimum cost. This formulation is hence called Monge's problem in the literature.

Later in 1947, the Russian economist *Leonid Kantorovich (1912-1986)* reformulated Monge's problem. Kantorovich's formulation is usually called Kantorovich's relaxation or Monge-Kantorovich problem in their honour. Kantorovich received Nobel Prize in 1975 for his contribution to the optimal transportation the-

ory in economics. The OT problem later draws a lot of attention from mathematicians since they find that the OT problem leads to many interesting mathematical problems. For example, Yann Brenier's work in 1987 on the existence and uniqueness of the optimal transportation plan paved the way towards a beautiful interplay between partial differential equations, geometry, probability theory, and functional analysis (Villani, 2003).

In the statistics community, the Wasserstein distance is usually used. It characterizes the convergence rate of the mixing distribution estimator under mixture models by Nguyen (2013). Recently, the Wasserstein distance gained popularity as a natural metric for the dissimilarity of two probability distributions (Dedecker and Merlevede, 2017). It is also widely used in functional data analysis (Chen et al., 2021; Chen and Muller, 2021). See Panaretos and Zemel (2019) for a comprehensive review of the use of OT in statistics.

Monge's Problem

Consider the sand moving problem as illustrated in Figure 2.1. Let there be a pile

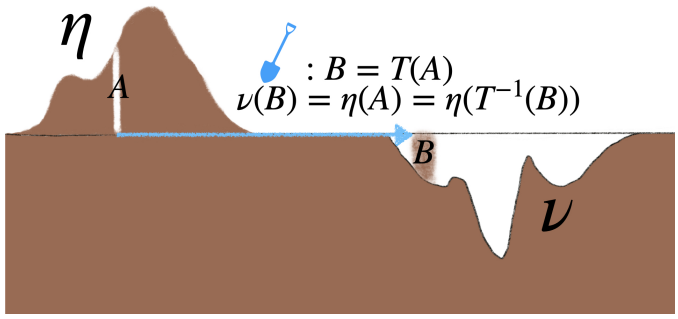


Figure 2.1: Illustration of Monge's problem.

of sand and a hole. We want to fill up the hole with sand completely. To fill up the hole, the volume of the sand and the volume of the hole must be the same. Without loss of generality, let the volumes of both sand and hole be 1. We use two probability measures η and ν to represent the distribution of sand and hole. Denote by $c(x, y)$ the unit cost of moving the sand at location x to the hole at location y.

The total cost to move all the sand depends on how we move the sand. Monge's problem seeks the transportation plan with the minimum cost to complete this task.

To formulate the problem mathematically, let Θ_1 and Θ_2 be two spaces and $\mathcal{P}(\Theta_1)$ and $\mathcal{P}(\Theta_2)$ be two spaces of probability measures on Θ_1 and Θ_2 respectively. Let $\eta \in \mathcal{P}(\Theta_1)$ and $\mathscr{T}$ be a map from Θ_1 to Θ_2.

Definition 2.3 (Push Forward Measure)**.** *A measure $\nu \in \mathcal{P}(\Theta_2)$ is a push forward measure of η by T if for any measurable set $\mathbb{B} \subset \Theta_2$, $\nu(\mathbb{B}) = \eta(\mathscr{T}^{-1}(\mathbb{B}))$.*

We denote the push forward measure of η by $\mathscr{T}$ as $\nu = \mathscr{T}_{\#}\eta$. We now introduce cost function as a bi-variate function $c(\cdot, \cdot)$ on $\Theta_1 \times \Theta_2$ such that $c(\theta_1, \theta_2) \geq 0$ for any $\theta_1 \in \Theta_1$ and $\theta_2 \in \Theta_2$.

Definition 2.4 (Monge's Problem)**.** *Let η and ν be two probability measures on Θ_1 and Θ_2 respectively. Let $c(\cdot, \cdot)$ be a cost function on $\Theta_1 \times \Theta_2$. For a given measure $\eta \in \mathcal{P}(\Theta_1)$ and a map $\mathscr{T}$ from Θ_1 to Θ_2, define*

$$\mathcal{I}_c(\mathscr{T}) = \int_{\Theta} c(x, \mathscr{T}(x))\, \eta(dx) \tag{2.9}$$

which is the total cost of moving η to $\nu = \mathscr{T}_{\#}\eta$. Monge's problem is to find a transportation plan $\mathscr{T}^*$ such that

$$\mathcal{I}_c(\mathscr{T}^*) = \inf\{\mathcal{I}_c(\mathscr{T}) :\ \mathscr{T}_{\#}\eta = \nu\}.$$

We call $\mathscr{T}^*$ an OT *plan.*

The total transportation cost $\mathcal{I}_c(\mathscr{T})$ as defined in (2.9) is induced by a transport plan $\mathscr{T}$ with unit cost $c(\cdot, \cdot)$. It can be interpreted as follows in the previous sand moving example. The $\eta(dx)$ amount of sand at location x is moved to the hole at location $\mathscr{T}(x)$ with unit cost being $c(x, \mathscr{T}(x))$. The push-forward measure $\nu = \mathscr{T}_{\#}\eta$ requires the hole with shape ν is filled up completely without any hollow.

Remark 2.1 (Non-existence of OT Plan in Monge's Problem)**.** *The* OT *Plan in Monge's problem does not always exist. Consider two discrete measures $\eta = 0.5\delta_{-1} + 0.5\delta_1$ and $\nu = \delta_0$ on real space $\Theta_1 = \Theta_2 = \mathbb{R}$. It is obvious that $\mathscr{T}(x) = |x| - 1$ is a transportation plan from η to ν. However, the transportation*

plan from ν to η does not exist: $\nu(\mathscr{T}^{-1}(\mathbb{B}))$ takes value either 0 or 1 for any $\mathbb{B}$ and $\mathscr{T}$. Hence, the push forward measure of any $\mathscr{T}$ can not assign measure 0.5 at either location -1 or 1 required by η.

To solve the issue that the optimal transport plan may not exist for the transportation problem based on Monge's formulation, Kantorovich proposes a relaxation of Monge's problem. This leads to the Monge-Kantorovich problem.

Monge-Kantorovich Problem

Kantorovich's relaxation allows splitting the sand at location x to different proportions and moving them to different locations of the hole. By allowing the partition of the sand, the Monge-Kantorovich problem learns a stochastic instead of a deterministic transportation plan.

For simplicity, let us consider the case when $\Theta_1 = \Theta_2 = \Theta$. For any $\boldsymbol{\pi} \in \mathcal{P}(\Theta^2)$, denote its marginal measures to be $\boldsymbol{\pi}_{1\cdot}$ and $\boldsymbol{\pi}_{\cdot 2}$. For any $\eta, \nu \in \mathcal{P}(\Theta)$, define the space of the couplings of η and ν to be

$$\Pi(\eta, \nu) = \{\boldsymbol{\pi} \in \mathcal{P}(\Theta^2) : \boldsymbol{\pi}_{1\cdot} = \eta,\ \boldsymbol{\pi}_{\cdot 2} = \nu\}.$$

The coupling space $\Pi(\eta, \nu)$ consists of bi-variate measures with marginal measures η and ν. For convenience, we denote $\Pi(\eta, \cdot)$ as the space of measures with first marginal measure being η, and similarly for $\Pi(\cdot, \nu)$.

Definition 2.5 (Monge-Kantorovich Problem)**.** *Let η, ν, and $c(\cdot, \cdot)$ be two measures and cost function as in Definition 2.4. For any $\boldsymbol{\pi} \in \mathcal{P}(\Theta^2)$, let*

$$\mathcal{I}_c(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\{c(X, Y)\} = \int_{\Theta^2} c(x, y)\, \boldsymbol{\pi}(dx, dy). \tag{2.10}$$

Kantorovich's relaxation of the optimal transportation problem is to find a $\boldsymbol{\pi}^* \in \mathcal{P}(\Theta^2)$ such that

$$\mathcal{I}_c(\boldsymbol{\pi}^*) = \inf\{\mathcal{I}_c(\boldsymbol{\pi}) : \boldsymbol{\pi} \in \Pi(\eta, \nu)\}.$$

Intuitively, $\boldsymbol{\pi}(x, y)$ indicates how much "mass" is transported from x to y. The first constraint $\int_\Theta \boldsymbol{\pi}(x, dy) = \eta(x)$ on $\boldsymbol{\pi}$ is to ensure that the sand at location x is

spread over the space of y. The second constraint $\int_\Theta \boldsymbol{\pi}(dx, y) = \nu(y)$ ensures that the hole at location y is filled up with sand from all places.

When $c(x, y)$ is lower semi-continuous, Villani (2003, Theorem 1.3) shows the infimum over $\Pi(\eta, \nu)$ is attainable and optimal transportation plans exist. This formulation is called the relaxation of Monge's problem since Monge's problem is its special case and its solution π has the special form

$$\boldsymbol{\pi}(dx, dy) = \eta(dx)\delta_{y=T(x)}(dy).$$

The notation $\delta_{y=T(x)}(\cdot)$ is a Dirac mass at y that equals $T(x)$.

The application of OT requires the computation of the optimal transport plan. There does not exist a closed-form solution for the optimal transport plan $\boldsymbol{\pi}^*$ in general. Some numerical algorithms are developed to find the optimal transport plan between two discrete measures. Approximate optimal transport plans are usually obtained by discretizing the continuous measures (Peyré and Cuturi, 2019).

Numerical Computation of Optimal Transport Plans

Let $\eta = \sum_{n=1}^N u_n \delta_{x_n}$ and $\nu = \sum_{m=1}^M v_m \delta_{y_m}$ be two discrete probability measures on some space Θ. Let the cost function be $c(x, y)$ and C be a cost matrix such that its (n, m)th element $C_{nm} = c(x_n, y_m)$. Let $\boldsymbol{\pi} = \sum_{n,m} \pi_{nm}\delta_{(x_n,y_m)}$ be a transportation plan from η to ν so that it is a member of

$$\Pi(\eta, \nu) = \left\{ \boldsymbol{\pi} = \sum_{n,m} \pi_{nm}\delta_{(x_n,y_m)} : \sum_{m=1}^M \pi_{nm} = u_n \text{ and } \sum_{n=1}^N \pi_{nm} = v_m \right\}.$$

By Definition 2.10, the total cost of transporting η to ν based on plan $\boldsymbol{\pi}$ is given by

$$\mathcal{I}_c(\boldsymbol{\pi}) = \sum_{n=1}^N \sum_{m=1}^M \pi_{nm} C_{nm}.$$

This formulation enables us to use linear programming to find the optimal transportation plan $\boldsymbol{\pi}^*$ (Peyré and Cuturi, 2019, Section 3.1).

Let $\boldsymbol{c} = \text{Vec}(C)$ be a vector formed by the entries of matrix C column-wise.

Similarly, let $\overrightarrow{\boldsymbol{\pi}} = \text{Vec}(\boldsymbol{\pi})$. We create a matrix

$$A = \begin{pmatrix} \mathbf{1}_N^\top \otimes \mathbf{I}_M \\ \mathbf{I}_N \otimes \mathbf{1}_M^\top \end{pmatrix}$$

where $\mathbf{1}_N$ is a vector of all 1s with length N, $\mathbf{I}_M$ is an identity matrix with dimension M, and $\otimes$ is the Kronecker product between two matrices. Recall $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)^\top$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_m)^\top$ are the weights of η and ν in vector form. The optimal transportation plan from η to ν can then be presented as

$$\overrightarrow{\boldsymbol{\pi}}^* = \arg\min \left\{ \boldsymbol{c}^\top \overrightarrow{\boldsymbol{\pi}} : \overrightarrow{\boldsymbol{\pi}} \in \mathbb{R}_+^{NM}, A\overrightarrow{\boldsymbol{\pi}} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} \right\}.$$

The optimization problem is exactly a linear programming problem. Most linear programming algorithms, such as simplex or interior point methods, have algorithm complexity at $\mathcal{O}(N^3 \log(N))$ for such a problem when $N = M$ (Cuturi, 2013). Since the cost is cubic in N, the computational cost is considered very expensive when N is large. To reduce the computation burden, Cuturi (2013) proposes an entropic regularized problem. This problem is computationally less expensive with a solution approximates the original transportation plan.

Definition 2.6 (Entropic Regularized Optimal Transport)**.** *Let probability measures η, ν, the transportation plan $\boldsymbol{\pi}$, and the cost matrix C all be the same as given earlier. Let*

$$\mathcal{H}(\boldsymbol{\pi}) = -\sum_{n=1}^{N}\sum_{m=1}^{M} \pi_{nm}(\log \pi_{nm} - 1),$$

be a version of entropy of $\boldsymbol{\pi}$. Let $\lambda \geq 0$ and

$$\mathcal{I}_{c,\lambda}(\boldsymbol{\pi}) = \mathcal{I}_c(\boldsymbol{\pi}) - \lambda\mathcal{H}(\boldsymbol{\pi}).$$

An entropic regularized optimal transport plan between η and ν with regularization strength $\lambda > 0$ is defined to be a transport plan $\boldsymbol{\pi}^*$ satisfies

$$\mathcal{I}_{c,\lambda}(\boldsymbol{\pi}^*) = \min\{\mathcal{I}_{c,\lambda}(\boldsymbol{\pi}) : \boldsymbol{\pi} \in \Pi(\eta, \nu)\}. \tag{2.11}$$

It can be found that $\mathcal{I}_{c,\lambda}(\boldsymbol{\pi})$ is strongly convex. Hence, it has a unique minimum $\boldsymbol{\pi}^*_\lambda$. Peyré and Cuturi (2019) shows that as $\lambda \to \infty$, the regularization term dominates and

$$\boldsymbol{\pi}^*_\lambda \to \arg\max\{\mathcal{H}(\boldsymbol{\pi}) : \boldsymbol{\pi} \in \Pi(\eta, \nu)\}.$$

In the vector form, we have $\overrightarrow{\boldsymbol{\pi}}^*_\lambda = \boldsymbol{u}\boldsymbol{v}^\top$.

Let $\boldsymbol{a} = (a_1, a_2, \ldots, a_N)^\top$. Denote the diagonal matrix whose nth main diagonal element is a_n by $\text{diag}(\boldsymbol{a})$.

Proposition 2.1. *Let K be a matrix with its (n, m)th entry $K_{nm} = \exp(-C_{nm}/\lambda)$. The vector form of the unique solution to* (2.11) *can be expressed as*

$$\overrightarrow{\boldsymbol{\pi}}^*_\lambda = \{\textit{diag}(\boldsymbol{a})\}\, K\, \{\textit{diag}(\boldsymbol{b})\}$$

for some $\boldsymbol{a} \in \mathbb{R}^N_+$ and $\boldsymbol{b} \in \mathbb{R}^M_+$.

By Proposition 2.1, the optimal solution $\overrightarrow{\boldsymbol{\pi}}^*_\lambda$ satisfies constraints

$$\{\text{diag}(\boldsymbol{a})\}\, K\, \{\text{diag}(\boldsymbol{b})\}\, \mathbf{1}_M = \boldsymbol{u}, \quad \{\text{diag}(\boldsymbol{b})\}\, K^\top\, \{\text{diag}(\boldsymbol{a})\}\, \mathbf{1}_N = \boldsymbol{v}.$$

Since $\text{diag}(\boldsymbol{b})\mathbf{1}_M = \boldsymbol{b}$, these constraints can also be written as

$$\boldsymbol{a} \odot (K\boldsymbol{b}) = \boldsymbol{u}, \quad \boldsymbol{b} \odot \left(K^\top \boldsymbol{a}\right) = \boldsymbol{v} \tag{2.12}$$

where $\odot$ is the element-wise product of two vectors. Proposition 2.1 can be proved using Lagrangian multiplier method (Peyré and Cuturi, 2019, Section 4.2). Based on this proposition, the entropic regularized optimal transportation plan can be obtained efficiently via an iterative minimization scheme that only involves simple matrix operations. An intuitive iterative scheme to solve (2.12) is known as the Sinkhorn's algorithm:

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{u}/\{K\boldsymbol{b}^{(t)}\}, \quad \boldsymbol{b}^{(t+1)} = \boldsymbol{v}/\{K^\top \boldsymbol{a}^{(t+1)}\}$$

where the division operator between two vectors is defined entry-wise.

Transportation Divergence and Wasserstein Distance – A General Definition

We have given the intuition behind the optimal transportation plan in the previous section. It is seen that a byproduct of OT is that the optimal transportation cost is a divergence on the space of probability measures. For some choice of cost functions, the optimal transportation cost becomes a distance, which is called the Wasserstein distance (Villani, 2003). The Wasserstein distance measures the similarity between two mixing distributions or mixtures (Nguyen, 2013). In this thesis, we use the transportation divergence for aggregation and approximate inference.

Definition 2.7 (Transportation Divergence)**.** *Let η and ν be two probability measures, and $c(\cdot, \cdot)$ be a cost function on Θ. Let $\mathcal{I}_c(\boldsymbol{\pi})$ be the same as that in* (2.10)*. Denote by*

$$\boldsymbol{\pi}^* = \arg\inf\{\mathcal{I}_c(\boldsymbol{\pi}) : \boldsymbol{\pi} \in \Pi(\eta, \nu)\}$$

the corresponding optimal transportation plan. We call

$$\mathcal{T}_c(\eta, \nu) = \mathcal{I}_c(\boldsymbol{\pi}^*)$$

the transportation divergence.

Based on the intuition behind OT, the transportation divergence captures the human perception of similarity very well. Therefore, the transportation divergence becomes a natural candidate to measure the similarity between probability measures.

The transportation divergence is defined through a cost function $c(\cdot, \cdot)$ on $\Theta \times \Theta$. Because the cost function is non-negative, the optimal transportation cost is a divergence between two measures. The cost function is also very helpful when Θ has complex structures such as trees and graphs. For example, it is used to measure the similarity between phylogenetic trees (Evans and Matsen, 2012). This property distinguishes itself from other distances such as the total variation distance and Kolmogorov Smirnov distance.

Suppose $D(\cdot, \cdot)$ is a distance on space Θ and $r \geq 1$ is a real number. We say a probability measure $\eta \in \mathcal{P}(\Theta)$ has finite rth moment if there exists an $\theta_0 \in \Theta$

such that

$$\int_{\theta\in\Theta} D^r(\theta,\theta_0)\,\eta(d\theta) < \infty.$$

In fact, if the above integration is finite for some θ_0, it is finite when θ_0 is replaced by any other values in Θ. We denote $\mathcal{P}_r(\Theta)$ as the space of probability measures on Θ with finite rth moment.

If one chooses the cost function in the transportation divergence to be $c(\cdot,\cdot) = D^r(\cdot,\cdot)$ for some $r \geq 1$, then the transportation divergence becomes the famous r-Wasserstein distance. The following result can be found in Villani (2003, Theorem 7.3).

Proposition 2.2 (r-Wasserstein distance)**.** *Let $D(\cdot,\cdot)$ be a distance defined on space Θ and $c(\cdot,\cdot) = D^r(\cdot,\cdot)$ for some $r \geq 1$. Then*

$$W_{D,r}(\cdot,\cdot) = \mathcal{T}_c^{1/r}(\cdot,\cdot)$$

is a distance on $\mathcal{P}_r(\Theta)$. We call $W_{D,r}(\cdot,\cdot)$ r-Wasserstein distance.

Remark 2.2. *We refer to distance $D(\cdot,\cdot)$ on Θ as ground distance. When Θ is an Euclidean space, the most widely used ground distance is the Euclidean distance $D(x,y) = \|x-y\|$. We simplify the notation of this version of Wasserstein distance into W_r, omitting the ground distance D in the subscript.*

Remark 2.3. *Let X and Y be two random variables with probability laws η and ν and cumulative distribution functions F and G. We adopt the convention*

$$W_r(X,Y) = W_r(F(\cdot),G(\cdot)) = W_r(\eta,\nu).$$

The Wasserstein distance is often used in probability theory for the metrization of the weak convergence of probability measures as well as the convergence of the moments. The following results are used when we study the properties of the Wasserstein distance-based minimum distance estimator of finite location-scale mixtures. The next two lemmas can be found in Villani (2003, Chapter 7).

Lemma 2.2 (Weak convergence)**.** *Let $X_1, X_2, \ldots, X_n, \ldots$ be a sequence of random variables and Y is another random variable. Assume that all of these random*

variable have finite rth moment. Then $W_r(X_n, Y) \to 0$ if and only if (i) $X_n \xrightarrow{d} Y$ and (ii) $\mathbb{E}\{D^r(X_n, x_0)\} \to \mathbb{E}\{D^r(Y, x_0)\}$ for some (and therefore all) nonrandom constant x_0. The $\mathbb{E}(\cdot)$ is the expectation of the corresponding function with respect to the distribution of the random variable with obvious notation.

The Wasserstein distance also has a useful ordering property.

Lemma 2.3 (Ordering of Wasserstein distance)**.** *For any $q \geq p \geq 1$,*

$$W_q(\eta, \nu) \geq W_p(\eta, \nu). \tag{2.13}$$

Despite many of their nice properties, the use of transportation divergence or Wasserstein distance is hindered by the computation challenge in general. In some special cases, the Wasserstein distance has a closed-form as shown in the following two examples.

Example 2.1 (Wasserstein distance between measures on $\mathbb{R}$.)**.** *Let η and ν be two probability measures on the one-dimensional Euclidean space $\mathbb{R}$ and $F(x)$ and $G(x)$ be their corresponding CDFs. Then the r-Wasserstein distance between η and ν has a closed-form expression*

$$W_r(\eta, \nu) = \left(\int_0^1 |F^{-1}(t) - G^{-1}(t)|^r dt \right)^{1/r} \tag{2.14}$$

where $F^{-1}(t) = \inf\{x : F(x) \geq t\}$ and $G^{-1}(t) = \inf\{x : G(x) \geq t\}$ are quantile functions.

Example 2.2 (2-Wasserstein distance between two Gaussians)**.** *Let X and Y be two Gaussian random vectors with mean vectors μ_X and μ_Y, and covariance matrices Σ_X and Σ_Y. Let the ground distance be the Euclidean distance:*

$$D(x, y) = \|x - y\|.$$

Then, the 2-Wasserstein distance has the following closed-form

$$W_2^2(X, Y) = \|\mu_X - \mu_Y\|^2 + tr\Big(\Sigma_X + \Sigma_Y - 2(\Sigma_X^{1/2}\Sigma_Y\Sigma_X^{1/2})^{1/2}\Big).$$

See (Peyré and Cuturi, 2019).

With the definition of transportation divergence and Wasserstein distance, we can apply them to directly measure the similarity between two mixing distributions. Let $f(x;G) = \sum_{n=1}^{N} w_n f(x;\theta_n)$ and $f(x;\widetilde{G}) = \sum_{m=1}^{M} \widetilde{w}_m f(x;\widetilde{\theta}_m)$ be two finite mixtures parameterized by mixing distributions G and $\widetilde{G}$. Let $\boldsymbol{w}$ and $\widetilde{\boldsymbol{w}}$ be two vectors represent their corresponding mixing weights. Let the ground cost function $c(\cdot,\cdot) : \Theta \times \Theta \to \mathbb{R}_+$ be any sensible divergence on the parameter space Θ. Based on the definition of the transportation divergence, the transportation plan between two discrete measures is also a discrete measure. To simplify the notation, we denote by

$$\Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{N\times M} : \boldsymbol{\pi}\mathbf{1}_M = \boldsymbol{w}, \boldsymbol{\pi}^\top \mathbf{1}_N = \widetilde{\boldsymbol{w}}\}$$

the coupling between G and $\widetilde{G}$. In this notation, we denote by $\boldsymbol{\pi}$ the weights that the transportation plan puts on its support points. We use this matrix notation for the transportation plan between two discrete measures in the rest of the thesis.

The transportation divergence between G and $\widetilde{G}$ then becomes

$$\mathcal{T}_c(G, \widetilde{G}) = \inf\left\{\sum_{n,m} \pi_{nm} c(\theta_n; \widetilde{\theta}_m) : \boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})\right\}.$$

Under finite GMM with parameter space $\Theta = \mathbb{R}^d \times \mathbb{S}_+^d$, the ground cost function between two parameters $\theta_n = (\mu_n, \Sigma_n)$ and $\widetilde{\theta}_m = (\widetilde{\mu}_m, \widetilde{\Sigma}_m)$ can be chosen to be

$$c(\theta_n, \widetilde{\theta}_m) = \|\mu_n - \widetilde{\mu}_m\| + \|\Sigma_n^{1/2} - \widetilde{\Sigma}_m^{1/2}\|_F \tag{2.15}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

The transportation divergence can be used to measure the similarity between any distributions. Is it helpful to measure the similarity between two mixtures? The transportation divergence between two mixtures with the cost function being the Euclidean distance on the sample space is well defined. However, this transportation divergence is difficult to compute for continuous mixtures. We may take advantage of the special structure of finite mixture to find some version of transportation divergence that permits easy computation. This consideration leads to a

Composite Transportation Divergence (CTD) discussed in Nguyen (2013), which we introduce in the next section.

Composite Transportation Divergence under Mixtures

The key to the easy-to-compute transportation divergence between finite mixtures is to view the finite mixtures as discrete distributions on the space of probability distributions. Let $\mathcal{F} = \{f(x;\theta) : x \in \mathbb{R}^d, \theta \in \Theta\}$ be a parametric distribution family of subpopulations and $\widetilde{\mathcal{F}} = \{F(x;\theta) : x \in \mathbb{R}^d, \theta \in \Theta\}$ be the corresponding collection of CDFs. A finite mixture is a discrete distributions on $\mathcal{F}$. Let $c(\cdot,\cdot) : \widetilde{\mathcal{F}} \times \widetilde{\mathcal{F}} \to \mathbb{R}_+$ be a divergence defined on $\tilde{\mathcal{F}}$.

Definition 2.8 (CTD between Mixtures)**.** *We define the* CTD *between mixtures to be*

$$\mathcal{T}_c(F(\cdot;G), F(\cdot;\widetilde{G})) = \inf_{\boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})} \left\{ \sum_{n,m} \pi_{nm} c(F(\cdot;\theta_n), F(\cdot;\widetilde{\theta}_m)) \right\}. \quad (2.16)$$

As we discussed previously for the numerical computation of OT, the cost for evaluating the divergence is high when the order N of mixture $F(x;G)$ is large. We can similarly replace it with an entropic regularized version of CTD between two mixtures which admits a low computational cost. The notations in the following definition are the same as before.

Definition 2.9 (Entropic Regularized CTD)**.** *Let $F(\cdot;G)$ and $F(\cdot;\widetilde{G})$ be two mixture Cumulative Distribution Function (*CDF*)s as defined earlier. Let $\mathcal{H}(\boldsymbol{\pi}) = -\sum_{i,j} \pi_{ij}(\log \pi_{ij} - 1)$ be the entropy of the transportation plan $\boldsymbol{\pi}$. We define the entropic regularized* CTD *between two mixtures to be*

$$\mathcal{T}_{c,\lambda}(F(\cdot;G), F(\cdot;\widetilde{G})) = \inf_{\boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})} \left\{ \sum_{n,m} \pi_{nm} c(F(\cdot;\theta_n), F(\cdot;\widetilde{\theta}_m)) - \lambda \mathcal{H}(\boldsymbol{\pi}) \right\}$$

with some regularization parameter $\lambda \geq 0$.

Let the cost function on $\Theta \times \Theta$ be $\widetilde{c}(\theta, \widetilde{\theta}) = c(F(\cdot;\theta), F(\cdot;\widetilde{\theta}))$. Then its implied CTD $\mathcal{T}_{\widetilde{c}}(G, \widetilde{G})$ between two mixtures is also a transportation divergence between two mixing distributions. With many candidates for the cost function $c(\cdot,\cdot)$, there are a large variety of CTDs, we explore the benefit of this flexibility in Chapter 5.

2.4 Barycentre of Probability Measures

Under regular models, a widely adopted aggregation approach in the split-and-conquer approach is to aggregate local estimators by their simple arithmetic mean. The finite mixture model is not regular and the simple arithmetic mean of the local estimators is not a sensible estimator. At the same time, the arithmetic mean is the solution to the least sum of squares problem. Let $x_1, x_2, \ldots, x_n$ be n real vectors, the arithmetic mean $\bar{x} = (x_1 + x_2 + \cdots + x_n)/n$ is the solution to

$$\arg\min_{y} \|x_n - y\|^2.$$

Replacing the quadratic function with a generic distance function, the solution to the least sum of distances resembles the geometric centre. This centre is referred to as barycentre or as Fréchet mean in statistics (Fréchet, 1948).

With this knowledge, one may use the barycentre of local estimators in the probability distribution space as the aggregated estimator (Agueh and Carlier, 2011). This particular approach for finite mixture models is discussed in Section 4.1. The notion of barycentre is utilized more broadly in this thesis. We devote a section to barycentre here for easy reference.

Definition 2.10 (Barycentre of Probability Measures)**.** *Let $(\mathcal{P}(\Theta), \rho)$ be a space of probability measures on Θ that is endowed with the divergence $\rho(\cdot, \cdot)$. Let $(\lambda_1, \lambda_2, \ldots, \lambda_M)$ be a vector of positive values of length M. The (weighted) barycentre of $\nu_1, \ldots, \nu_M \in \mathcal{P}(\Theta)$ is defined to be*

$$\bar{\nu} = \arg\min_{\nu} \sum_{m=1}^{M} \lambda_m \rho(\nu_m, \nu). \tag{2.17}$$

We can choose any sensible divergence $\rho(\cdot, \cdot)$ between two probability measures. Let $\rho(\cdot, \cdot) = W_r^r(\cdot, \cdot)$ with the Euclidean distance as the ground distance, we get the widely used r-Wasserstein barycentre (Cuturi and Doucet, 2014). A barycentre depends on the divergence ρ and the weights, we will omit the information on the divergence and weights in the name but specify them in the formal description.

The barycentre of the Gaussian distributions with respect to some divergence

has either an explicit solution or permits simple numerical solution.

Example 2.3 (Wasserstein Barycentre of Gaussians)**.** *Let ν_m be a Gaussian probability measure with mean vector μ_m and covariance matrix Σ_m for $m \in [M]$. It can be shown that the following matrix equation*

$$\sum_{m=1}^{M} \lambda_m \left(\Sigma^{1/2} \Sigma_m \Sigma^{1/2} \right)^{1/2} = \Sigma \tag{2.18}$$

has a unique positive definite matrix root in Σ (Agueh and Carlier, 2011). Denote this root as $\overline{\Sigma}$ and let $\overline{\mu} = \sum_{m=1}^{M} \lambda_m \mu_m$. Then, when the divergence ρ is chosen to be 2-Wasserstein distance, the barycentre of $\{\nu_m : m \in [M]\}$ is given by the Gaussian distribution with mean vector $\overline{\mu}$ and covariance matrix $\overline{\Sigma}$.

This result shows that the Wasserstein barycentre of Gaussian distributions is also a Gaussian distribution. This is an important property in image processing (Rabin et al., 2011) and colour modification (Solomon et al., 2016). For simplicity, we will simply call the r-Wasserstein barycentre of Gaussian distributions as r-Wasserstein barycentre. The Wasserstein barycentre is also used in large-scale Bayesian inference (Srivastava et al., 2018).

Example 2.4 (KL Barycentre of Gaussians)**.** *Another choice of divergence ρ that leads to simple barycentre solution under Gaussians is the* KL *divergence. Recall that the* KL *divergence between two distributions F_1 and F_2 with density functions $f_1(\cdot)$ and $f_2(\cdot)$ with respect to some measure ν is given by*

$$D_{KL}(F_1 \| F_2) = \int f_1(x) \log(f_1(x)/f_2(x)) \, \nu(dx).$$

The barycentre of (2.17) *with $\rho(F_1, F_2) = D_{KL}(F_1 \| F_2)$ in the space of Gaussian distributions, is a Gaussian distribution with mean vector $\overline{\mu} = \sum_{m=1}^{M} \lambda_m \mu_m$ and covariance matrix*

$$\overline{\Sigma} = \sum_{m=1}^{M} \lambda_m (\Sigma_m + (\mu_m - \bar{\mu})(\mu_m - \bar{\mu})^\top).$$

We offer a proof of this claim in Appendix C.2. For simplicity, we simply call

the barycentre based on KL divergence of Gaussian distributions as KL barycentre. If not confined in the space of Gaussian distributions, the minimizer to (2.17) would be the mixture $\sum_{m=1}^{M} \lambda_m \Phi(x; \mu_m, \Sigma_m)$ itself.

Figure 2.2 depicts the covariance matrices of the (Gaussian) barycentres of four 2-dimensional Gaussian measures arranged by $\boldsymbol{\lambda}$ values.

2.5 Performance Metrics in Experiments

We use same performance metrics in the experiments in Chapter 3 – Chapter 5. To avoid introducing the same notation repeatedly, we summarize the performance metrics in this section.

Under mixture models, the degree of overlap between subpopulations decides how difficult it is to learn the model. The larger the degree of overlap, the more difficult it is to learn the model. We give the formal definition of pairwise degree of overlap as follows.

Definition 2.11 (Pairwise Degree of Overlap)**.** *Let $f(x; G) = \sum_{k=1}^{K} w_k f(x; \theta_k)$ be the density function of a mixture of order K. The probability of a unit from subpopulation i misclassified as a unit in subpopulation j by the maximum posterior rule (in some sense) is*

$$o_{j|i} = \mathbb{P}\big(w_i f(X; \theta_i) < w_j f(X; \theta_j) | X \sim f(x; \theta_i)\big).$$

The degree of overlap between the ith and the jth subpopulations is defined as

$$o_{ij} = o_{j|i} + o_{i|j}. \tag{2.19}$$

In our simulation study, we pick various values for the degree of overlap of the population mixture. Consider an experiment with R repetitions, let $\mathcal{X}^{(r)}$ be a random sample from $F^*(x; G^{(r)})$ for $r \in [R]$. Let $F(x; \widehat{G}^{(r)})$ be the estimated mixture based on $\mathcal{X}^{(r)}$. We usually confine our study to the situation where F^* and F are from the same family. When study the robustness of an estimator, F^* and F may from different families. For example, $F^*(\cdot; G)$ may be a logistic mixture and $F(x; \widehat{G})$ can be a Gaussian mixture in this situation.

How to measure the performance of an estimator based on the outputs of a

(a) 2-Wasserstein barycentre

(b) KL barycentre

Figure 2.2: The covariance matrices of (a) Wasserstein barycentres and (b) KL barycentres of 4 randomly generated zero-mean 2-dimensional Gaussian measures arranged by the $\boldsymbol{\lambda}$ value. The four corners are those obtained with $\boldsymbol{\lambda} = (1,0,0,0)^\top, (0,1,0,0)^\top, (0,0,1,0)^\top, (0,0,0,1)^\top$. The middle one is obtained with $\boldsymbol{\lambda} = (1/4, 1/4, 1/4, 1/4)^\top$.

simulation study? For vector-valued parameters, the commonly used performance metric of their estimators is the MSE. Under the finite mixture model that is non-regular, the parameter is a discrete distribution and is called the mixing distribution. One or more of the following quantities are used to measure the performance of the various estimators.

1. **Integrated Squared Error (ISE) between Mixtures**. When F and F^* are from the same family, we measure the performance of the learned mixing distribution by the value of $D_{\text{ISE}}(F^*(\cdot; G^{(r)}), F(\cdot; \widehat{G}^{(r)}))$. The better the learned mixing distribution, the closer the value to 0.

2. **Distance between Mixing Distributions (W1)**. The transportation divergence between the learned and true mixing distributions can also be used to measure the performance of different estimators. The closer the value of the transportation divergence to 0, the better the performance of the estimator. Under GMM, we use

$$W_1(\widehat{G}^{(r)}, G^{(r)}) = \inf \left\{ \sum_{nm} \pi_{nm} D(\widehat{\theta}_n, \theta_m) : \boldsymbol{\pi} \in \Pi(\hat{\boldsymbol{w}}, \boldsymbol{w}) \right\}$$

   where $\widehat{\theta}_n$ and θ_m are subpopulations parameters in $\widehat{G}^{(r)}$ and $G^{(r)}$ respectively. The mixing weights of $\widehat{G}^{(r)}$ and $G^{(r)}$ are $\hat{\boldsymbol{w}}$ and $\boldsymbol{w}$ respectively. The ground cost function $D(\widehat{\theta}_n, \theta_m)$ is chosen to be the one in (2.15). Since the cost function is a distance and $r = 1$, this divergence is Wasserstein distance $W_{c,1}$. For simplicity of notation, we denote this divergence as W1.

3. **Adjusted Rand Index (ARI)**. According to Section 2.1, the finite mixture models are often used for clustering. Given an observed value x from the true mixture population $f^*(\cdot; G^*)$, according to (2.1), the observation is classified into the cluster $\kappa_*(x) = \arg\max_{j \in [K]} \{w_j^* f^*(x; \theta_j^*)\}$ based on the true mixture. Similarly, if $f(\cdot; \widehat{G})$ is the learned mixture, then the most likely cluster that x belongs to is $\hat{\kappa}(x) = \arg\max_{j \in [K]} \{\widehat{w}_j f(x; \widehat{\theta}_j)\}$. We measure the performance of the learning approach by measuring the degree of similarity between $\{\hat{\kappa}^{(r)}(x_i) : x_i \in \mathcal{X}^{(r)}\}$ and $\{\kappa_*^{(r)}(x_i) : x_i \in \mathcal{X}^{(r)}\}$ by the ARI given in Definition 2.1.

4. **Log-likelihood (LL)**. We compare the value of the log-likelihood (LL) function at the estimated mixing distribution. For the ease of presentation, we present the value of the log-likelihood per observation. Let $\widehat{G}^{(r)}$ be an estimator of true mixing distribution $G^{(r)}$ in the rth experiment based on an IID sample $\mathcal{X}^{(r)} = \{x_1^{(r)}, x_2^{(r)}, \ldots, x_N^{(r)}\}$, then the LL of estimator $\widehat{G}^{(r)}$ is defined to be

$$\mathrm{LL}(\widehat{G}^{(r)}) = N^{-1} \sum_{n=1}^{N} \log f(x_i^{(r)}; \widehat{G}^{(r)}).$$

   We can similarly define the LL of the true mixing distribution $G^{(r)}$. The higher the value of LL, the better the performance of the estimator.

For all performance metrics, we either report the value of each metric averaged across R repetitions or present the boxplot of their values across R repetitions. We specify the details in each simulation.

Chapter 3

Minimum Wasserstein Distance Estimator under Univariate Finite Location-Scale Mixture

For finite mixture models in (1.1), the most fundamental inference problem is the learning of the mixing distribution G based on data. In this chapter, we study the problem of learning the mixing distribution G given a set of IID univariate observations from a finite location-scale mixture, that is a mixture when $\mathcal{F}$ is a known location-scale family. The location-scale family $\mathcal{F}$ consists of densities

$$f(x;\boldsymbol{\theta}) = \frac{1}{\sigma} f_0\Big(\frac{x-\mu}{\sigma}\Big)$$

for some probability density function $f_0(x)$ with $x \in \mathbb{R}$ with respect to Lebesgue measure where $\boldsymbol{\theta} = (\mu, \sigma)^\top$ with $\Theta = \mathbb{R} \times (0, \infty)$.

In statistics, the Maximum Likelihood Estimate (MLE) is usually the first choice to learn model parameters due to its statistical efficiency. However, under finite location-scale mixture model, the MLE of G is not well-defined. The log-likelihood function of G based on a set of IID observations $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ from a fi-

nite location-scale mixture model is given by

$$\ell_N(G|\mathcal{X}) = \sum_{n=1}^{N} \log f(x_n; G) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \frac{w_k}{\sigma_k} f_0 \left(\frac{x_n - \mu_k}{\sigma_k} \right) \right\}.$$

This log-likelihood function is unbounded: its value goes to infinity for a specific combination of μ_k and some $\sigma_k \to 0$. Hence, the MLE of G is not well-defined or is ill defined as explained in Section 1.2.1.

The minimum distance estimator is one of many alternatives to MLE (Blum and Susarla, 1977; Choi, 1969; Choi and Bulgren, 1968; Clarke and Heathcote, 1994; Cutler and Cordero-Brana, 1996; Macdonald, 1971). A minimum distance estimator resembles the MLE in a way as the MLE minimizes the Kullback-Leibler (KL) divergence between the empirical distribution and the assumed model (Eguchi and Copas, 2006). Suppose we have a set of IID observations $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Let $F_N(x) = N^{-1} \sum_{n=1}^{N} \mathbb{1}(x_n \leq x)$ be the empirical distribution. Given a distance $D(\cdot, \cdot)$ on the space of cumulative distribution functions, a minimum distance estimator under a finite mixture model of order K is defined to be

$$\widehat{G}_N = \underset{G \in \mathbb{G}_K}{\arg\min}\, D(F_N(\cdot), F(\cdot; G)).$$

Note in the above notation in the distance, we denote by dot the input of the Cumulative Distribution Function (CDF)s to address that the distance is defined between two functions $F_N(\cdot)$ and $F(\cdot; G)$, rather than two values $F_N(x)$ and $F(x; G)$. Table 1.2 in Chapter 1 lists the distances and the corresponding minimum distance estimators studied under finite mixture models that we are aware of. Noticeably, the Wasserstein distance is not one of them.

Given the increased interest of Wasserstein distance in the machine learning community, we wish to know whether the Minimum Wasserstein Distance Estimator (MWDE) is a viable approach to learn finite location-scale mixtures. We answer the following questions in this chapter.

1. Is the MWDE well-defined under finite location-scale mixtures?

2. Is the MWDE a consistent estimator?

3. Is the MWDE more efficient than the penalized Maximum Likelihood Estimate (PMLE)?

4. Is the MWDE more robust than the PMLE?

We find that the MWDE is consistent, and we develop a numerical solution under finite location-scale mixtures. We compare the robustness of the MWDE with the PMLE in the presence of outliers and mild model mis-specifications. We conclude that the MWDE suffers some efficiency loss against the PMLE in general without an obvious gain in robustness. These findings reaffirm the general superiority of the likelihood-based learning strategies even for the non-regular finite location-scale mixtures.

This chapter is organized as follows. In Section 3.1, we give the formal definition of the MWDE and discuss its existence and consistency under finite location-scale mixtures. We obtain some algebraic results that are essential for computing the 2-Wasserstein distance between the empirical distribution and a finite location-scale mixture. We then develop a Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm scheme for computing the MWDE of the mixing distribution. In Section 3.2, we characterize the efficiency properties of the MWDE relative to PMLE in various circumstances via simulations. We also study the robustness of MWDE when the data contains outliers, is contaminated, or when the model is mis-specified. We then apply both methods in an image segmentation example in Section 3.3. We conclude this chapter with a summary in Section 3.4.

3.1 Minimum Wasserstein Distance Estimator (MWDE)

In this section, we introduce the MWDE estimator under finite location-scale mixtures. We also investigate its existence, statistical consistency, and numerical computation.

Let $W_r(\cdot,\cdot)$ be the r-Wasserstein distance between univariate random variables with ground distance $D(x,y) = \|x-y\|$. Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be a set of IID observations from a finite location-scale mixture $f(x;G)$ of order K and $F_N(x) = N^{-1}\sum_{n=1}^{N} \mathbb{1}(x_n \leq x)$ be the empirical distribution. Assume $f_0(x)$ has finite rth moment for some $r \geq 1$, we then propose the MWDE of the mixing

distribution G:

$$\widehat{G}_N^{\text{MWDE}} = \underset{G\in\mathbb{G}_K}{\arg\inf}\, W_r(F_N(\cdot), F(\cdot;G)) = \underset{G\in\mathbb{G}_K}{\arg\inf}\, W_r^r(F_N(\cdot), F(\cdot;G)). \quad (3.1)$$

3.1.1 Existence of MWDE

As we pointed out earlier, the MLE is not well-defined under finite location-scale mixtures. Is the MWDE well-defined? We examine the existence or sensibility of the MWDE in this section. We show that the MWDE exists when $f_0(\cdot)$ satisfies certain conditions.

Assume that $f_0(0) > 0$, $f_0(x)$ is bounded, continuous, and has finite rth moment for some $r \geq 1$. Under these conditions, we can see that

$$0 \leq W_r(F_N(\cdot), F(\cdot;G)) < \infty$$

for any $G \in \mathbb{G}_K$. When $N \leq K$, the solution to (3.1) requires special attention. Let $G_\epsilon = N^{-1}\sum_{n=1}^N \delta_{(x_n,\epsilon)}$ be a mixing distribution that assigns probability $1/N$ to $(x_n, \epsilon)^\top$. When $\epsilon \to 0$, each subpopulation in the mixture $F(x; G_\epsilon)$ degenerates to a point mass at x_n and the mixture $F(x; G_\epsilon) \to F_N(x)$. Hence, as $\epsilon \to 0$, we have

$$W_r(F_N(\cdot), F(\cdot; G_\epsilon)) \to 0.$$

Since none of $G \in \mathbb{G}_K$ has zero-distance from $F_N(\cdot)$, the MWDE does not exist unless we expand $\mathbb{G}_K$ to include $G_0 = N^{-1}\sum_{n=1}^N \delta_{(x_n,0)} = \lim_{\epsilon\to 0} G_\epsilon$. To remove this technical artifact, in Definition 3.1 of MWDE, we expand the space of σ to $[0, \infty)$. We denote by $F(\cdot; (\theta_0, 0)^\top)$ a distribution with point mass at $x = \theta_0$. With this expansion, G_0 is the MWDE when $N \leq K$.

In the next few paragraphs, we exhaust all possible ways that MWDE is not consistent and demonstrate each of them leads to a contradiction. By excluding all these possibilities, we conclude that MWDE is consistent.

Let $\delta = \inf\{W_r(F_N(\cdot), F(\cdot;G)) : G \in \mathbb{G}_K\}$. Clearly, $0 \leq \delta < \infty$. By definition, there exists a sequence of mixing distributions $\{G_m, m = 1, 2, \ldots\} \in \mathbb{G}_K$ such that $W_r(F_N(\cdot), F(\cdot; G_m)) \to \delta$ as $m \to \infty$. Suppose one mixing weight

of G_m has a limit 0. Then remove the corresponding support point and rescale the mixing weights to get a proper distribution, we obtain a new mixing distribution sequence and it still satisfies $W_r(F_N(\cdot), F(\cdot; G_m)) \to \delta$. Therefore, we assume that the mixing weights of G_m have non-zero limits by selecting converging subsequence if necessary to ensure the limits exist. Further, if we keep the same support points as that of G_m but replace the mixing weights with their limits, we still have $W_r(F_N(\cdot), F(\cdot; G_m)) \to \delta$ as $m \to \infty$. We hence discuss the sequence of mixing distributions whose mixing weights are fixed in the following discussion.

Suppose the first support point of G_m has its scale parameter $\sigma_1 \to \infty$ as $m \to \infty$. With the boundedness assumption on $f_0(x)$, the mass of this subpopulation will spread thinly over entire $\mathbb{R}$ because $\sigma_1^{-1} f_0((x - \mu_1)/\sigma_1) \to 0$ uniformly. Denote $\boldsymbol{\theta}_1 = (\mu_1, \sigma_1)$. For any fixed finite interval $[a, b]$, this thinning makes

$$F(b; \boldsymbol{\theta}_1) - F(a; \boldsymbol{\theta}_1) \to 0$$

as $m \to \infty$. It implies that for any given $t \in (0, 0.5)$, we have

$$|F^{-1}(t; \boldsymbol{\theta}_1)| + |F^{-1}(1 - t; \boldsymbol{\theta}_1)| \to \infty.$$

This further implies that for any $t \in (0, w_1/2)$ we have

$$|F^{-1}(t; G_m)| + |F^{-1}(1 - t; G_m)| \to \infty$$

where w_1 is the mixing weight corresponding to subpopulation with parameter $\boldsymbol{\theta}_1$ as $m \to \infty$. In comparison, the empirical quantile satisfies $x_{(1)} \leq F_N^{-1}(t) \leq x_{(N)}$ for any t. By the form of $W_r(\cdot, \cdot)$ in (2.14), this leads to $W_r(F_N(\cdot), F(\cdot; G_m)) \to \infty$ as $m \to \infty$. This contradicts with the assumption that $W_r(F_N(\cdot), F(\cdot; G_m)) \to \delta$. Hence, neither $\sigma_1 \to \infty$ nor $\sigma_k \to \infty$ for any k is a possible scenario of G_m.

Can a support point of G_m instead have its location parameter $\mu \to \infty$? Let the parameter of this support point corresponds to $\boldsymbol{\theta}_1$. Note that at least $w_1\{1 - F_0(0)\}$-sized probability mass of $F(x; G_m)$ is contained in the range $[\mu_1, \infty)$. Because of this, when $\mu_1 \to \infty$, we have $F^{-1}(1 - t; G_m) \to \infty$ for $t = w_1\{1 - F_0(0)\}/2$. Therefore, $W_r(F_N(\cdot), F(\cdot; G_m)) \to \infty$ by (2.14). This contradicts with the fact that $W_r(F_N(\cdot), F(\cdot; G_m)) \to \delta < \infty$. Hence, $\mu_1 \to \infty$ is not a possible scenario

of G_m either. For the same reason, we cannot have $\mu_k \to \pm\infty$ for any k.

After ruling out $\mu_k \pm \infty$ and $\sigma_k \to \infty$, we find G_m has a converging subsequence whose limit is a proper mixing distribution in $\mathbb{G}_K$. This limit is then an MWDE and its existence is verified.

The MWDE may not be unique and the mixing distribution may lead to a mixture with degenerated subpopulations when N is small. We will show that the MWDE is consistent as the sample size goes to infinity. Thus, allowing degenerated subpopulations in the learned mixture is a mathematical artifact for rigorous proof. In contrast, no matter how large the sample size becomes, there are always nonsensical degenerated mixing distributions with unbounded likelihood values.

3.1.2 Statistical Consistency of MWDE

In this section, we establish the consistency of MWDE when $\mathcal{X} = \{x_1, \ldots, x_N\}$ are IID observations from a finite location-scale mixture of order K. The true mixing distribution is denoted as G^*. Assume that $f_0(x)$ is bounded, continuous, and has finite rth moment. We say the location-scale mixture, or any mixture model in general, is identifiable if

$$F(x; G_1) = F(x; G_2)$$

for all x given $G_1, G_2 \in \mathbb{G}_K$ implies $G_1 = G_2$. We allow subpopulation scale parameter $\sigma = 0$. The most commonly used finite locate-scale mixtures, such as the normal (univariate Gaussian) mixture, are well-known to be identifiable (Teicher, 1961). Holzmann et al. (2004) give a sufficient condition for the identifiability of general finite location-scale mixtures. Let $\varphi(\cdot)$ be the characteristic function of $f_0(t)$. The finite location-scale mixture is identifiable if for any $\sigma_1 > \sigma_2 > 0$, $\lim_{t\to\infty} \varphi(\sigma_1 t)/\varphi(\sigma_2 t) = 0$.

We consider the MWDE based on r-Wasserstein distance with ground distance $D(x, y) = |x - y|$ for some $r \geq 1$. For the same r, we assume that $f_0(x)$ has finite rth moment. We show that the MWDE under finite location-scale mixture model as defined in (3.1) is asymptotically consistent.

Theorem 3.1. *With the same notations above, assume that $f_0(\cdot)$ is bounded, continuous, and has finite rth moment, we have the following conclusions.*

1. *For any sequence* $G_m \in \mathbb{G}_K$ *and* $G^* \in \mathbb{G}_K$, $W_r(F(\cdot; G_m), F(\cdot; G^*)) \to 0$ *implies* $G_m \xrightarrow{d} G^*$ *as* $m \to \infty$.

2. *The* MWDE *satisfies* $W_r\left(F(\cdot; G^*), F(\cdot; \widehat{G}_N^{\text{MWDE}})\right) \to 0$ *as* $N \to \infty$ *almost surely.*

3. *The* MWDE *is consistent:* $W_r\left(\widehat{G}_N^{\text{MWDE}}, G^*\right) \to 0$ *as* $N \to \infty$ *almost surely.*

Proof. We present these three conclusions in the current order which is easy to understand. A different order is better for proof. For ease of presentation, we write $F^* = F(\cdot; G^*)$ and $\widehat{G} = \widehat{G}_N^{\text{MWDE}}$ in this proof.

We first prove the second conclusion. By the triangular inequality and the definition of MWDE, we have

$$W_r\left(F^*, F(\cdot; \widehat{G}_N)\right) \leq W_r(F_N, F^*) + W_r\left(F_N, F(\cdot; \widehat{G}_N)\right) \leq 2W_r(F_N, F^*).$$

Note that F_N is the empirical distribution and F^* is the true distribution, we have $F_N(x) \to F^*(x)$ uniformly in x almost surely by the Glivenko-Cantelli uniform Law of large numbers (Van der Vaart, 2000, Chapter 19). At the same time, under the assumption that $F_0(x)$ has finite rth moment, $F^*(x)$ also has finite rth moment. The rth moment of $F_N(x)$ converges to that of $F^*(x)$ almost surely by the law of large numbers. Given the ground distance $D(x, y) = |x - y|$, the rth moment in Wasserstein distance sense is the usual moments in probability theory. By Lemma 2.2, we conclude $W_r(F_N, F(\cdot; G^*)) \to 0$ as both conditions are satisfied.

Conclusion 3 is implied by Conclusions 1 and 2. With Conclusion 2 already established, we need only prove Conclusion 1 to complete the whole proof. By Helly's lemma (Van der Vaart, 2000, Lemma 2.5) again, G_m has a converging subsequence though the limit can be a sub-probability measure. Without loss of generality, we assume that G_m itself converges with limit $\widetilde{G}$. If $\widetilde{G}$ is a sub-probability measure, so would be $F(\cdot; \widetilde{G})$. This will lead to

$$W_r(F(\cdot; G_m), F(\cdot; G^*)) \to W_r\left(F(\cdot; \widetilde{G}), F(\cdot; G^*)\right) \neq 0$$

which violates the theorem condition. If $\widetilde{G}$ is a proper distribution in $\mathbb{G}_K$ and

$$W_r\left(F(\cdot;\widetilde{G}), F(\cdot;G^*)\right) = 0,$$

then by identifiability condition, we have $\widetilde{G} = G^*$. This implies $G_m \to G^*$ and completes the proof. □

The multivariate Gaussian mixture is another type of location-scale mixture. The above consistency result of the MWDE can be easily extended to finite multivariate Gaussian mixtures.

Theorem 3.2. *Consider the problem when $\mathcal{X} = \{x_1, \ldots, x_N\}$ are* IID *observations from a finite Gaussian mixture distribution of order K and $\widehat{G}_N^{\text{MWDE}}$ is the* MWDE *defined by* (3.1)*. Let the true mixing distribution be G^*. The* MWDE *is consistent: $W_r\left(\widehat{G}_N^{\text{MWDE}}, G^*\right) \to 0$ as $N \to \infty$ almost surely.*

The rigorous proof is long though the conclusion is obvious. We offer a less formal proof based on several well-known probability theory results:

(I) A multivariate random variable sequence Y_n converges in distribution to Y if and only if $\mathbf{a}^\top Y_n$ converges to $\mathbf{a}^\top Y$ for any unit vector $\mathbf{a}$;

(II) If Y is multivariate Gaussian if and only if $\mathbf{a}^\top Y$ is normal for all $\mathbf{a}$;

(III) The normal distribution has finite moment of any order.

Let X_m be a random vector with distribution $F(x; G_m)$ for some $G_m \in \mathbb{G}_K$, $m = 0, 1, 2, \ldots$, in a general mixture model setting. Suppose as $m \to \infty$, with the notations we introduced previously,

$$W_r(X_m, X_0) \to 0.$$

Then for any unit vector $\mathbf{a}$, based on Lemma 2.2 of the Wasserstein distance and the result (I), we can see that

$$W_r\left(\mathbf{a}^\top X_m, \mathbf{a}^\top X_0\right) \to 0.$$

Next, we apply this result to normal mixture so that $F(x; G_m)$ becomes $\Phi(x; G_m)$ which stands for a finite multivariate normal mixture with mixing distribution G_m. In this case, X_m is a random vector with distribution $\Phi(x; G_m)$. Let (μ_k, Σ_k) be generic subpopulation parameters. We can see that the distribution of $\mathbf{a}^\top X_m$ is $\Phi_{\mathbf{a}}(x; \widetilde{G}_m)$, which is a finite normal mixture with subpopulation parameters $(\mathbf{a}^\top \mu_k, \mathbf{a}^\top \Sigma_k \mathbf{a})$, and mixing weights the same as those of G_m. Let the mixing distributions after projection be $\widetilde{G}_{m,\mathbf{a}}$ and $\widetilde{G}_{0,\mathbf{a}}$.

By the same argument in the proof of Theorem 3.1,

$$W_r\left(\Phi(\cdot; \widehat{G}_N^{\text{WMDE}}), \Phi(\cdot; G^*)\right) \to 0$$

almost surely as $N \to \infty$. This implies

$$W_r\left(\Phi_{\mathbf{a}}(\cdot; \widehat{G}_N^{\text{WMDE}}), \Phi_{\mathbf{a}}(\cdot; G^*)\right) \to 0$$

almost surely as $N \to \infty$ for any $\mathbf{a}$. Let $\widehat{G}_{N,\mathbf{a}}^{\text{WMDE}}$ and $\widehat{G}_{\mathbf{a}}^*$ be the projections of $\widehat{G}_N^{\text{MWDE}}$ and G^* in direction $\mathbf{a}$ respectively. By Conclusion 1 of Theorem 3.1, $\widehat{G}_{N,\mathbf{a}}^{\text{WMDE}} \xrightarrow{d} \widehat{G}_{\mathbf{a}}^*$ almost surely for any unit vector $\mathbf{a}$. We therefore conclude the consistency result: $\widehat{G}_N^{\text{WMDE}} \xrightarrow{d} G^*$ almost surely.

3.1.3 Numerical Computation of MWDE

Both in applications and in simulation experiments, we need an effective way to compute the MWDE. We develop an algorithm that leverages the explicit form of the Wasserstein distance between two measures on $\mathbb{R}$ for the numerical solution to the MWDE. The strategy works for any r-Wasserstein distance but we only provide specifics for $r = 2$ as it is the most widely used. We leave the algebraic details in the Appendix A. For the rest of this chapter, we only discuss the univariate location-scale mixture due to its computational simplicity.

Let Y be a random variable with distribution $f_0(\cdot)$. Denote the mean and variance of Y by $\mu_0 = \mathbb{E}(Y)$ and $\sigma_0^2 = \text{Var}(Y)$. Recall that $G = \sum_{k=1}^K w_k \delta_{(\mu_k, \sigma_k)}$. Let $x_{(1)} \le x_{(2)} \le \cdots \le x_{(N)}$ be the order statistics, $\overline{x^2} = N^{-1} \sum_{n=1}^N x_n^2$, and $\xi_n = F^{-1}(n/N; G)$ be the (n/N)th quantile of the mixture for $n = 0, 1, \ldots, N$.

Let

$$T(x) = \int_{-\infty}^{x} t f_0(t)\, dt \tag{3.2}$$

and define

$$\Delta F_{nk} = F_0\left(\frac{\xi_n - \mu_k}{\sigma_k}\right) - F_0\left(\frac{\xi_{n-1} - \mu_k}{\sigma_k}\right),$$
$$\Delta T_{nk} = T\left(\frac{\xi_n - \mu_k}{\sigma_k}\right) - T\left(\frac{\xi_{n-1} - \mu_k}{\sigma_k}\right).$$

When $r = 2$, we expand the squared W_2 distance, $\mathbb{W}_N$, between the empirical distribution and $F(\cdot; G)$ as follows:

$$\begin{aligned}
\mathbb{W}_N(G) =& W_2^2(F_N(\cdot), F(\cdot; G)) \\
=& \int_0^1 \{F_N^{-1}(t) - F^{-1}(t; G)\}^2\, dt \\
=& \overline{x^2} + \sum_{k=1}^{K} w_k\{\mu_k^2 + \sigma_k^2(\mu_0^2 + \sigma_0^2) + 2\mu_k\sigma_k\mu_0\} \\
& - 2\sum_k w_k \left\{\mu_k \sum_{n=1}^{N} x_{(n)}\Delta F_{nk} + \sigma_k \sum_{n=1}^{N} x_{(n)}\Delta T_{nk}\right\}.
\end{aligned}$$

The MWDE minimizes $\mathbb{W}_N(G)$ with respect to G. The mixing weights and subpopulation scale parameters in this optimization problem have natural constraints. We may replace the optimization problem with an unconstrained one by the following parameter transformation:

$$\sigma_k = \exp(\tau_k),$$
$$w_k = \frac{\exp(t_k)}{\sum_{j=1}^{K} \exp(t_j)}$$

for $k \in [K]$. We may then minimize $\mathbb{W}_N$ with respect to $\{(\mu_k, \tau_k, t_k) : k \in [K]\}$ over the unconstrained space $\mathbb{R}^{3K}$. Furthermore, we adopt the quasi-Newton BFGS algorithm (Nocedal and Wright, 2006, Section 6.1). To use this algorithm, it is best

to provide the gradients of $\mathbb{W}_N(G)$, which are given as follows:

$$\begin{aligned}
\frac{\partial \mathbb{W}_N}{\partial t_j} &= \sum_{k=1}^{K} \left\{ \frac{\partial w_k}{\partial t_j} \frac{\partial \mathbb{W}_N}{\partial w_k} \right\} = \sum_k w_j(\delta_{jk} - w_k) \frac{\partial \mathbb{W}_N}{\partial w_k}, \\
\frac{\partial \mathbb{W}_N}{\partial \mu_j} &= 2w_j \left\{ \mu_j + \sigma_j \mu_0 - \sum_{n=1}^{N} x_{(n)} \Delta F_{nj} \right\}, \\
\frac{\partial \mathbb{W}_N}{\partial \tau_j} &= 2w_j \left\{ \sigma_j(\mu_0^2 + \sigma_0^2) + \mu_j \mu_0 - \sum_{n=1}^{N} x_{(n)} \Delta T_{nj} \right\} \frac{\partial \sigma_j}{\partial \tau_j}
\end{aligned}$$

for $j \in [K]$, where

$$\begin{aligned}
\frac{\partial \mathbb{W}_N}{\partial w_k} = & \{\mu_k^2 + \sigma_k^2(\mu_0^2 + \sigma_0^2) + 2\mu_k \sigma_k \mu_0\} \\
& - 2 \sum_{n=1}^{N-1} \{x_{(n+1)} - x_{(n)}\} \xi_n F(\xi_n; \mu_k, \sigma_k) \\
& - 2 \left\{ \mu_k \sum_{n=1}^{N} x_{(n)} \Delta F_{nk} + \sigma_k \sum_{n=1}^{N} x_{(n)} \Delta T_{nk} \right\}.
\end{aligned}$$

Since $\mathbb{W}_N(G)$ is non-convex, the algorithm may find a local minimum of $\mathbb{W}_N(G)$ instead of a global minimum as required for MWDE. We use multiple initial values for the BFGS algorithm, and regard the one with the lowest $\mathbb{W}_N(G)$ value as the solution.

This algorithm involves computing the quantiles ξ_n and ΔT_{nj} repeatedly which may lead to high computational cost. Since ξ_n is between $\min_k F^{-1}(n/N; \boldsymbol{\theta}_k)$ and $\max_k F^{-1}(n/N; \boldsymbol{\theta}_k)$, it can be found efficiently via a bisection method, see Appendix A for details. Fortunately, $T(x)$ in (3.2) has simple analytical forms under two widely used location-scale mixtures which make the computation of ΔT_{nj} efficient:

1. When $f_0(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ which is the density function of the standard normal, we have $x f_0(x) = -f_0'(x)$. In this case, we find

$$T(x) = -f_0(x).$$

2. For finite mixture of location-scale logistic distributions, we have

$$f_0(x) = \frac{\exp(-x)}{(1+\exp(-x))^2}$$

and

$$T(x) = \int_{-\infty}^{x} t f_0(t)\, dt = \frac{x}{1+\exp(-x)} - \log(1+\exp(x)). \tag{3.3}$$

3.2 Simulation

We now study the performance of the MWDE and the PMLE under finite location-scale mixtures. We explore the potential advantages of the MWDE and quantify its efficiency loss, if any, by simulation experiments. Consider the following three location-scale families (Chen et al., 2020):

1. Normal distribution: $f_0(x) = (2\pi)^{-1/2}\exp(-x^2/2)$. Its mean and variance are given by $\mu_0 = 0$ and $\sigma_0^2 = 1$.
2. Logistic distribution: $f_0(x) = \exp(-x)/(1+\exp(-x))^2$. Its mean and variance are given by $\mu_0 = 0$ and $\sigma_0^2 = \pi^2/3$.
3. Gumbel distribution (type I extreme-value distribution): $f_0(x) = \exp(-x - \exp(-x))$. Its mean and variance are given by $\mu_0 = \gamma$ and $\sigma_0^2 = \pi^2/6$ where γ is the Euler constant (Gumbel, 1954, Table 3.1).

We also include a real data example to compare the image segmentation result of using the MWDE and PMLE.

3.2.1 Homogeneous Model

The homogeneous location-scale model is a special mixture model with a single subpopulation $K = 1$. Both the MWDE and the MLE are applicable for parameter estimation. There have been no studies of MWDE in this special case in the literature. It is therefore of our interest to see how MWDE performs under this model.

Under three location-scale models given earlier, the MWDE has closed-forms. Using the same notations, their analytical forms are as follows.

1. Normal distribution:

$$\widehat{\mu}^{\text{MWDE}} = \bar{x},\ \widehat{\sigma}^{\text{MWDE}} = \sum_{n=1}^{N} x_{(n)} \left\{ f_0(\xi_{n-1}) - f_0(\xi_n) \right\}.$$

2. Logistic distribution:

$$\widehat{\mu}^{\text{MWDE}} = \bar{x},\ \widehat{\sigma}^{\text{MWDE}} = \frac{3}{\pi^2} \sum_{n=1}^{N} x_{(n)} \left\{ T(\xi_n) - T(\xi_{n-1}) \right\}$$

   where $T(x)$ is given in (3.3).

3. Gumbel distribution:

$$\widehat{\mu}^{\text{MWDE}} = \{1 - \gamma r\}^{-1} \{\bar{x} - \gamma T\},\ \widehat{\sigma}^{\text{MWDE}} = T - r\widehat{\mu}^{\text{MWDE}}$$

   where

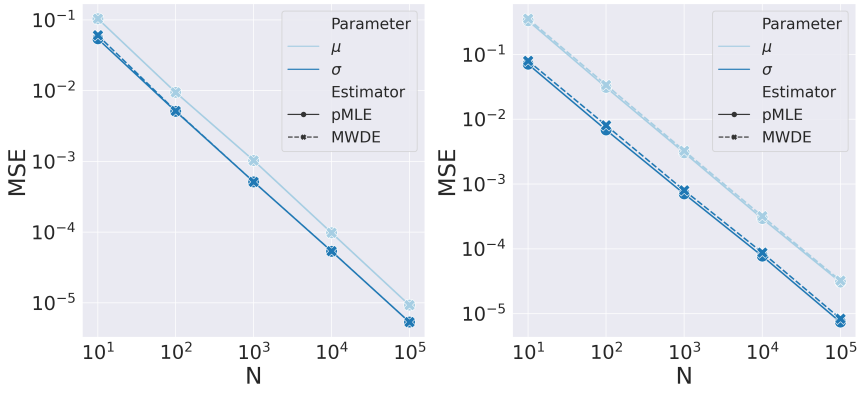
$$T = \{\gamma^2 + \pi^2/6\}^{-1} \sum_{n=1}^{N} x_{(n)} \int_{\xi_{n-1}}^{\xi_n} t f_0(t)\, dt$$

   and $r = \gamma/(\gamma^2 + \pi^2/6)$.

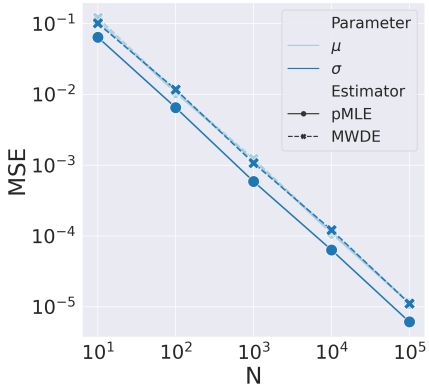
The MLEs under the logistic and Gumbel distributions do not have an easy-to-use analytical form, we therefore employ BFGS to solve for MLE. We generate samples of sizes between $N = 10$ to $N = 100000$ with $R = 1000$ repetitions. Under the homogeneous model, it is most convenient to compute the Mean Squared Error (MSE) of the location and scale parameters separately. Due to invariance property, we only need to study the problem when the data are from distributions with $\mu = 0$ and $\sigma = 1$.

The simulation results are summarized as plots in Figure 3.1. Both the x and y axes in these plots are in logarithm scale. For both MLE and MWDE, their log-MSE and $\log(N)$ values are close to the straight lines with slope -1. This phenomenon indicates that both estimators have the expected convergence rates $O_p(N^{-1/2})$ as the sample size $N \to \infty$.

The performance of the estimators for the location parameter and scale parameter are different. For the location parameter under all three models, the lines



(a) Normal **(b)** Logistic



(c) Gumbel

Figure 3.1: The MSEs of the MWDE and the MLE for location and scale parameters versus sample size N under different homogeneous (a) normal distribution, (b) logistic distribution, and (c) Gumbel distribution.

formed by MLE and MWDE are nearly indistinguishable though the MSE line of MLE is always below the that of MWDE. For the scale parameter σ, the MLE is also more efficient than the MWDE but the difference is negligible under the normal and logistic models. Under the Gumbel model, the MWDE is less efficient.

In summary, using MWDE under a homogeneous model may not be preferred but appear to be acceptable under the normal and logistic models. We do not investigate the performance of MWDE under Gumbel mixture further due to its

efficiency loss under the homogeneous model. With these observations, we move to its performance under finite location-scale mixtures.

3.2.2 Efficiency and Robustness

We study the efficiency and robustness of the MWDE for learning finite location-scale mixtures. Since the MLE is not well-defined, we compare the performance of MWDE with the PMLE (Chen and Tan, 2009) instead. We compare their performances when the mixture model is correctly specified, when the data is contaminated, or when the model is mildly misspecified.

Efficiency

A widely employed two-component mixture model (Cutler and Cordero-Brana, 1996; Zhu, 2016) has a density function in the following form:

$$f(x;G) = pf(x;0,a) + (1-p)f(x;b,1) \tag{3.4}$$

with some density function $f(\cdot;\boldsymbol{\theta})$ from a location-scale family. Namely, we have $K = 2$, the mixing weights $w_1 = p, w_2 = 1-p$, and subpopulation parameters $\boldsymbol{\theta}_1 = (0,a)^\top$ and $\boldsymbol{\theta}_2 = (b,1)^\top$. By choosing different combinations of p, a, and b, we obtain mixtures with different properties. Due to the invariance property, we need only consider the case where one of the location parameters is 0, and one of the scale parameter is 1.

We generate samples from $f(x;G)$ according to the following scheme: generate an observation Y from distribution with density function $f_0(x)$ and let

$$X = \begin{cases} aY, & \text{with probability } p; \\ Y + b, & \text{otherwise.} \end{cases} \tag{3.5}$$

We can easily see that the distribution of X is $f(x;G)$ specified earlier.

The level of difficulty in precisely estimating the mixture largely depends on the degree of overlap between the subpopulations. We employ the following a, b, and p values in our simulation experiments:

1. mixing proportion $p = 0.15, 0.25, 0.5, 0.75, 0.85$;

2. scale of the first subpopulation $a^2 = 1, 2$;

3. The size of the location parameter b is chosen according to the degree of "overlap" between two subpopulations. Since the order of the mixture we consider here is only 2, we use a special case of the general definition of degree of overlap in (2.19). Suppose two subpopulations have means $\mu_1 < \mu_2$ and x_c solves $wf(x_c; \mu_1, \sigma_1) = (1 - w)f(x_c; \mu_2, \sigma_2)$. For each observation x, it makes sense to classify this unit to be a member of the first subpopulation (with mean μ_1) when $x < x_c$; a member of the second subpopulation otherwise. The overlap is then defined as the misclassification probability of X which is given by

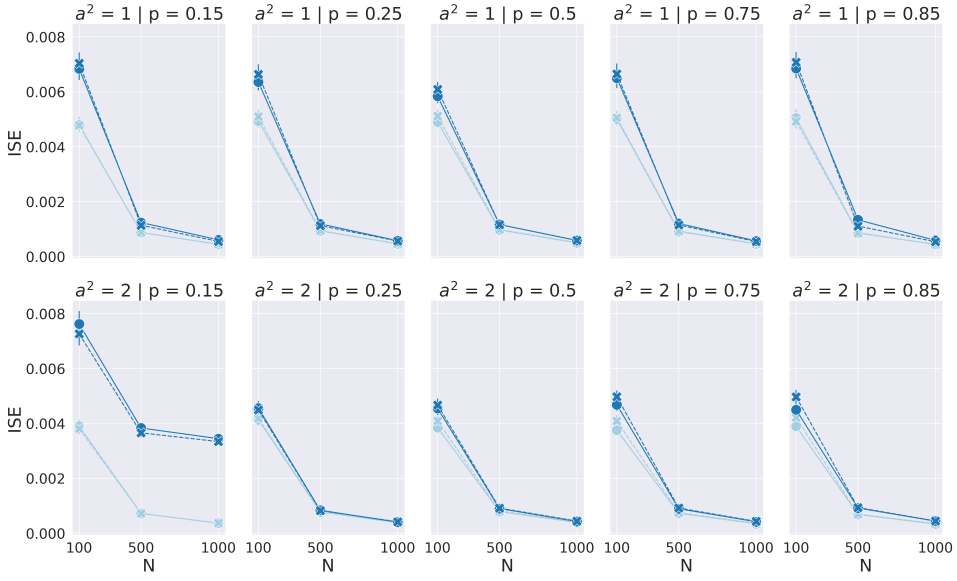
$$pF(1 - x; \mu_2, \sigma_2) + (1 - p)F(x; \mu_1, \sigma_1) \tag{3.6}$$

   In this simulation, we choose the degree of overlap to be either 0.03 or 0.1 and obtain the required b value accordingly.

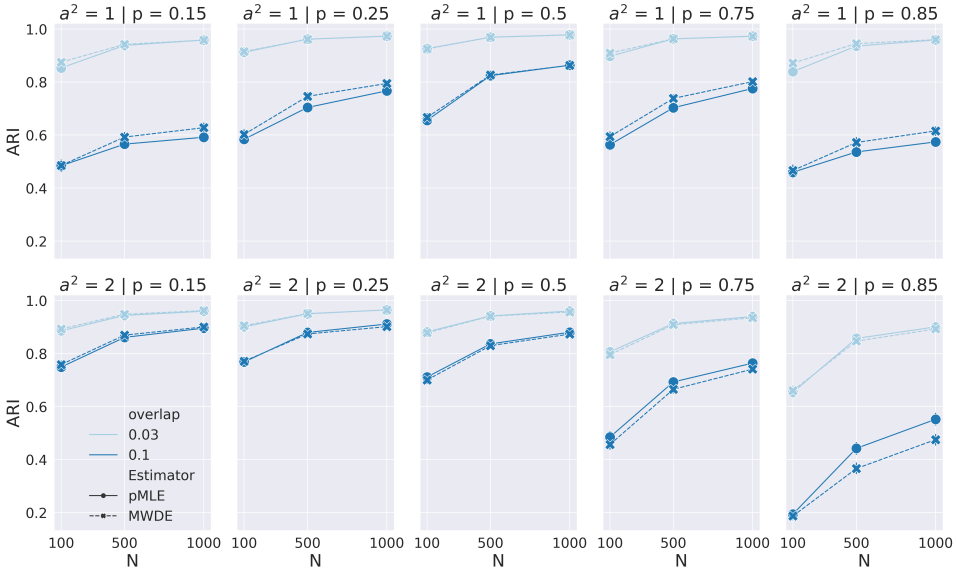
The combination of these choices leads to 24 mixtures with various shapes. The sample size N in our experiments is chosen to be 100, 500, and 1000 respectively.

We obtain the Integrated Squared Error (ISE) and Adjusted Rand Index (ARI) based on $R = 1000$ repetitions on data generated from normal and logistic mixture distributions as specified by (3.5). Figure 3.2 and Figure 3.3 respectively contains plots of ISE and ARI of the MWDE and the PMLE estimators. In these plots, the markers represent the mean value against sample size N under these two models.

We can see that when the sample size increases, the ISE of both estimators decrease and the ARI of both estimators increase, supporting the theory that both MWDE and PMLE are consistent. Under the normal mixture, these two estimators have nearly equal ISE. The MWDE slightly outperforms the PMLE in terms of the ARI, when the degree of overlap is large ($o_{12} = 0.1$) and the two subpopulations have both equal scale and highly unbalanced weights. Under logistic mixture, as shown in plots (a) and (b) of Figure 3.3, the PMLE always outperforms the MWDE in terms of the ISE. In terms of the ARI, the MWDE is better when the scale parameters are equal and weights are highly unbalanced. When the scale parameters are different, the PMLE is better than MWDE when $p > 0.5$ and worse than MWDE

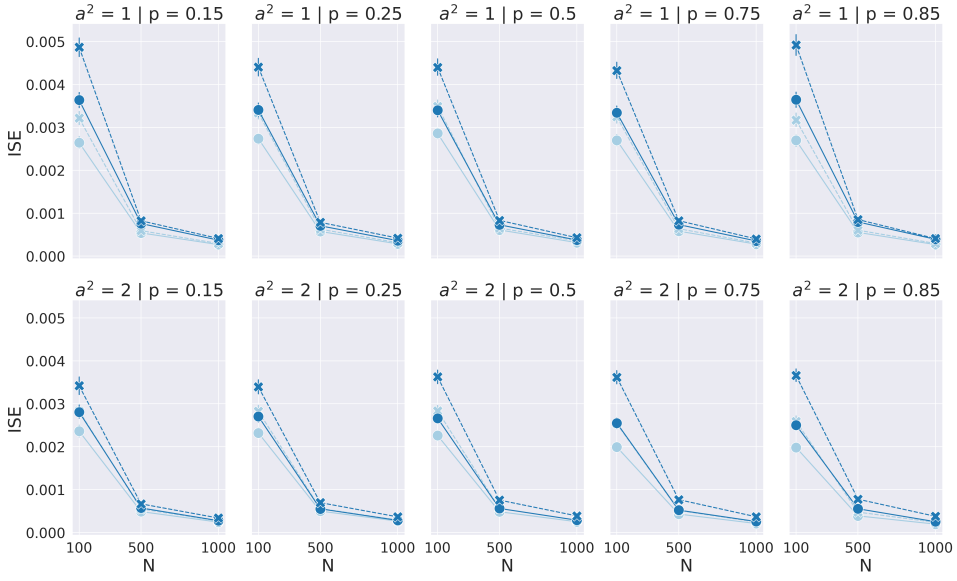


(a) ISE

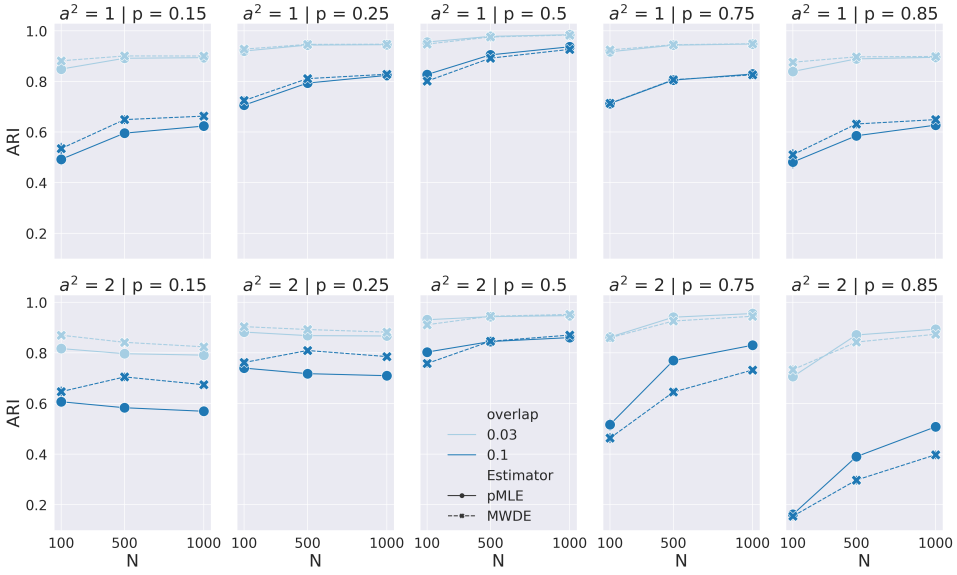


(b) ARI

Figure 3.2: Performances of PMLE and MWDE under 2-component normal mixture in (3.4) when $f_0(x)$ is the standard normal distribution.



(a) ISE



(b) ARI

Figure 3.3: Performances of PMLE and MWDE under 2-component logistic mixture in (3.4) when $f(x;\boldsymbol{\theta})$ is the logistic distribution.

when $p < 0.5$.

We next investigate the performance of the MWDE and PMLE for learning 3-component normal mixtures. We come up with 8 such distributions with different configurations. The three subpopulations have the same or different weights and same or different scale parameter values. They lead to different degrees of overlap as defined by

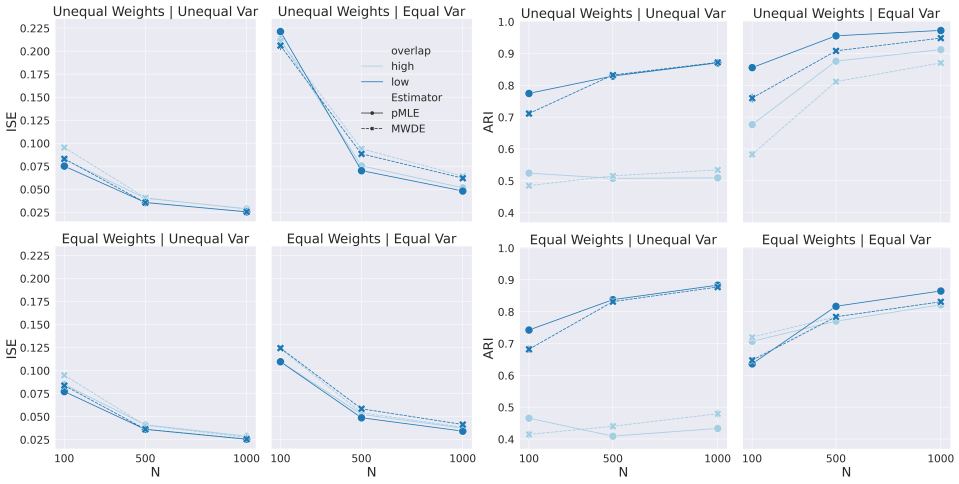
$$\texttt{MeanOmega} = \text{mean}_{1 \le i < j \le 3}\{o_{ij}\}.$$

where o_{ij} is the degree of overlap between subpopulations i and j in (2.19). See Table 3.1 for detailed parameter values.

Table 3.1: Parameter values of 3-component normal mixtures with different degree of overlap. I and II have the same subpopulations means but different subpopulation variances and mixing weights. III and IV have the same subpopulation parameters but different mixing weights. V and VI have the same variances but different subpopulations means and mixing weights. VII and VIII have the same mixing weights and subpopulation variances but different subpopulation means.

	MeanOmega	w_1	w_2	w_3	μ_1	μ_2	μ_3	σ_1	σ_2	σ_3
I	0.288 (low)	0.4	0.5	0.1	-2	0	1	0.3	2	0.4
II	0.367 (high)	0.4	0.5	0.1	-2	0	1	0.3	1	0.4
III	0.097 (low)	0.3	0.5	0.2	-3	0	3	1	1	1
IV	0.249 (high)	0.3	0.5	0.2	-2	0	2	1	1	1
V	0.148 (low)	1/3	1/3	1/3	-1	0	1	1.5	0.1	0.5
VI	0.267 (high)	1/3	1/3	1/3	-0.5	0	0.5	1.5	0.1	0.5
VII	0.091 (low)	1/3	1/3	1/3	-3	0	3	1	1	1
VIII	0.226 (high)	1/3	1/3	1/3	-2	0	2	1	1	1

Figure 3.4 contains plots of the ISE and ARI values of two estimators. The interpretation of the figures is the same as that of Figure 3.2. It is seen that the PMLE consistently outperforms MWDE in terms of ISE but their difference is small. The performances of the MWDE and PMLE are mixed in terms of ARI and the differences are small. The PMLE is clearly better under distributions I and II.



(a) ISE **(b)** ARI

Figure 3.4: Performances of PMLE and MWDE under 3-component normal mixtures whose parameter values are given in Table 3.1.

Robustness

Robustness is another important property of estimators. The sample mean is the most efficient unbiased estimator of the population mean in terms of variance under normality or some other well-known parametric models. However, the value of the sample mean changes dramatically even if the dataset contains merely a single extreme value. Sample median offers a respectable alternative and still has high efficiency across a broader range of parametric models.

In the context of learning finite location-scale mixture models, both PMLE and MWDE rely on a parametric distribution family assumption through $f_0(x)$. How important is to have $f_0(x)$ correctly specified? We shed some light on this problem via empirical experiments in this section.

Let $\phi(x;\mu,\sigma) = (2\pi\sigma^2)^{-1/2}\exp(-(x-\mu)^2/2\sigma^2)$ be the density function of a normal distribution with mean μ and variance σ^2. Note in this chapter, the second parameter in the normal density function $\phi(x;\mu,\sigma)$ is the scale parameter σ not the variance σ^2. Unless otherwise specified, we use the variance parameter for the Gaussian distribution in the rest of this thesis. We learn finite normal mixtures

assuming $K = 2$ but generate data from the following distributions:

1. Mixture with outliers: $(1-\alpha)\{p\phi(x;0,a)+(1-p)\phi(x;b,1)\}+\alpha\phi(x;8,1)$ with $\alpha = 0.01$.

2. Mixture with contamination: $(1-\alpha)\{p\phi(x;0,a)+(1-p)\phi(x;b,1)\}+\alpha\phi(x;b/2,7)$ with $\alpha = 0.01$.

3. Mis-specified mixture I: $pf_0(x;0,a)+(1-p)f_0(x;b,1)$ with $f_0(x)$ being Student-t with 4 degrees of freedom.

4. Mis-specified mixture II: $pf_1(x;0,a)+(1-p)f_2(x;b,1)$ with $f_1(x)$and $f_2(x)$ being Student-t with 2 and 4 degrees of freedom.

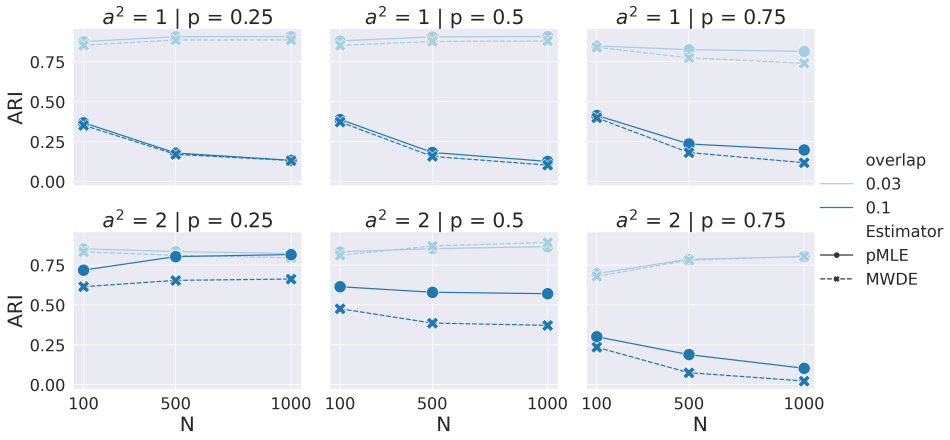
In every case, we use the combinations of the a, b, and p value-combinations that is obtained in studying the efficiency of 2-component normal mixture in (3.4). We regard $\{p\phi(x;0,a)+(1-p)\phi(x;b,1)\}$ as the true distribution in all cases to compute the ARI accordingly.

We obtain the ARI values based on $R = 1000$ repetitions with sample sizes $N = 100$, 500, and 1000, see Figure 3.5 and Figure 3.6. We see that when the degree of overlap is low, MWDE and PMLE have similar performances. When the subpopulation variance is larger ($a^2 = 2$), the performance of PMLE is generally better. In general, we conclude that PMLE is preferred.

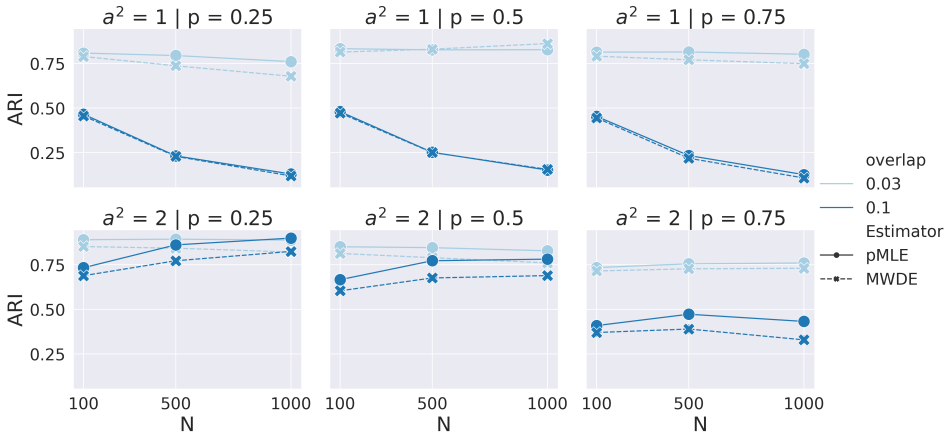
Statistical inference usually becomes more accurate when the sample size increases. This is not the case in this simulation experiment. We can see that ARI often decreases (becomes less accurate) when the sample size increases. This is not caused by simulation error. When the model is mis-specified, the learned model does not converge to the "true model" as $N \to \infty$. Hence, the quality of inference does not necessarily improve. The conclusion of this simulation study is that the MWDE is not more robust than the PMLE.

3.3 Application in Image Segmentation

Image segmentation aims to partition an image into regions, each with a reasonably homogeneous visual appearance or corresponds to objects or parts of ob-

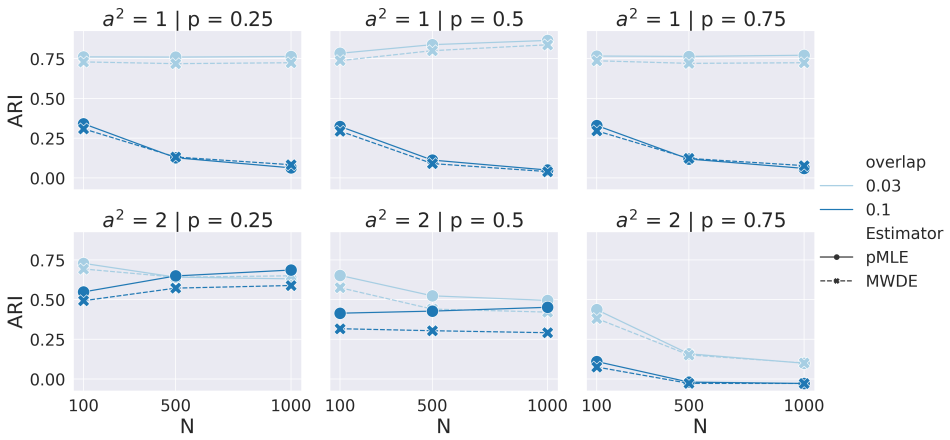


(a) Mixture with outliers

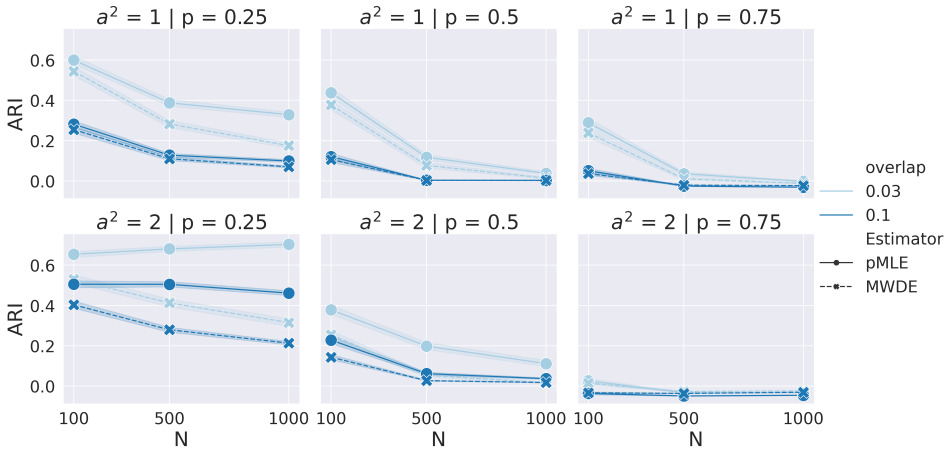


(b) Mixture contaminated

Figure 3.5: Adjusted rand index based on PMLE and MWDE when data contains outliers or is contaminated



(a) Mixture mis-specified I



(b) Mixture mis-specified II

Figure 3.6: Adjusted rand index based on PMLE and MWDE when subpopulation distributions are mis-specified.

jects (Bishop, 2006, Chapter 9). In this section, we perform image segmentation with finite normal mixtures.

Each pixel in an image is represented by three numbers within the range of $[0, 1]$ that corresponds to the intensities of the Red, Green, and Blue (RGB) channels. Since the intensities values are always between 0 and 1, we transform the intensity values to ensure the normal mixture model fits better. Let $y = \Phi^{-1}((x + 1/N)/(1 + 2/N))$ with x being the intensity and N the total number of pixels in the image. Since our numerical algorithm is developed for univariate data, we learn a normal mixture on y values from each channel. Namely, we learn three normal mixtures on red, green, and blue channels respectively.

We use the maximum posterior probability rule to assign each pixel to clusters. We then form an image segment by pixels assigned to the same cluster. We visualize the segregated images channel-by-channel by re-drawing the image with the original intensity value replaced by the average intensity of the pixels assigned to the specific cluster. The segmented images depend heavily on the fitted mixture distributions. We compare the segmented images obtained by the normal mixtures learned via the PMLE and the MWDE. We retrieve an image from Pexel [1] as shown in Figure 3.7 (a). Clark (2015) resized the original high-resolution image to 433×650 grids using Lanczos filter. We learn a normal mixture of order $K = 2$ for each channel based on resized datasets and evaluated its utility of segmenting the foreground and the background.

Table 3.2: Estimated mixing distributions of 2-component mixtures fitted on red, green, and blue channel of the flower image respectively by PMLE and MWDE.

Channel	Estimator	w_1	w_2	μ_1	μ_2	σ_1	σ_2
Red	PMLE	0.896	0.104	-1.668	1.139	1.321	0.277
	MWDE	0.915	0.085	-1.617	1.220	1.316	0.213
Green	PMLE	0.804	0.196	-0.935	0.637	0.373	0.595
	MWDE	0.819	0.181	-0.926	0.724	0.378	0.510
Blue	PMLE	0.735	0.265	-0.753	0.268	0.414	1.034
	MWDE	0.862	0.138	-0.722	1.019	0.473	0.592

[1] https://www.pinterest.se/pin/761952830692007143/

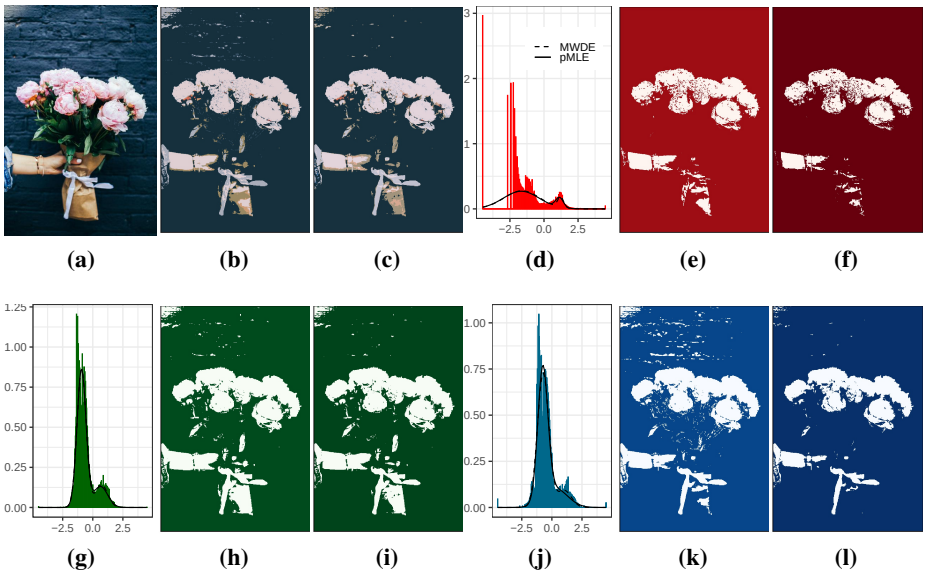


Figure 3.7: Flower image and its segmentation outcomes. Original image in (a). Results for the red, green, and blue channels are shown in panels (d)-(f), (g)-(i), and (j)-(l) respectively. In each group, the left panel shows the histogram and fitted 2-component normal densities based on PMLE and MWDE, the middle panel and the right panel are the image segmentation results based on PMLE and MWDE respectively.

We present the specifications of the learned mixing distributions by PMLE and MWDE in Table 3.2. Plots (d), (g), and (j) in Figure 3.7 are histograms of the transformed intensity values of RGB channels, together with the mixture densities learned via PMLE and MWDE. The corresponding segmented images are shown as plots (e), (h), and (k) for PMLE; (f), (i), and (l) for MWDE. The estimated parameter values and the fitted density on the red and green channels based on these two approaches are very similar. For the blue channel, the fitted densities and the segmentation results are very similar although the estimated parameter values of the second component are different. Both approaches can produce images with meaningful structures segmenting foreground from background.

There are two clusters in each of 3 channels leading to 8 refined clusters. We may paint each pixel with the average RGB intensity triplet according to these

8 refined clusters. The re-created image via PMLE and MWDE respectively are shown in (b) and (c). We note these two images are very similar, showing that both learning strategies are effective.

3.4 Conclusion

The MWDE provides another approach for learning finite location-scale mixtures. We have shown the MWDE is well-defined and consistent. Our moderate scaled simulation study shows it suffers some efficiency loss against a penalized version of MLE in general without a noticeable gain in robustness. The MWDE is computationally more expensive than the PMLE. Therefore, we reaffirm the general superiority of the likelihood-based learning strategies even for non-regular models.

Chapter 4

Distributed Learning of Finite Gaussian Mixtures

In the era of big data, there are various challenges for statistical inference when dealing with large-scale datasets. The sizes of the datasets for various applications may often be so large that they cannot be stored on a single machine. For example, Google distributes its huge database around the world (Corbett et al., 2013). Distributed data storage is also natural when the datasets are collected and managed by independent agencies. Examples include patient information collected from different hospitals and data collected by different government agencies (Agrawal et al., 2003). Privacy considerations may also make it difficult or even impossible to pool the separate collections of data into a single dataset stored in a single facility. Even if the dataset is stored on a single machine, it may not be possible to load all of it into the computer memory. Data analysis methods should therefore be designed so that they can work with subsets of the dataset, in parallel or sequentially. The information extracted from the subsets can then be combined to draw conclusions about the whole population.

In this chapter, we consider the learning of finite Gaussian Mixture Model (GMM) when the data are stored in a distributed fashion. Suppose we have an IID random sample $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ from a distribution $f(x; \theta)$. The dataset $\mathcal{X}$ is said to be a distributed dataset if it is partitioned into M subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M$ completely at random and is stored on M local machines. Let N_m denotes the

sample size on the mth local machine. Clearly, $\sum_{m=1}^{M} N_m = N$.

As indicated in the introduction, this chapter focuses on developing split-and-conquer procedures to learn the finite GMMs. In the split step, we perform standard statistical inference on local machines. We denote by $\widehat{\theta}_m$ the local estimate of θ based on $\mathcal{X}_m$. For example, under finite GMM, the parameter θ becomes the mixing distribution G and the local estimate is the penalized Maximum Likelihood Estimate (PMLE) introduced in Section 2.2.2:

$$\widehat{G}_m = \arg\max \left\{ \sum_{i \in \mathcal{X}_m} \log \phi(x_i; G) - a_m \sum_{k=1}^{K} \left\{ \mathrm{tr}(\Sigma_k^{-1} S_m) + \log \det(\Sigma_k) \right\} \right\}.$$

In the aggregation step, we transmit these local estimates to a central machine to be aggregated.

Various aggregation approaches have been studied in the literature under different settings. The most widely used aggregation approach combines the local estimates by their linear average (Chang et al., 2017; Zhang et al., 2015). Let $\widehat{\theta}_m$ be a local estimate on the mth local machine. The aggregated estimate is then the weighted average

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \widehat{\theta}_m$$

with $\lambda_m = N_m/N$ being the sample proportion.

Liu and Ihler (2014) proposes a different aggregation procedure to learn models from an exponential family. Let $\widehat{\theta}_m$ be the local MLE on the mth local machine, Liu and Ihler (2014) proposes to find the aggregated estimator by

$$\bar{\theta}^{\mathrm{KL}} = \arg\min_{\theta \in \Theta} \sum_{m=1}^{M} \lambda_m D_{\mathrm{KL}}(F(\cdot; \widehat{\theta}_m) \| F(\cdot; \theta)). \tag{4.1}$$

We refer to this aggregated estimator as the KL Averaging (KLA). It is shown in Liu and Ihler (2014) that when $F(x; \theta)$ belongs to the exponential family, then $\bar{\theta}^{\mathrm{KL}}$ is as efficient as the global MLE based on the full dataset. The KLA estimator is less efficient for a distribution that does not belong to the exponential family. The information loss can be characterized by how close $F(x; \theta)$ is to full exponential

family.

In a distributed system with many machines, some worker machines send arbitrarily erroneous information due to hardware or software breakdowns, data crashes, or communication failures (Lamport et al., 1982). This issue is called the Byzantine failure. Distributed learning under the Byzantine failure setting has attracted a lot of attention in recent years, see Blanchard et al. (2017), Yin et al. (2018), Alistarh et al. (2018), Xie et al. (2018), and Tu et al. (2021) for the median estimator and its variants. The robust alternative of simple average, the coordinate-wise median of the local estimates in a vector space, is also widely used when a Byzantine failure occurs.

There are various issues when applying existing aggregation approaches under finite mixtures. First, the simple average approach is appropriate for parameters in a vector space, but it is nonsensical if the average of the parameters is not well defined. Under finite mixtures, the simple average approach leads to a mixture with an inflated number of subpopulations whose mixing distribution is no longer in the same parameter space. Second, the GMM does not belong to any exponential families, the KLA approach therefore does not perform. Moreover, as we show in Section 4.4, the computation of the KLA estimator in (4.1) is difficult under finite mixtures. Under finite mixture models, it is vital to find a statistically efficient aggregation procedure with a low computational cost.

We investigate two aggregation approaches namely the "barycentre" and the "reduction". The barycentre in Section 2.4 generalizes the simple average to probability distribution spaces. Our first approach is to aggregate the local estimates by their barycentre. We find that the aggregation by barycentre may be distorted and therefore focus on aggregating the local estimates by the reduction approach. The rest of this chapter is structured as follows. In Section 4.2, we design an algorithm for computing the reduction estimator. In Section 4.3, we show that the proposed reduction estimator has the best possible statistical convergence rate under certain conditions. In Section 4.4, we review some existing approaches for large-scale learning and particularly the distributed learning of GMM. Numerical experiments on simulated and real data are presented in Section 4.5. We find that the proposed reduction approach performs better than existing approaches. We apply our method to atmospheric datasets in Section 4.5.3. We also study the robustness of the reduc-

tion approach under some scenarios. In Section 4.6, we study the robustness of the reduction approach when the partition of the subsets is not completely at random. In Section 4.7, we study the robustness of the reduction approach when the order of the mixture is over-specified on local machines. In Section 4.8, we provide a discussion and some concluding remarks.

4.1 Aggregation Approaches under Mixture

In this section, we discuss two aggregation approaches: the barycentre approach and the reduction approach.

4.1.1 Aggregation by Barycentre

Under finite mixtures, let $\widehat{G}_1, \ldots, \widehat{G}_m$ be local estimates. The weighted average of the local estimates is

$$\overline{G} = \sum_{m=1}^{M} \lambda_m \widehat{G}_m. \tag{4.2}$$

Its corresponding mixture has density function $f(x; \overline{G}) = \sum_{m=1}^{M} \lambda_m f(x; \widehat{G}_m)$. While $f(x; \overline{G})$ is a good estimate, it can be unsatisfactory for revealing the latent structure of the mixture model. For instance, this estimator is a mixture with MK subpopulations rather than the assumed K, which is useless for clustering the dataset into K clusters.

The analogy of linear average in distribution space is the barycentre introduced in Section 2.4. We take average of the local estimators $\widehat{G}_1, \ldots, \widehat{G}_m$ through their barycentre:

$$\overline{G}^C = \operatorname*{arg\,inf}_{G \in \mathbb{G}_K} \sum_{m=1}^{M} \lambda_m \rho\left(\widehat{G}_m, G\right) \tag{4.3}$$

for some choice of the divergence $\rho(\cdot, \cdot)$ between to mixing distributions. Recall that $\mathbb{G}_K$ is the space of mixing distributions with K support points.

Unfortunately, the barycentre approach may not give sensible aggregated estimator for some choice of the divergence $\rho(\cdot, \cdot)$.

Example 4.1 (Barycentre of Two Univariate Gaussian Mixtures with Identical Sub-

populations). *Suppose we wish to aggregate two local estimates given by*

$$\begin{aligned}\Phi(x;G_1) &= 0.4\Phi(x;-1,1)+0.6\Phi(x;1,1) := 0.4\Phi_{-1}+0.6\Phi_1,\\ \Phi(x;G_2) &= 0.6\Phi(x;-1,1)+0.4\Phi(x;1,1) := 0.6\Phi_{-1}+0.4\Phi_1\end{aligned}$$

with $\lambda_1 = \lambda_2 = 0.5$. *These two local estimates have same subpopulation parameters but different mixing weights. We anticipate that whatever distance or divergence we choose, the barycentre is given by*

$$\Phi(x;\overline{G}) = 0.5\Phi(x;-1,1)+0.5\Phi(x;1,1) = 0.5\Phi_{-1}+0.5\Phi_1.$$

Consider the 2*-Wasserstein distance between two univariate Gaussian distributions with Euclidean ground distance (see Example 2.2)*

$$W_2\left(\Phi(\cdot;\mu_1,\sigma_1^2),\Phi(\cdot;\mu_2,\sigma_2^2)\right) = \{(\mu_1-\mu_2)^2+(\sigma_1-\sigma_2)^2\}^{1/2}.$$

When the divergence $\rho(\cdot,\cdot)$ *in the barycentre is chosen to be the composite transportation divergence with the cost function being the squared* 2*-Wasserstein distance* W_2^2*, that is* $\rho(G_1,G_2)_{=}\mathcal{T}_{W_2}(\Phi(\cdot;G_1),\Phi(\cdot;G_2))$. *Surprisingly, we find that the barycentre with this divergence is given by*

$$\Phi(x;\overline{G}^C) = 0.4\Phi(x;-1,1)+0.6\Phi(x;2/3,1) := 0.4\Phi_{-1}+0.6\Phi_{2/3}.$$

We defer the technical details to Appendix B.1. This abnormal result is rooted in the divergence ρ employed in defining the barycentre. Other choices of ρ may lead to a solution that is consistent with our intuition. However, for the reduction estimator we discuss in the next section, whatever divergence ρ is employed, we always have the anticipated result.

4.1.2 Aggregation by Reduction

Recall that the mixing distribution $\overline{G} = \sum_{m=1}^{M}\lambda_m\widehat{G}_m$ is likely close to the true mixing distribution G^*, except for the incorrect number of support points. This problem can be solved by approximating $\overline{G}$ by some $G \in \mathbb{G}_K$, suggesting another aggregation approach. Let $\rho(\cdot,\cdot)$ be a divergence in the space of mixing distri-

butions. We can aggregate the local estimates via the reduction estimator, given by

$$\overline{G}^R = \underset{G \in \mathbb{G}_K}{\arg\inf}\, \rho(\overline{G}, G). \tag{4.4}$$

In the machine learning community, approximating a GMM by one with a lower order is called Gaussian Mixture Reduction (GMR) (Schieferdecker and Huber, 2009; Williams and Maybeck, 2006; Yu et al., 2018). Williams and Maybeck (2006) uses an optimization-based approach that minimizes

$$\rho(\overline{G}, G) = D_{\text{ISE}}(\Phi(\cdot; \overline{G}), \Phi(\cdot; G))$$

where the analytical form of Integrated Squared Error (ISE) between two mixtures is given in (2.8). Although the ISE between two GMMs has an analytical form, the optimization of $D_{\text{ISE}}(\Phi(\cdot; \overline{G}), \Phi(\cdot; G))$ with respect to $G \in \mathbb{G}_K$ is computationally expensive.

One key observation is that it is usually difficult to compute the divergence between two mixtures, but easy to compute the divergence between two Gaussian distributions. This fact motivates us to consider the Composite Transportation Divergence (CTD) between two Gaussian mixtures as the objective function in the reduction approach. The corresponding GMR estimator is

$$\overline{G}^R = \underset{G \in \mathbb{G}_K}{\arg\inf}\, \mathcal{T}_c(\Phi(\cdot; \overline{G}), \Phi(\cdot; G)) \tag{4.5}$$

for some CTD $\mathcal{T}_c(\cdot, \cdot)$ as given in (2.16). In the rest of this chapter, we write $\mathcal{T}_c(\Phi(\cdot; \overline{G}), \Phi(\cdot; G))$ and $\mathcal{T}_c(\overline{G}, G)$ interchangeably for the ease of presentation.

At a high level, the statistical efficiency of the reduction estimator is guaranteed since $\overline{G}$ is a good estimate for the truth G^*. For the same ρ, it is found that the computation of the barycentre estimator is more expensive than the reduction estimator. We hence propose to aggregate the local estimates by reduction approach. Before we describe the algorithm for the numerical solution to the reduction estimator, we show that these two aggregation approaches – barycentre and reduction – are connected when specific divergences are used.

4.1.3 Connection between Barycentre and Reduction Estimators

Let the divergence $\rho(\cdot, \cdot)$ in the barycentre definition (4.3) or in the reduction estimator (4.4) be the Kullback-Leibler (KL) divergence between two mixtures

$$\rho(G_1, G_2) = D_{\mathrm{KL}}(\Phi(\cdot; G_1) \| \Phi(\cdot; G_2)).$$

In this case, recall $\overline{G} = \sum_{m=1}^{M} \lambda_m \widehat{G}_m$, we have

$$\begin{aligned} D_{\mathrm{KL}}(\Phi(\cdot; \overline{G}) \| \Phi(\cdot; G)) &= \int \phi(x; \overline{G}) \log \left\{ \phi(x; \overline{G}) / \phi(x; G) \right\} \, dx \\ &= C_1 - \int \phi(x; \overline{G}) \log \phi(x; G) \, dx \\ &= C_1 - \sum_{m=1}^{M} \lambda_m \int \phi(x; \widehat{G}_m) \log \phi(x; G) \, dx \\ &= C_2 + \sum_{m=1}^{M} \lambda_m D_{\mathrm{KL}} \left(\Phi(\cdot; \widehat{G}_m) \| \Phi(\cdot; G) \right) \end{aligned}$$

where C_1 and C_2 are constants that do not dependent on G. This relationship implies that for KL divergence, we have

$$\overline{G}^R = \operatorname*{arg\,inf}_{G \in \mathbb{G}_K} \rho(\overline{G}, G) = \operatorname*{arg\,inf}_{G \in \mathbb{G}_K} \left\{ \sum_{m=1}^{M} \lambda_m \rho(\widehat{G}_m, G) \right\} = \overline{G}^C. \tag{4.6}$$

Thus, the two aggregation methods give identical aggregated estimators. It can be seen that with this divergence, the barycentre estimator becomes the KLA estimator proposed in Liu and Ihler (2014).

4.2 Numerical Algorithm for Reduction Estimator

Let $\overline{G}$ be defined as in (4.2). Let the subpopulations in $\overline{G}$ be $\overline{\Phi}_i = \Phi(x; \mu_i, \Sigma_i)$ and the mixing weights be w_i for $i \in [MK]$. Let G be any mixing distribution of order K with the K subpopulations $\Phi_k = \Phi(x; \mu_k, \Sigma_k)$ and the mixing weights v_k for $k \in [K]$. In vector format, the weights are $\boldsymbol{w}$ and $\boldsymbol{v}$.

Let the cost function $c(\cdot, \cdot)$ be a divergence in the space of d-dimensional Gaussian distributions for which the computational cost is low. Then the transportation

divergence (Nguyen, 2013) between mixing distributions $\overline{G}$ and $G \in \mathbb{G}_K$ with cost function c becomes

$$\mathcal{T}_c(\overline{G}, G) = \inf \left\{ \sum_{i,k} \pi_{ik} c(\overline{\Phi}_i, \Phi_k) : \boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \boldsymbol{v}) \right\}.$$

The corresponding GMR estimator is

$$\overline{G}^R = \arg\inf \left\{ \mathcal{T}_c(\overline{G}, G) : G \in \mathbb{G}_K \right\}.$$

For this estimator, it may appear that calculating our estimator involves two optimizations: (i) computing $\mathcal{T}_c(\overline{G}, G)$ for each pair of $\overline{G}$ and G, and (ii) searching for $\arg\inf_G \mathcal{T}_c(\overline{G}, G)$. We are able to design a more efficient optimization algorithm based on the following observation. The optimization problem in $\mathcal{T}_c(\overline{G}, G)$ involves searching for transportation plans $\boldsymbol{\pi}$ under two marginal constraints specified by $\boldsymbol{w}$ and $\boldsymbol{v}$. While constraint $\boldsymbol{w}$ is strict, $\boldsymbol{v}$ is a moving constraint. Instead of searching for $\boldsymbol{\pi}$ satisfies constraint $\boldsymbol{v}$, we move $\boldsymbol{v}$ to meet $\boldsymbol{\pi}$. This makes the marginal distribution constraint $\boldsymbol{v}$ on $\boldsymbol{\pi}$ redundant.

Let us define two functions of G, with $\overline{G}$ hidden in the background:

$$\mathcal{J}_c(G) = \inf_{\boldsymbol{\pi}} \left\{ \sum_{i,k} \pi_{ik} c(\overline{\Phi}_i, \Phi_k) : \boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \cdot) \right\}, \tag{4.7}$$

$$\boldsymbol{\pi}(G) = \arg\inf_{\boldsymbol{\pi}} \left\{ \sum_{i,k} \pi_{ik} c(\overline{\Phi}_i, \Phi_k) : \boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \cdot) \right\}. \tag{4.8}$$

Note that both functions depend on G through its subpopulations Φ_k but are free of its mixing weights $\boldsymbol{v}$. The optimizations in (4.7) and (4.8) involve only the linear constraint in terms of $\boldsymbol{w}$. Hence, the optimal transportation plan $\boldsymbol{\pi}(G)$ for a given G has an analytical form:

$$\pi_{ik}(G) = \begin{cases} w_i & \text{if } k = \arg\min_{k'} c(\overline{\Phi}_i, \Phi_{k'}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

When $c(\overline{\Phi}_i, \Phi)$ has multiple minimizers, we transport $\overline{\Phi}_i$ evenly to every minimum Φ. For example, if $c(\overline{\Phi}_1, \Phi)$ have two minimizers $\Phi_{k'}$ and $\Phi_{k''}$, we let $\pi_{1k'}(G) = \pi_{1k''}(G) = w_1/2$.

Theorem 4.1. *Let $\overline{G}$, $\mathcal{T}_c(\cdot)$, $\mathcal{J}_c(\cdot)$, $\boldsymbol{\pi}(\cdot)$, and other notations be the same as earlier. We have*

$$\inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\} = \inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}. \tag{4.10}$$

The subpopulations of the GMR estimator are hence given by

$$\overline{G}^R = \arg\inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\} \tag{4.11}$$

and the mixing weights are given by $\boldsymbol{v}$ with

$$v_k = \sum_i \pi_{ik}(\overline{G}^R). \tag{4.12}$$

The existence of a solution to (4.11) is guaranteed under a simple condition on cost function $c(\cdot, \cdot)$, see Theorem 4.2. The proof of Theorem 4.1 is in Appendix B.1. Based on this theorem, the optimization reduces to search for K subpopulations Φ_k for $k \in [K]$ to make up G. The mixing proportions are then determined by (4.12). An iterative algorithm quickly emerges following the well-known Majorization Maximization (MM) idea (Hunter and Lange, 2004).

The algorithm starts with some $G^{(0)}$ with K subpopulations specified. Let $G^{(t)}$ be the mixing distribution at the tth MM iterations. Define a majorization function of $\mathcal{J}_c$ at $G^{(t)}$ to be

$$\mathcal{K}_c(G|G^{(t)}) = \sum_{i,k} \pi_{ik}(G^{(t)}) c(\overline{\Phi}_i, \Phi_k) \tag{4.13}$$

where $\pi_{ik}(G^{(t)})$ is computed according to (4.9). Once $\boldsymbol{\pi}(G^{(t)})$ made of $\pi_{ik}(G^{(t)})$ has been obtained, we update the mixing proportion vector of $G^{(t)}$ easily via

$$v_k^{(t+1)} = \sum_i \pi_{ik}(G^{(t)}).$$

In fact, $\boldsymbol{v}^{(t)}$ is not needed until the algorithm converges.

Algorithm 1 MM algorithm for GMR estimator with KL divergence cost function.

Input: $\overline{\Phi}_1, \overline{\Phi}_2, \ldots, \overline{\Phi}_{NK}$
Initialization: Φ_k, $k \in [K]$
repeat
  for $k \in [K]$ **do**
    for $i \in [MK]$ **do**
      Let

$$\pi_{ik} = \begin{cases} w_i & \text{if } k = \arg\min_{k'} D_{\mathrm{KL}}(\overline{\Phi}_i \| \Phi_{k'}) \\ 0 & \text{otherwise} \end{cases}$$

    end for
    Let

$$\pi_{\cdot k} = \sum_{i=1}^{MK} \pi_{ik}, \mu_k = \sum_{i=1}^{MK} \{\pi_{ik}/\pi_{\cdot k}\}\mu_i$$

$$\Sigma_k = \sum_{i=1}^{MK} \{\pi_{ik}/\pi_{\cdot k}\}\{\Sigma_i + (\mu_i - \mu_k)(\mu_i - \mu_k)^\top\}$$

  end for
until the change in the value of the objective function $\sum_{i,k} \pi_{ik} D_{\mathrm{KL}}(\overline{\Phi}_i, \Phi_k)$ is below some threshold $\epsilon > 0$
Let $v_k = \sum_i \pi_{ik}$ for $k \in [K]$
Output: $\{(v_k, \mu_k, \Sigma_k) : k \in [K]\}$

The subpopulations Φ_k are separated in the majorization function (4.13). This allows us to update the subpopulation parameters, one Φ_k at a time and possibly in parallel, as the solutions to

$$\Phi_k^{(t+1)} = \arg\inf_{\Phi} \sum_i \pi_{ik}(G^{(t)}) c(\overline{\Phi}_i, \Phi). \tag{4.14}$$

The MM algorithm then iterates between the majorization step (4.13) and the minimization step (4.14) until some user-selected convergence criterion is met.

The most expensive step in the MM algorithm is the optimization in (4.14). If we choose the cost function $c(\cdot, \cdot) = \rho^r(\cdot, \cdot)$ with $\rho(\cdot, \cdot)$ being a divergence in the space of probability measures, the solution to (4.14) is a barycentre as given in Def-

inition 2.10. The following lemma shows that the KL based barycentre of Gaussian distributions has an analytical form and is therefore computationally simple.

Due to the ease of computing the barycentre as shown in this lemma, we recommend $c(\overline{\Phi}_i, \Phi_k) = D_{\mathrm{KL}}(\overline{\Phi}_i \| \Phi_k)$ in (4.5). Cost functions define the geometries on the Gaussian distribution space (Peyré and Cuturi, 2019), leading to slightly different outputs. We do not rule out the possibility of better choices. The pseudo-code for the MM algorithm with KL divergence as the cost function is given in Algorithm 1.

To make the notation simple, in the following theorem, we use Φ for both the parameter (μ, Σ) and the corresponding distribution, and similarly for Φ^*.

Theorem 4.2. *Suppose the cost function $c(\cdot, \cdot)$ is continuous in both arguments. For some distance in the parameter space of Φ, assume that for any constant $\Delta > 0$ and Φ^* the following set is compact:*

$$\{\Phi : c(\Phi^*, \Phi) \leq \Delta\}. \tag{4.15}$$

Let $\{G^{(t)}\}$ be the sequence generated by $G^{(t+1)} = \arg\min \mathcal{K}_c(G|G^{(t)})$ with some initial mixing distribution $G^{(0)}$. Then

(i) $\mathcal{J}_c(G^{(t+1)}) \leq \mathcal{J}_c(G^{(t)})$ for any t;

(ii) if G^* is a limiting point of $G^{(t)}$, then $G^{(t)} = G^*$ implies $\mathcal{J}_c(G^{(t+1)}) = \mathcal{J}_c(G^*)$.

These two properties imply that $\mathcal{J}_c(G^{(t)})$ converges monotonically to some constant $\mathcal{J}^*$. All the limiting points $G^{(t)}$ are stationary points of $\mathcal{J}_c(\cdot)$: iterations from G^* do not further reduce the value of the objective function $\mathcal{J}_c(\cdot)$. We have practically cloned the global convergence theorem (Zangwill, 1969). We do not see a way to directly apply it and therefore provide a proof of the theorem in Appendix B.1.

We have all the ingredients for the split-and-conquer learning of a finite GMM. We then consider the statistical properties of the GMR estimator and the experimental evidence for the efficiency of our method.

4.3 Statistical Properties of Reduction Estimator

We show that the proposed GMR estimator $\overline{G}^R$ is consistent and retains the optimal rate of convergence in a statistical sense. We first state some conditions on the data and the estimation methods.

C1 The data $\mathcal{X}$ are IID observations from the finite Gaussian mixture $\Phi(x; G^*)$ with K distinct subpopulations, that is the order of G^* is known to be K. The subpopulations have distinct parameters and positive definite covariance matrices.

C2 The dataset $\mathcal{X}$ is partitioned into M subsets $\mathcal{X}_1, \ldots, \mathcal{X}_M$. Each dataset $\mathcal{X}_m$ contains IID observations from the same finite Gaussian mixture distribution and is of size N_m. The number of local machines M does not increase with $N = \sum_m N_m$.

C3 The local machine sample ratio N_m/N have a nonzero limit as $N \to \infty$.

C4 The cost function $c(\Phi_k, \Phi_0) \to 0$ or $c(\Phi_0, \Phi_k) \to 0$ as $k \to \infty$, if and only if $\Phi_k \to \Phi_0$ in distribution, and $c(\Phi_1, \Phi_2)$ is continuous in both Φ_1 and Φ_2.

Condition C4 is necessary to ensure consistency. It further rules out the case that $\mathcal{T}_c(G, G^*) = \infty$ for any G with different mixing weights from that of G^*.

Our proposed reduction estimator is aggregated from the PMLEs learned at the local machines. Under the condition that $\min N_m \to \infty$ stated above, all the local estimators are consistent by Lemma 2.1. Hence, the consistency of the aggregate estimator $\overline{G}$ is taken as granted when the number of local estimators M does not increase with sample size N.

Theorem 4.3 (Consistency of $\overline{G}^{\mathrm{R}}$)**.** *Let $\overline{G}$ be the linear average estimator defined by* (4.2) *and $\overline{G}^R$ be the aggregated estimator by reduction defined by* (4.4) *with $\rho = \mathcal{T}_c$. Assume conditions C1–C4 are satisfied. Then $\overline{G}^R$ is strongly consistent. Specifically, $\mathcal{T}_c(\overline{G}^R, G^*) \to 0$ almost surely as $N \to \infty$.*

The proof of the theorem is given in Appendix B.1. The following theorem shows that under one additional mild condition on the cost function $c(\cdot, \cdot)$, the reduction estimator $\overline{G}^R$ has the standard $N^{-1/2}$ convergence rate. We denote by

$\|\Phi_1 - \Phi_2\|$ the Euclidean norm in μ, Σ in the sense of (2.15). We use $\overline{\Phi}_k^R$ for the kth subpopulation of $\overline{G}^R$ and $\overline{w}_k$ for its mixing weight for $k \in [K]$.

Theorem 4.4 (Convergence Rate of $\overline{G}^{\mathrm{R}}$)**.** *Let $\overline{G}$ be the aggregate estimator defined by* (4.2) *and $\overline{G}^R$ be the aggregate estimator by reduction defined by* (4.4)*. Assume conditions C1–C4 are satisfied and further assume that*

C5 *For any Φ, there exists a small neighbourhood Ω of Φ and a positive constant A, such that for any $\Phi_1, \Phi_2 \in \Omega$, we have*

$$A^{-1}\|\Phi_1 - \Phi_2\|^2 \leq c(\Phi_1, \Phi_2) \leq A\|\Phi_1 - \Phi_2\|^2.$$

Then with proper labelling of subpopulations, we have

$$\overline{\Phi}_k^R - \Phi_k^* = O_p(N^{-1/2}), \;\; \overline{w}_k^R - \pi_k^* = O_p(N^{-1/2}).$$

Condition C5 requires the cost function $c(\cdot, \cdot)$ behaves locally as a quadratic loss function. This is a most natural property for a cost function. In Appendix B.1, we show this condition holds for the KL divergence. The conclusion should hold with any other reasonable choices. The proof of the theorem is given in Appendix B.1. Our proof remains valid, for instance, if we replace $\|\Phi_1 - \Phi_2\|^2$ by $\|\Phi_1 - \Phi_2\|^r$ for any $r > 0$ in C5.

4.4 Related Work

In this section, we describe several related approaches for learning GMM with large datasets. We compare some of these approaches with our proposed approach in Section 4.5 in the experiments.

KL Averaging

Liu and Ihler (2014) considers the distributed learning of models from an exponential family by the split-and-conquer approach. The parameter θ is a real vector in this case. Liu and Ihler (2014) proposes to perform local inference by finding the local MLEs and aggregate them by their KL barycentre. This estimator is referred to as the KLA. When the model belongs to the exponential family, the aggregated

estimator is as efficient as the global MLE based on the full dataset. For models not in the exponential family, such as GMM, the KLA estimator is less efficient. Moreover, the exact computation of the KL barycentre of local estimators under GMM is difficult. Liu and Ihler (2014) suggests to find an approximate solution instead. Liu and Ihler (2014) first generates random samples $\widehat{\mathcal{X}}_m$ of size 1000 from the local estimates $\widehat{G}_m$ at the central machine. Then a GMM of order K is fitted on the pooled sample $\cup_m \widehat{\mathcal{X}}_m$ which has a moderate size of $1000M$. This approach does not need to transmit the raw data but requires refitting of the mixture on the central machine.

Distributed EM Algorithm

The distributed learning of GMMs can also be tackled by developing a distributed version of the Expectation Maximization (EM) algorithm (DEM) (Nowak, 2003). We briefly describe DEM and provide a conceptual comparison to our approach in this section.

Under distributed learning setting, let $\mathcal{X}_m$ be the dataset stored at the mth local machine $m \in [M]$ and N be the total sample size. Note that the quantities required in defining $Q(G; G^{(t)})$ based on full dataset $\mathcal{X}$ have the following decomposition: for $k \in [K]$,

$$\begin{aligned}
\Omega_k^{(t)} &:= \sum_{i=1}^{N} \omega_{ik}^{(t)} = \sum_{m=1}^{M} \left\{ \sum_{i \in \mathcal{X}_m} \omega_{ik}^{(t)} \right\} := \sum_{m=1}^{M} \Omega_{m,k}^{(t)}, \\
\mathbf{A}_k^{(t)} &:= \sum_{i=1}^{N} \omega_{ik}^{(t)} x_i = \sum_{m=1}^{M} \left\{ \sum_{i \in \mathcal{X}_m} \omega_{ik}^{(t)} x_i \right\} := \sum_{m=1}^{M} \mathbf{A}_{m,k}^{(t)}, \\
\mathbf{B}_k^{(t)} &:= \sum_{i=1}^{N} \omega_{ik}^{(t)} x_i x_i^\top = \sum_{m=1}^{M} \left\{ \sum_{i \in \mathcal{X}_m} \omega_{ik}^{(t)} x_i x_i^\top \right\} := \sum_{m=1}^{M} \mathbf{B}_{m,k}^{(t)}.
\end{aligned}$$

Given $G^{(t)}$, one can compute $\omega_{ik}^{(t)}$ defined by (2.2) for ith observation on local machine m for all $m \in [M]$ and $k \in [K]$. Hence, one can obtain the local summary statistics $\cup_{k=1}^{K} \{\Omega_{m,k}^{(t)}, \mathbf{A}_{m,k}^{(t)}, \mathbf{B}_{m,k}^{(t)}\}$ at the mth local machine and have them transmitted to a central machine. One can then construct $Q(G; G^{(t)})$ on the central machine and carry out the M-step to get $G^{(t+1)}$, which reproduces the EM iteration

based on the full dataset.

Nowak (2003) considers the situation where the local machines form a sensor network and the transmission cost cannot be ignored. This paper suggests the mth machine transmits $\cup_{j=1}^{m}\{\Omega_{j,k}^{(t)}, \mathbf{A}_{j,k}^{(t)}, \mathbf{B}_{j,k}^{(t)}\}$ to the next machine in the queue. Furthermore, it adopts the incremental E and M steps of Neal and Hinton (1998) to speed up the convergence of the algorithm. Nowak (2003) further shows that the DEM has a local linear convergence rate.

The DEM and proposed GMR approaches are designed for distributed learning with different communication schemes. DEM requires a high level of coordination between local machines and repeated access of the local data. The computation at each local machine is equivalent to an EM iteration based on local data. Our proposed method allows local machines to complete the learning on their own. Moreover, our method only requires one round of communication across all machines and is communication efficient. If successful, DEM leads to a solution to the original learning problem retaining full statistical efficiency. We do not include DEM in the experiment because the conclusions are already known. Our proposed method is superior in terms of communication cost. The statistical efficiency of DEM is same as the global estimator which is compared with our proposed method in the experiment.

Learning at Scale via Coresets

Most machine learning problems can be formulated as an optimization problem that minimizes a cost function $\text{cost}(\mathcal{X}, \theta)$ over the parameter space. For our problem, the cost function could be the negative log-likelihood, $\mathcal{X}$ is the full dataset, and the parameter space $\mathbb{G}_K$ is given by (1.2).

When $\mathcal{X}$ is very large, the computational burden can be extremely heavy. Feldman et al. (2011) suggests to replace $\mathcal{X}$ by a much smaller weighted subset $\mathcal{C}$, called coreset hereafter, such that

$$\frac{|\text{cost}(\mathcal{X}, G) - \text{cost}(\mathcal{C}, G)|}{\text{cost}(\mathcal{X}, G)} \leq \epsilon \tag{4.16}$$

for some given small $\epsilon > 0$. Minimize the $\text{cost}(\mathcal{C}, G)$ based on the coreset can be much faster than the original cost based on the full dataset. The construction of

coreset is to assure the minimizer of cost$(\mathcal{C}, G)$ approximates that of cost$(\mathcal{X}, G)$. Lucic et al. (2017) provides theoretical analysis and techniques for constructing coresets under GMM. For GMM of dimension d, order K, it gives a scheme to obtain coresets of size $|\mathcal{C}| = O(d^4 K^6 \epsilon^{-2})$ satisfying (4.16) uniformly over some compact subset of $\mathbb{G}_K$. This is a surprising result as the size of $\mathcal{C}$ does not depend on the size of $\mathcal{X}$.

When the datasets are stored in distributed fashion, one may first reduce the dataset in each machine into the first generation coresets. Then these coresets are paired up to create a second generation coreset from each pair. This procedure is repeated if needed until we get a final coreset. Due to the composition properties of coreset (Lucic et al., 2017, Section 5), the quality of the final coreset can be maintained. We refer to this approach as the Coreset approach hereafter.

The Coreset approach is computationally very efficient. Unlike DEM, it looks for approximate solutions leading to inevitable loss in statistical efficiency. Moreover, the Coreset approach requires the transmission of the raw data unlike other approaches for distributed learning that only requires the communication of summary statistics. We include the Coreset approach in our experiment for efficiency comparison.

We wish to remark that the log-likelihood function in statistics is defined up to an additive constant. The precision specification (4.16) can be affected by how it is normalized. We adopt the normalization convention of (Lucic et al., 2017).

Bayesian Moment Matching (BMM)

Direct Bayesian inference under GMM is challenging due to the well-known label switching problem. See Murphy (2012, Chapter 11) for explanation and possible solutions. Jaini and Poupart (2016) proposes an approximate inference procedure that does not suffer from the issue of label switching. Under GMM with known order K, given a prior π_0 with Dirichlet for weights and Gaussian-Gamma for subpopulation parameters, the posterior distribution $\widetilde{\pi}_1$ with **a single observation** x_1 is a complex mixture of Dirichlet and Gaussian-Gamma combination. Repeating this operation given another observation would lead to an even more complex posterior. Instead, Jaini and Poupart (2016) suggests to approximate $\widetilde{\pi}_1$ posterior

with a simple Dirichlet and Gaussian-Gamma combination $\boldsymbol{\pi}_1$ so that $\widetilde{\pi}_1$ and π_1 have the same lower order moments. Let π_1 be the prior distribution, with the next single observation x_2, we obtain π_2 in the same way. Repeat sequentially until we have exhausted all data to get π_N. Under multivariate GMMs, one replaces Gaussian-Gamma prior by Gaussian-Wishart. The end product π_N is regarded as an approximate posterior and it serves well for Bayes inferences.

Being sequential in nature, BMM is computationally efficient but apparently loses statistical efficiency due to approximation. Based on Table 2 in Jaini and Poupart (2016) and our Table 4.1, the per observation log-likelihood value of BMM when applied to Magic04 is -32.1, which is much lower than that of our proposed GMR -26.6. Jaini and Poupart (2016) shows that BMM has higher statistical efficiency than the online EM approach of Cappé and Moulines (2009, Theorem 2), whose convergence rate is lower than $N^{-1/2}$.

We do not include BMM in the experiment not only because of the comparison above but also because we do not know how they handle the redundant moment equations. Under the multivariate GMM, there are $K + dK + d(d+1)K/2$ parameters to be estimated at each step but there are $dK + 2K - 1 + 3d(d+1)K/2$ moment equations. The redundant equations make it difficult for us to replicate their approach.

ADMM for Distributed Optimization

The distributed learning of GMM is essentially a distributed optimization problem. We may therefore directly use the Alternating Direction Method of Multipliers (ADMM) of Boyd et al. (2011). Consider the optimization problem defined as

$$\min \left\{ \sum_{m=1}^{M} f(\theta_m | \mathcal{X}_m) : \theta_m - \theta = 0,\ m \in [M],\ \theta \in \Theta \right\}$$

for some function $f(\cdot|\mathcal{X}_m)$ and a Euclidean parameter space Θ. The parameters θ_m are called local variables and θ is called a global variable. The ADMM for this

optimization problem is based on the augmented Lagrangian

$$L_\lambda(\theta_m, \eta_m, \theta) = \sum_{m=1}^{M} \left\{ f(\theta_m|\mathcal{X}_m) + \eta_m^\top(\theta_m - \theta) + (\lambda/2)\|\theta_m - \theta\|_2^2 \right\}$$

for some regularization parameter $\lambda > 0$. The ADMM then iterates according to

$$\begin{aligned}
\theta_m^{(t+1)} &= \arg\min_{\theta_m} \left\{ f(\theta_m|\mathcal{X}_m) + (\eta_m^{(t)})^\top(\theta_m - \bar{\theta}^{(t)}) + (\lambda/2)\|\theta_m - \bar{\theta}^{(t)}\|_2^2 \right\}, \\
\bar{\theta}^{(t+1)} &= M^{-1} \sum_{m=1}^{M} \theta_m^{(t+1)}, \\
\eta_m^{(t+1)} &= \eta_m^{(t)} + \lambda\{\theta_m^{(t+1)} - \bar{\theta}^{(t+1)}\}.
\end{aligned}$$

Similar to DEM, the ADMM requires a high level of coordination between local machines at each iteration. If successful, it gives the solution to the original optimization problem, void the statistical efficiency comparison. In the context of GMM, one must look for suitable substitutes for the term $(\eta_m^{(t)})^\top(\theta_m - \bar{\theta}^{(t)})$ in the augmented Lagrangian since θ_m is a discrete distribution in our context. We therefore do not include ADMM in the experiment.

4.5 Experiments

We conduct experiments on both simulated and real data to illustrate the effectiveness of the proposed GMR estimator in (4.5) with $c(\overline{\Phi}_i, \Phi_k) = D_{\mathrm{KL}}(\overline{\Phi}_i \| \Phi_k)$ between any two Gaussian distributions $\bar{\Phi}_i$ and Φ_k. We compare its performance with some existing approaches in terms of their statistical efficiency and computational costs. Our experiments include the following estimators:

1. *Global*. The PMLE based on the full dataset. The Global estimator is statistically most efficient and therefore used as the baseline for comparison.

2. *Median*. An off-the-shelf aggregation approach is the median

$$\overline{G}^{\mathrm{M}} = \arg\min_{G \in \{\widehat{G}_1, \widehat{G}_2, \ldots, \widehat{G}_M\}} \sum_{m=1}^{M} \lambda_m \mathcal{T}_{D_{\mathrm{KL}}}(\widehat{G}_m, G).$$

The sample median is intuitively a robust alternative with minor efficiency loss. We call this estimator as Median, it is different from the median estimator for split-and-conquer aggregation of models with vector parameter space. The median estimator in the latter case is the coordinate-wise median of local estimators in vector space.

3. *KLA*. The KL-Averaging in Liu and Ihler (2014) with $n_m = 1000$ observations generated from local estimate $\widehat{G}_m$. The real datasets have different dimensions and sample sizes, the size n_m for real data experiments is specified if different.

4. *Coreset*. The Coreset approach with card$(\mathcal{C}_m) = 1000$ on each local machine. They are repeatedly merged as in Lucic et al. (2017) to arrive at the final coreset $\mathcal{C}$ of size 1000, the coreset sizes for real data experiment is specified if different.

For the ease of comparison, we use the PMLE defined in (2.3) with penalty size $N_m^{-1/2}$ as local estimates when applicable. The PMLE is also used in the KLA estimator of Liu and Ihler (2014) on the central machine with penalty size $(1000M)^{-1/2}$, and for the Coreset method with penalty size card$(\mathcal{C})^{-1/2}$. We use the EM algorithm to compute PMLE and declare convergence when the per observation penalized log-likelihood function is less than 10^{-6}. With very large sample sizes of the simulated data, the maxima of the penalized likelihood should be attained at a mixing distribution close to the true mixing distribution. We therefore use the true mixing distribution as the initial value and regard the output of the EM algorithm as the global maximum of the penalized likelihood. This strategy does not work for the real-world data in the absence of a true mixing distribution. For real-world data, we use *kmeans++* with default arguments in `scikit-learn` package (Pedregosa et al., 2011) to generate 10 initial values for the EM algorithm. Ideally, we run the EM algorithm with these initial values until convergence and regard the output of the EM algorithm with the highest penalized log-likelihood function as the PMLE. To save time on the real dataset, we use a warm up strategy. We run the EM algorithm with these 10 initial values for 20 iterations and pick the one with the highest penalized log-likelihood value. We use the output of this

one as the initial value to run the EM algorithm further until convergence and this output is treated as the PMLE.

The choice of the initial value in the aggregation step is also important. When the sample size is large, we have good reason to believe that the optimal solution is close to the true value. Also, by the principle of majority rules, the median of the local estimates is likely the closest to the optimal solution. Thus, in simulation studies, we initialize the algorithm with the true mixing distribution and the median estimate. For real-world data, we use the local estimators as multiple initial values and output of the MM algorithm with the lowest objective function value is regarded as the GMR estimator. We declare the convergence of the MM algorithm for the GMR estimator when the change in the objective function is less than 10^{-6}.

All experiments are conducted on the Compute Canada (Baldwin, 2012) Cedar cluster with Intel E5 Broadwell CPUs with 64G memory. The codes are written in `Python` and are publicly available at https://github.com/SarahQiong/SCGMM. The code for Coreset method is provided by the author of Lucic et al. (2017).

Performance Measure

The split-and-conquer approach may also reduce the computational time by performing local inference on multiple machines. Besides the metrics in Section 2.5 to measure the statistical efficiency of the estimators, we also report the computational times of all the methods. The computational time of a split-and-conquer approach is defined to be the sum of the time for the local estimates and that for the aggregated estimator. Since the local estimates can be computed in parallel, we record the longest local machine time as the time for the local estimates.

4.5.1 Simulated Datasets

Distributed learning methods are designed for learning at scale where the observations have high dimensions and large sample size. To reduce the potential influence of human bias, we simulate data from finite Gaussian mixtures with randomly generated parameter values. We use the `R` package `MixSim` (Maitra and Melnykov, 2010; Melnykov et al., 2012). An important quantity of a finite mixture is pairwise overlap o_{ij} as given in Definition 2.11. The maximum overlap of a finite mixture

is defined to be

$$\texttt{MaxOmega} = \max_{i,j \in [K]} o_{ij}.$$

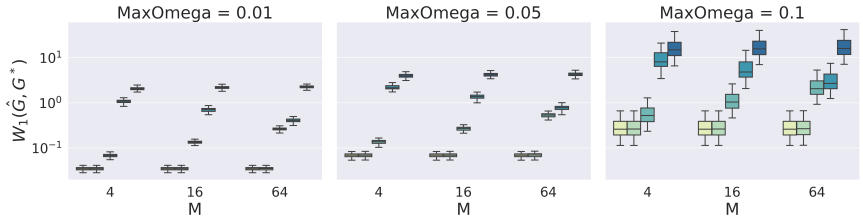
We use `MixSim` to generate 100 finite Gaussian mixtures with $d = 50$ and $K = 5$. The results for other d and K are in the Appendix B.2. We let `MaxOmega` be 1%, 5%, and 10%, $N = 2^l$ for $l = 17, 19, 21$ and $M = 2^l$ for $l = 2, 4, 6$. The simulated data are divided evenly over the local machines. We combine 100 outcomes from each combination of dimension, order, `MaxOmega`, sample size, and number of local machines to form boxplots for each estimation method. Figure 4.1 and Figure 4.2 show the results.

Figure 4.1 reports the result when the total sample size is $N = 2^{21}$. Within each subfigure, the `MaxOmega` increases from the left panel to the right panel. Within each panel, the x-axis gives the number of local machines: 4, 16, or 64. The plots in Figure 4.1 contain boxplots of W1, Adjusted Rand Index (ARI), LL, and the computational time.

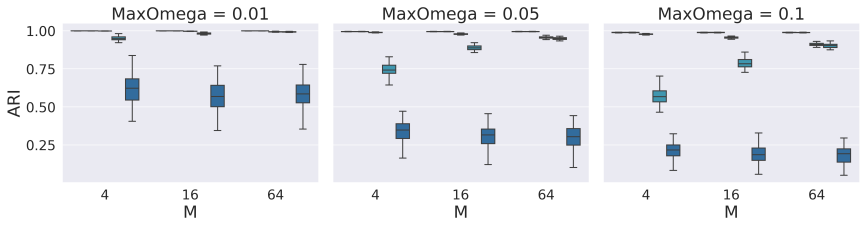
Based on Figure 4.1, all methods have better performance in terms of W1, ARI, and LL when `MaxOmega` is lower. This is consistent with our intuition and the experiment survives the sanity check.

The proposed GMR has comparable performance to the gold-standard global estimator in all three metrics. It is arguably the best aggregation approach. The number of local machines has little influence on its performance. KLA has relatively poor performance though it improves moderately when the number of local machines increases. Recall that KLA generates a fixed number of observations from each local estimator. More local machines lead to a larger total sample size in its aggregation step. This helps its statistical performance but not computational efficiency. The median estimator does not perform with a large number of local machines. The Coreset estimator does not perform in all cases.

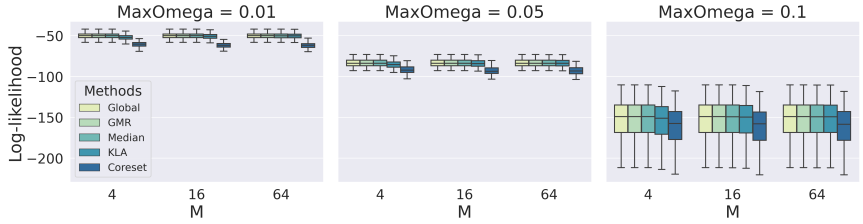
All aggregation approaches take less computational time than the global estimator. The Coreset estimator takes the least amount of computational time. However, the computational time does not take the transmission cost into account. Moreover, the computational time does not make up the poor statistical performance. By far a large chunk of computational time of the split-and-conquer approaches is spent on learning the local estimators. Because GMR and the median



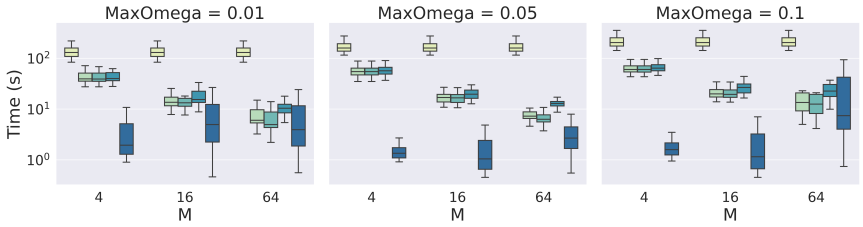
(a) Transportation distance W1



(b) ARI



(c) Log-likelihood



(d) Computation time

Figure 4.1: Performance of five estimators: Global, GMR, median, KLA, and Coreset from left to right in each block of 5 in terms of (a) W1 distance, (b) ARI, (c) log-likelihood per observation, and (d) computational time for learning 50-dimensional order $K = 5$ Gaussian mixtures with sample size $N = 2^{21}$. M is the number of local machines. For W1 distance, the smaller the better. For ARI and log-likelihood, the larger the better.

estimators spend negligible time on aggregation, they use about $1/M$ of the global estimator computational time. When the data generating mixture has a high degree of overlapping, the EM algorithm needs more iterations to converge leading to higher computational time. KLA must re-learn the GMM based on the pooled data generated from the local estimates, therefore generally takes longer time than the GMR approach.

Figure 4.2 presents results when `MaxOmega` is 10% and $M = 64$. It is seen the proposed GMR has very good performance, comparable to the global estimator. The relative performance of the median estimator improves when the sample size increases, but still not good enough to be recommended. The performance of KLA and Coreset approaches do not improve with increased sample size. They are far from competitive against the proposed GMR approach.

Coreset approach does not benefit from larger sample size likely because the coreset size is fixed at 1000. KLA approach does not because the larger sample size improves only the precision of the local estimates. Its aggregation step is heavily influenced by the built-in randomness when we generate samples from the local estimates. The other aspects of the simulation results are as expected.

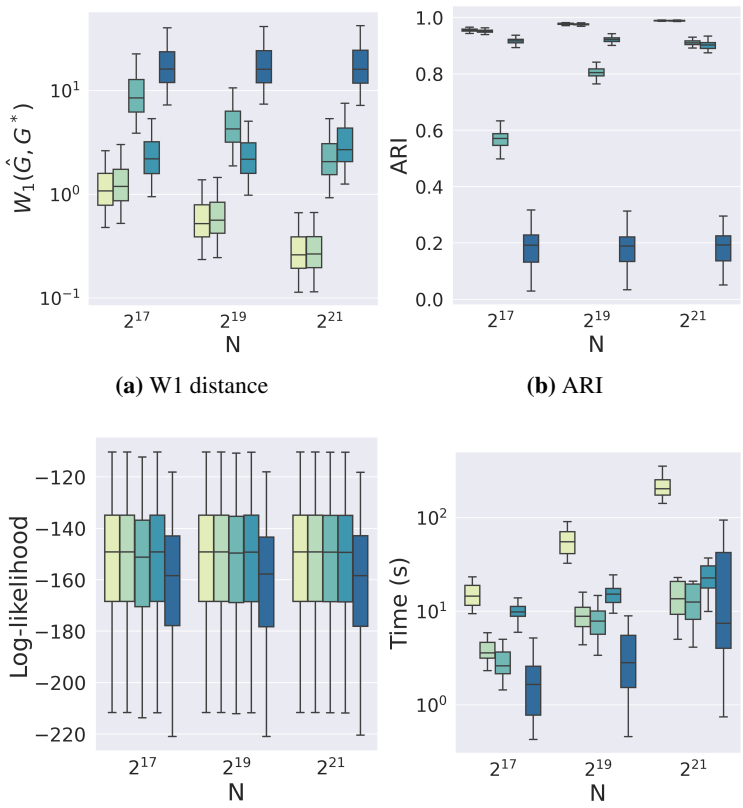
We have more simulation results when the dimensions $d = 10, 50$ combined with orders $K = 5$, 10, and 50. They are presented in Appendix B.2. These results are consistent with what we find so far in terms of the statistical efficiency. Since the Coreset estimator is found not competitive, it is not included in these experiments.

4.5.2 Real Datasets

We now examine the performance of the proposed approach on large-scale public datasets in Section 4.5.2, for clustering the handwritten digits in Section 4.5.2, and for clustering a large-scale spatio-temporal data in Section 4.5.3.

Public Datasets

We experiment on the public datasets that are widely used for learning Gaussian mixtures at scale in this section. The following datasets are used in Lucic et al. (2017) and Jaini and Poupart (2016).



(a) W1 distance **(b)** ARI

(c) Log-likelihood **(d)** Computation time

Figure 4.2: Performance of five estimators: Global, GMR, median, KLA, and Coreset from left to right in each block of 5 in terms of (a) W1 distance, (b) ARI, (c) log-likelihood per observation, and (d) computational time for learning 50-dimensional $K = 5$ Gaussian mixtures with sample size $N = 2^{21}$ and when `MaxOmega` $= 0.1$ and $M = 64$. For W1 distance, the smaller the better. For ARI and log-likelihood, the larger the better.

1. MAGIC04. This is a simulated dataset for classifying gamma particles in the upper atmosphere. It contains $19,020$ observations with 10 real valued features and is publicly available at UCI machine learning repository.

2. MINIBOONE. The dataset is taken from the MiniBooNE experiment that is used to distinguish electron neutrinos from muon neutrinos. It contains $130,065$ observations with 50 real valued features and is publicly available at UCI machine learning repository.

3. KDD. This dataset is used in Lucic et al. (2017). It contains $145,751$ observations with 74 real valued features for predicting the protein types. It is available at https://kdd.org/kdd-cup/view/kdd-cup-2004/Data.

4. MSYP. The dataset is used to predict the release year of a song from audio features. It contains $515,345$ observations with 90 real valued features. The dataset is publicly available at UCI machine learning repository. Following Lucic et al. (2017), we reduce the dataset to its top 25 principal components and fit the mixture model with these 25 features.

The nature of features in these datasets is not important in our demonstration. Details about these features can be found in the corresponding data repository sources. For the first three datasets, we divide the dataset onto $M = 4$ local machines completely at random. Since MSYP is very big and the order of the mixture to be fitted is high, we divide the dataset onto $M = 16$ local machines. The random partition of the dataset is repeated $R = 100$ times. The size of the generated sample and coreset size are set to be 1000 for MAGIC04, and $10,000$ for other datasets. The order of the fitted mixture on each dataset follows the setting in Lucic et al. (2017) and is specified in Table 4.1.

For each method, due to repetition, we have 100 LL values based on the full dataset at the learned mixing distributions. We summarize these values by its median and inter quartile range (IQR). We also obtain the total computational time for each method. These results are given in Table 4.1.

Based on Table 4.1, it is clear that the proposed GMR has the best performance among all split-and-conquer based approaches in terms of the LL value. For the MiniBooNE dataset, the LL values of KLA and Coreset approaches are very small.

Table 4.1: Performance of five learning approaches Global, GMR, Median, KLA, and Coreset on four large-scale public datasets.

Dataset	N	d	K	M	Global	GMR	Median	KLA	Coreset
Median (IQR) LL values (the larger the better)									
MIGIC04	19020	10	10	4	-24.15	-24.30(0.07)	-26.60(0.05)	-26.73(0.07)	-27.16(0.55)
MiniBooNE	130065	50	10	4	-19.46	-22.00(0.53)	-24.60(0.32)	$-6.41(1.95) \times 10^3$	$-8.6(2.56) \times 10^9$
KDD	145751	74	10	4	-221.80	-223.25(0.42)	-232.93(8.02)	-235.00(8.96)	-374.43(193.58)
MSYP	515345	25	50	16	-166.56	-167.05(0.04)	-171.10(0.04)	-170.72(0.01)	-181.64(1.78)
Median (IQR) computational times in seconds									
MIGIC04	19020	10	10	4	19.3	7.0(3.2)	6.7(3.2)	10.2(3.1)	2.2(0.6)
MiniBooNE	130065	50	10	4	346.9	313.1(162.6)	313.2(162.6)	511.3(213.2)	26.6(64.3)
KDD	145751	74	10	4	1033.9	544.4(309.5)	543.0(310.0)	706.0(290.3)	4.3(64.0)
MSYP	515345	25	50	16	67048.8	2611.6(474.0)	1777.5(511.2)	5515.9(1629.7)	67.4(12.6)

This is likely because the total sample size to refit the GMM on the central machine is relatively small in 50-dimensional space. The fitted order 10 mixture may not be able to cover the entire space properly and the log-likelihood contribution of some observations is practically negative infinity. A single near zero likelihood value could lead to a very small LL value. To evaluate the improvement of the GMR approach over other approaches, we consider how many extra subpopulations are needed to achieve the same gain in the LL value. Note that the famous BIC (Schwarz, 1978) would favour a model with an extra parameter if the gain in LL is more than $\log(N)/(2N)$. The $\log(N)/(2N)$ values for these datasets are $(26, 4.5, 4.1, 1.3) \times 10^{-5}$. A subpopulation in GMM with $d = 74$ needs $1 + 74 + 74 \times 75/2 = 2854$ parameters. This translates into a difference of 0.116 in LL. For the KDD data, the gain in GMR compared to KLA would allow another 17 subpopulations.

All the split-and-conquer learning methods are much faster than the global method. For the MSYP dataset, the split-and-conquer methods can be 10 times as fast compared to the global estimator. The Coreset method takes the shortest time. The proposed GMR approach takes comparable computational time with the KLA approach.

NIST Handwritten-Digit Dataset

The finite GMM is often used for model-based clustering (Fraley and Raftery, 2002; Friedman et al., 2001, Chapter 14.3). When the dataset is large and/or distributed

over many local machines, split-and-conquer approaches such as the proposed GMR become useful. In this section, we demonstrate the use of the GMR method on the famous NIST dataset for character recognition (Grother and Hanaoka, 2016). We use the second edition of the dataset, named *by_class.zip* [1]. It consists of approximately 4M images of handwritten digits and characters (0–9, A–Z, and a–z) by different writers. Our experiment focuses on the digits and we still refer to it as the NIST dataset. The images of the digits are in directories 30–39. According to the user guide [2], the images in the *train_30* to *train_39* and *hsf_4* folders are used as the training and test sets respectively. The numbers of training images for each digit are listed in the following table:

Table 4.2: The numbers of training images for each digit in NIST dataset.

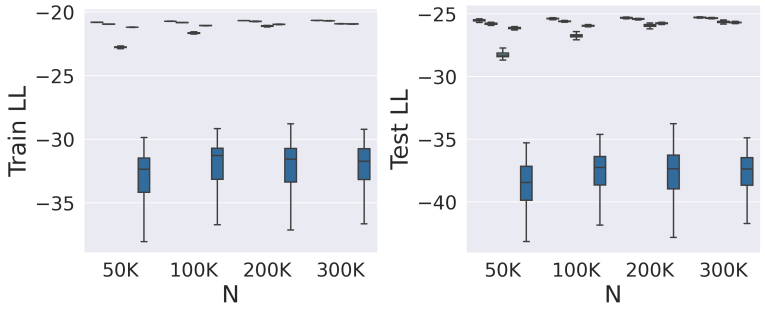
Digits	0	1	2	3	4	5	6	7	8	9
Training	34803	38049	34184	35293	33432	31067	34079	35796	33884	33720
Test	5560	6655	5888	5819	5722	5539	5858	6097	5695	5813

Each image is a 128×128 pixel greyscale matrix whose entries are real values between 0 and 1 that record the darkness of the corresponding pixels. A darker pixel has a value closer to 1. Following the common practice, we first train a 5-layer convolutional neural network and reduce each image to a $d = 50$ feature vector of real values. The details of the neural network for the dimension reduction are given in Appendix B.2. A naïve approach to build a classifier is to regard the features of each digit as a random sample from a distinct Gaussian distribution. The pooled data is therefore a sample from a finite Gaussian mixture of order $K = 10$. We may learn this model based on the whole dataset or through split-and-conquer approaches.

We randomly select $R = 100$ datasets of size $N = 50K$ from the training set. Each dataset is then randomly partitioned into $M = 10$ subsets. We obtain global, GMR, Median, KLA, and Coreset estimates for a Gaussian mixture of order 10 on each dataset. The size of the generated sample for the KLA method and the coreset size in the Coreset method are both set to be 3000. This experiment is also carried out with the sample sizes $N = 100K$, $200K$, and $300K$. These mixture estimates

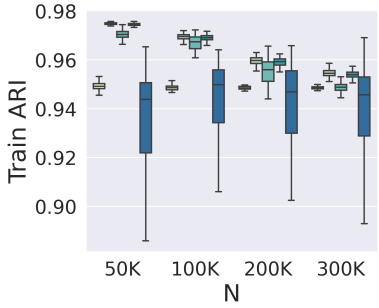
[1] Available at https://www.nist.gov/srd/nist-special-database-19.

[2] Available at https://s3.amazonaws.com/nist-srd/SD19/sd19_users_guide_edition_2.pdf.

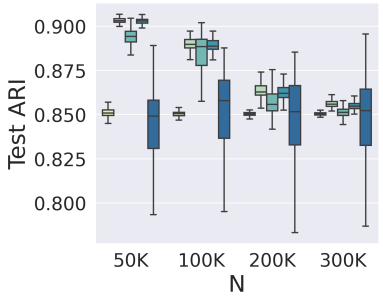


(a) Training LL

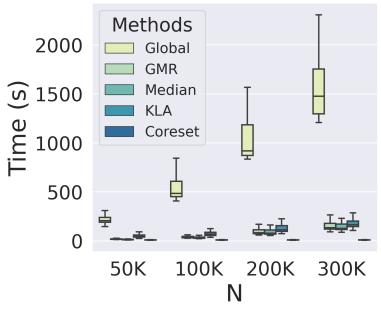
(b) Test LL



(c) Training ARI



(d) Test ARI



(e) Computation time

Figure 4.3: Performance of five estimators: Global, GMR, median, KLA, and Coreset from left to right in each block of 5 in terms of (a) training LL, (b) test LL, (c) training ARI, (d) test ARI, and (e) computational time for learning of 50-dimension order $K = 10$ Gaussian mixture for NIST digit classification. For LL and ARI, the higher the better.

are then used to cluster images of handwritten digits in the training and test sets. We show the boxplots of the LL values based on the training dataset and the test dataset in Figure 4.3(a) and Figure 4.3(b) respectively. The ARI between the true label of the image and the predicted label based on (2.1) is respectively given in Figure 4.3(c) and Figure 4.3(d).

In terms of the LL value, the proposed GMR approach attains the highest log-likelihood among all split-and-conquer approaches. The performance of Coreset estimator is far behind. In this experiment, the number of local machine is fixed at $M = 10$ and the sample sizes are from $50K$ to $300K$. Increasing the sample size benefits median estimator most notably. This is because the local sample size increases as the total sample size increases. When the total sample size is $300K$, the number of samples used to fit the local models and the samples generated to fit the aggregated model in the KLA approach are the same. The LL value of median estimator and the KLA estimator are about the same.

In terms of clustering performance, the global estimator surprisingly performs noticeably worse than the split-and-conquer approaches. The high LL value of the global estimator does not help. A likely explanation is that a GMM of order 10 is merely a **working** model rather than the **true** model, whereas true models are used in simulated data. This eliminates the advantage of the global estimator. This can also be seen that an increased total sample size N does not lead to an improved fit in general. The ARIs of all approaches get worse when the sample size increases. We think that the damage of the model-misspecification is more severe when the sample size is large. Nevertheless, the proposed GMR method has the best performance in all cases. It has the highest average ARI values and smaller variations.

Figure 4.3(e) gives the computational time. All split-and-conquer approaches save computational time. The Coreset estimator is most time efficient, the GMR and median estimators takes slightly longer and the KLA takes the longest.

4.5.3 Applications in Atmospheric Data Analysis

In this Section, we follow Chen et al. (2013) and show an example of using the distributed learning for the application of clustering in large-scale spatio-temporal

data.

We apply the proposed GMR approach to fit a finite GMM to an atmospheric dataset [3] named *CCSM run cam5.1.amip.2d.001* following Chen et al. (2013). These data are computer simulated based on Community Atmosphere Model version 5 (CAM5). The dataset contains daily observations of multiple atmospheric variables between years 1979 and 2005 over 192 longitudes (lon), 288 latitudes (lat), and 30 vertical altitude levels (lev). There variables are included: the moisture content (Q), temperature (T), and vertical velocity (Ω, OMEGA) of the air.

For ease of comparison, we analyze only observations in December, January, and February, i.e., winter in the northern hemisphere. The number of days is thus $2,430$, and the restriction reduces the variation in the dataset. At each surface location, we filter out non-wet days (less than 1 mm of daily precipitation) and focus on days with precipitation above the 95th wet-day percentile. This step reduces the number of observations at each location, not necessarily evenly. The analysis aims to cluster the locations according to the multivariate variable of dimension $d = 91$: 30 lev $\times$ $\{$Q, T, $\Omega\}$ plus the daily precipitation (PRECL) at the surface. Following Chen et al. (2013), we fit a finite GMM of order $K = 4$. They suggest that this model is helpful in identifying modes of extreme precipitation in 3D atmospheric space over a few atmospheric variables.

After this pre-processing, the dataset still takes about 3 GB of memory, so we cannot learn a global mixture in a reasonable time. We partition the dataset evenly into $M = 128$ subsets and apply the proposed GMR approach with the same numerical strategies as in the NIST experiments. For comparison, we also aggregate the local estimates by the KLA with 500 observations generated from each local estimate.

Once a finite GMM is learned, we cluster the observations based on (2.1). Each combination of day and surface location is clustered into one of four subpopulations. To visualize the clusters, we further allocate each surface location to the cluster in which it belongs on most days. Figure 4.4 shows the geographical distribution of these four clusters represented by different colours. Similar to Chen et al. (2013), the GMR clusters reveal a strong latitudinal structure, they clearly

[3] Available at https://www.earthsystemgrid.org/dataset.

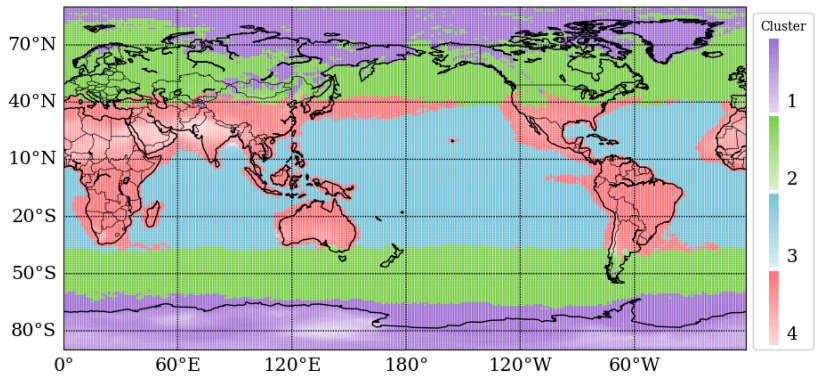


Figure 4.4: Surface locations coloured by clusters. The clusters are obtained based on a mixture fitted with 91 atmospheric features at surface locations around the world. Within each cluster, the darker the colour, the more wet days at that location.

separate the frigid, temperate, and tropical zones. The KLA results in similar clusters. Unlike Chen et al. (2013), the proposed GMR clusters are able to separate the continental and oceanic areas in the temperate zone.

4.6 Extension Where Data are Not Split at Random

In the split-and-conquer learning approach described so far, we assume that the full dataset is split completely at random into M machines. What happens if the observations are not split completely at random? In this case, the population distributions at each local machine may be different although the full dataset is an IID sample from a single model. The data at a local machine can be overly represented by some subpopulations but completely absent of data from other subpopulations. Suppose we proceed to use the proposed GMR estimator. Does this estimator leads to a sensible estimate of the overall mixture? We empirically study the performance of the aggregated estimators in this scenario in this section.

Simulation Setting

We simulation data from the mixtures in the same way as in Section 4.5.1. For the ease of computation, we only consider the case where the total sample size

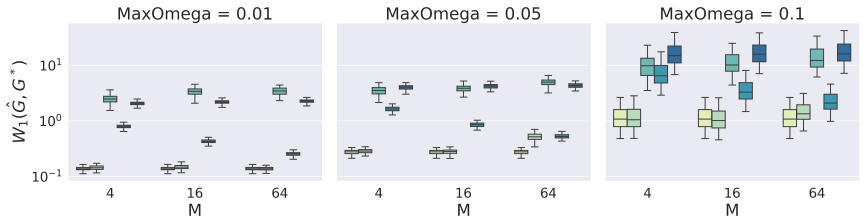
$N = 2^{17}$. We use the following procedure to partition the samples onto $M = 4$, 16, and 64 local machine respectively. As described in Section 2.1, the subpopulation identity of each randomly generated sample from a mixture is known. We first sort the generated random samples based on their subpopulation identities in an increasing order. For example, all random samples from the $(k-1)$th subpopulation are followed by all random samples from the kth subpopulation. We then split the sorted dataset in order into M subsets with equal sizes. With this split, the observations on each local machine may only from a subset of the subpopulations of the mixture. As M increases, each local machine has information about fewer subpopulations.

The same five estimators in Section 4.5.1 are considered in the experiment. Due to non-random split, the population mixture on each local machine may not be the same as the true mixture. Therefore, to compute the local estimators, we do not use the true mixture to initialize the EM algorithm and use *kmeans++* to generate 10 initial values with the warm up strategy as described in Section 4.5. The rest of the setting is the same as that in Section 4.5.

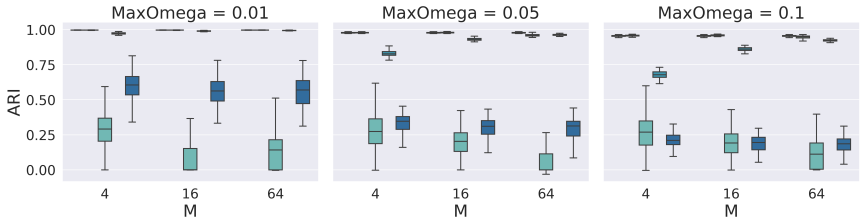
Simulation Results

We use the same set of metrics as in Section 4.5.1 to measure the performance of the estimators. The simulation results are given in Figure 4.5.

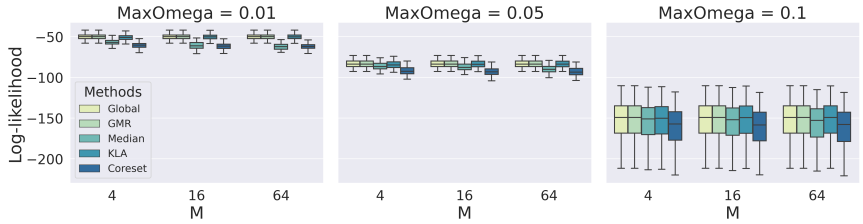
In terms of the W1 distance, ARI, and LL, it can be seen that the performance of the Median estimator is clearly deteriorated under this setting, comparing to the results in Figure 4.1. The poor performance is due to the fact that the population mixture is different from the true mixture due to the non-random split. The other three estimators based on the aggregation are quite robust to the violation of the completely random split. The performance of the GMR estimator is still almost as good as the global estimator, both the KLA estimator and the Coreset estimator have worse performance. However, we notice that when the number of local machine is $M = 64$, the performance of the GMR estimator is slightly worse than the Global estimator in terms of W_1 and ARI, especially when the degree of overlap is large. With $N = 2^{17}$ and $M = 64$, it is very likely that the local machine only contains observations from a single component.



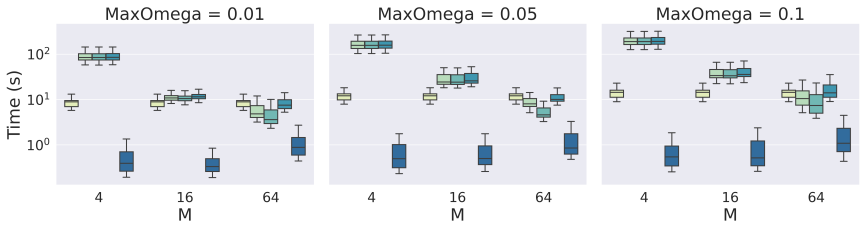
(a) W1 distance



(b) ARI



(c) Log-likelihood



(d) Computational time

Figure 4.5: Comparison of Global, Median, GMR, KLA, and Coreset estimators in terms of distance between learned and true mixing distributions in (a), the similarity of clustering outcomes based on learned and true mixture in (b), the log-likelihood per observation based on the full dataset in (c), and the computational time in (d), for distributed learning of 50-dimensional order 5 mixture under non-random data split when $N = 2^{17}$.

In terms of the computational time, since we initialize the EM algorithm for computing the Global estimator with the true value and the warm up strategy for the local estimators, then computational time for the local estimators are about 10 times longer than the Global estimator. If the same initialization strategy is used for all estimators, then the split-and-conquer learning based estimators can save the computational time than the Global estimator.

Summary

Based on the simulation results, all the split-and-conquer based approaches are robust to the violation of the assumption of splitting completely at random, under finite Gaussian mixtures. Among all these approaches, the proposed GMR estimator is still the most efficient estimator. Although the Coreset estimator is computationally fast, its statistically efficiency is compromised. Although the Median, GMR, and KLA estimators take longer time to compute than the Global estimator in general. This observations, however, is dependent on the choice of the initial values in the EM algorithm. If the EM algorithm is run with multiple initial values for the Global estimator, the split-and-conquer approach is still more time efficient than the Global estimator.

4.7 Discussion on the Known Order Assumption

In the split-and-conquer learning of GMM described above, we assume the order of the mixture is known and correctly specified. Every local machine learns a finite Gaussian mixture with the same and correct order. While the machine learning community has devoted most energy to this special case, it is of interest and great importance to develop the split-and-conquer approaches when the order of the mixture model is potentially over-specified.

In this section, we empirically evaluate the performance of a number of split-and-conquer approaches under this case. We simulate data from a mixture of order K and have the simulated data partitioned into M subsets completely at random. On each local machine, we learn the mixture model based on the allocated subset under four scenarios.

i) the order of the mixture is correctly specified;

ii) the order of the mixture on the local machine is specified to be $K + 1$;

iii) the order of the mixture on the local machine is specified to be $K + 2$;

iv) the order of the mixture on the mth local machine is specified to be $K + m$.

In the last scenario, the order of the mixture model varies with the local machines, and we refer to this scenario as mixed order hereafter. In the aggregation step, we first combine the local estimates to form $\phi(x; \overline{G})$ and the order of this mixture is greater than MK under cases ii) – iv).

Estimators for comparison

We use the reduction approach in (4.4) to reduce the order of $\phi(x; \overline{G})$ to true order K with various choices of the divergence $\rho(\cdot, \cdot)$. The divergences considered in the experiment are:

1. ***ISE***. The ISE between two mixtures, that is

$$\rho(G_1, G_2) = D_{\text{ISE}}(\Phi(\cdot; G_1), \Phi(\cdot; G_2)).$$

2. ***CTD-KL***. The CTD with the cost function being the KL divergence between two Gaussians. That is

$$\rho(G_1, G_2) = \mathcal{T}_{D_{\text{KL}}}(\Phi(\cdot; G_1), \Phi(\cdot; G_2)).$$

3. ***CTD-ISE***. The CTD with the cost function being the ISE between two Gaussians. That is

$$\rho(G_1, G_2) = \mathcal{T}_{D_{\text{ISE}}}(\Phi(\cdot; G_1), \Phi(\cdot; G_2)).$$

We also include a KLA approach of Liu and Ihler (2014) in the simulation study. The Coreset method is not applicable under this setting, we therefore do not include this method for comparison in our experiment. We use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm to compute ISE estimator, the CTD-ISE can also be computed numerically via the MM algorithm, we describe the details of their computation in Chapter 5.

Data Generation

We generate data from the following four mixtures in our experiment. Since the ISE estimator is computationally difficult for high dimension d, we only generate random samples from $d = 1$ and $d = 2$. The first two are chosen as Gaussian mixtures of order $K = 3$ and dimension $d = 1$. Their density functions are given by

I. $\phi(x; G) = 1/3\phi(x; -3, 1) + 1/3\phi(x; 0, 1) + 1/3\phi(x; 3, 1)$;

II. $\phi(x; G) = 0.1\phi(x; -2, 1) + 0.3\phi(x; 0, 1) + 0.6\phi(x; 3, 1)$.

The next two are chosen as a Gaussian mixtures of order $K = 3$ and dimension $d = 2$. To introduce the density function of these two mixtures, we first denote $\mu(r, \theta) = r(\cos\theta, \sin\theta)^\top$ and

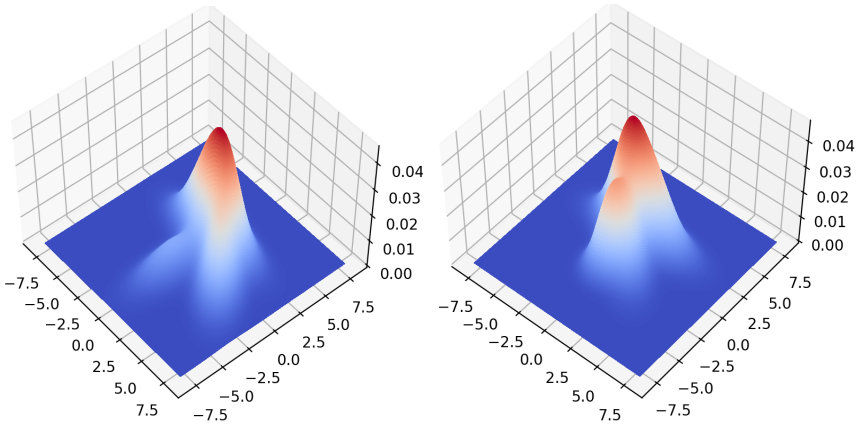
$$\Sigma(\lambda_1, \lambda_2, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}^\top.$$

III. The mixing weights are 0.15, 0.35, and 0.5 respectively. The subpopulation means are $\mu(2, 3\pi/2)$, $\mu(3, 0)$, and $\mu(2, \pi/2)$. The subpopulation covariances are $\Sigma(1, 5, 0)$, $\Sigma(1, 5, \pi/4)$, and $\Sigma(1, 5, \pi)$.

IV. The mixing weights are 0.15, 0.35, and 0.5 respectively. The subpopulation means are $\mu(2, 3\pi/2)$, $\mu(0, 0)$, and $\mu(2, \pi/2)$. The subpopulation covariances are $\Sigma(1, 1, 0)$, $\Sigma(1, 5, \pi/4)$, and $\Sigma(1, 5, \pi)$.

The density function of the two mixtures in III and IV are visualized in Figure 4.6.

We generate samples of sizes $N = 2^{19}$ or $N = 2^{21}$ respectively from each of the mixtures given above. Each sample is then split into $M = 4$ or $M = 8$ subsets completely at random and they are regarded as stored on $M = 4$ or $M = 8$ local machines. With these two choices of sample sizes and two choices of the number of local machines, we obtain 4 combinations. The four split-and-conquer methods in Section 4.7 are applied to obtain the aggregated estimates. When the local mixtures are over-specified, we use the *kmeans++* to generate 10 initial values for the EM



(a) III **(b)** IV

Figure 4.6: The density function of two 3-component mixtures in 2 dimensional in III and IV.

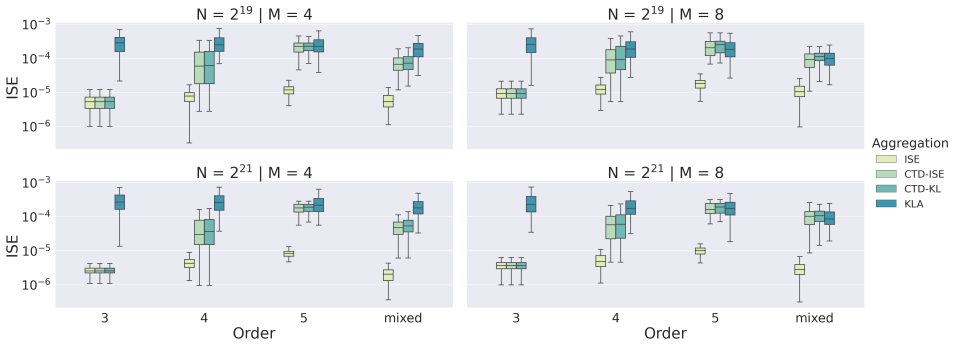
algorithm with the warm up strategy as described in Section 4.5. In the aggregation step with the reduction approach, we use the true value to initialize the algorithm. The rest of the setting is the same as that in Section 4.5.

Simulation Results under Distributions I and II

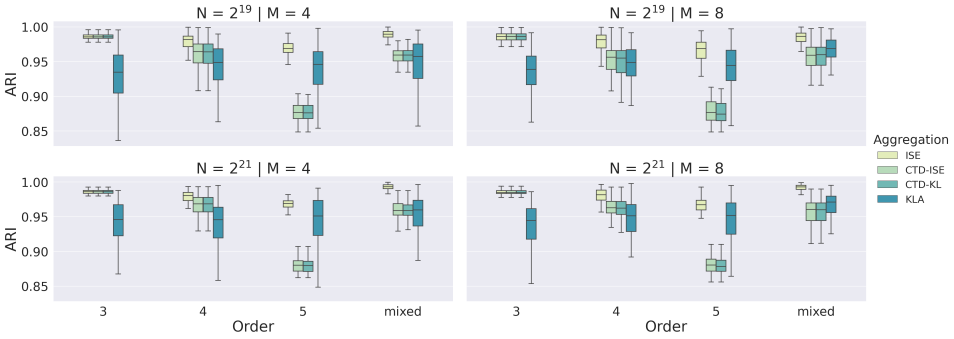
We summarize the results in Figure 4.7 and Figure 4.8 when the split-and-conquer methods are applied to data generated from distributions I and II. Note the plots in the first and second columns are results under distributions I and II respectively. We then divide each plot into 4 panels labeled by $M = 4$ or $M = 8$ on the top, and $N = 2^{19}$ and $N = 2^{21}$ on the right margin with an obvious interpretation. Within each panel are box-plots of one of the performance measures for 4 methods. The lower ISE and higher ARI indicate better performance.

According to Figure 4.7 and Figure 4.8 , when the true order $K = 3$ is specified at local machines, ISE, CTD-ISE, and CTD-KL have similar and good performances. When the number of machines increases or the sample size increases, the box-plots get shorter, indicating lower variations. In comparison, the ISE of KLA approach is hundreds of times larger. It is therefore less efficient.

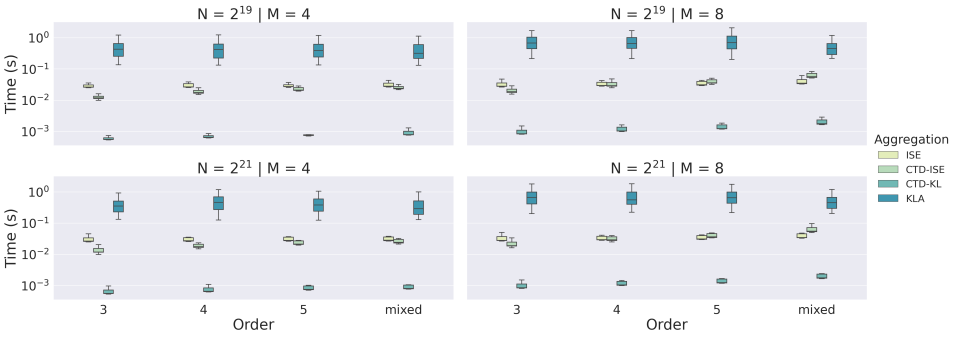
When the order on local machines is over-specified with $K = 4$, the ISE



(a) ISE

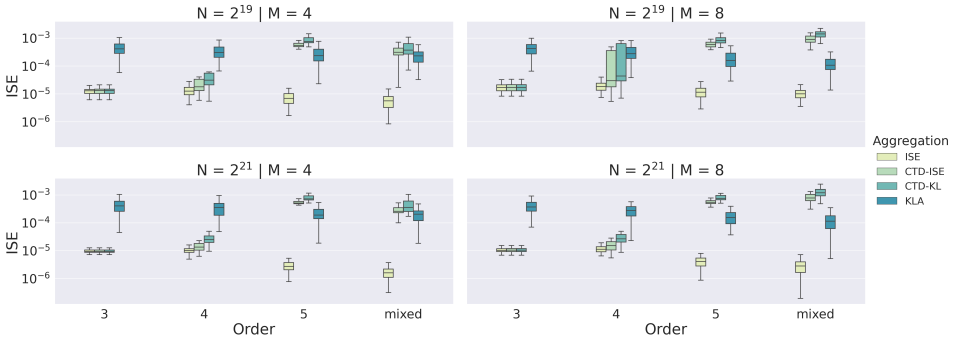


(b) ARI

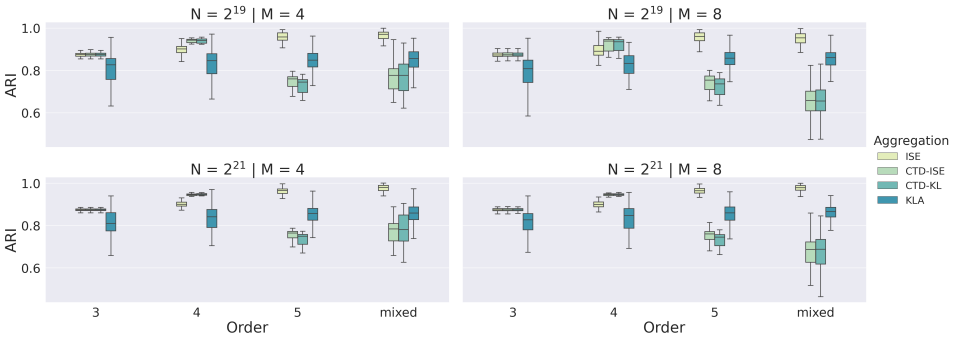


(c) Time

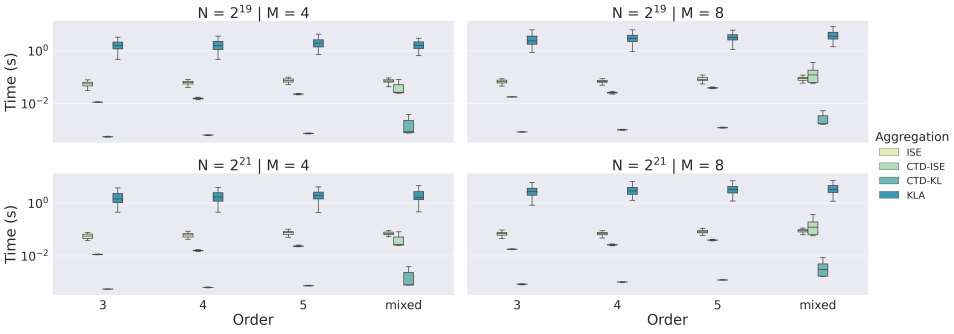
Figure 4.7: Performances of four split-and-conquer approaches for learning 1-dimensional 3-component mixture in I.



(a) ISE



(b) ARI



(c) Time

Figure 4.8: Performances of four split-and-conquer approaches for learning 1-dimensional 3-component mixture in II.

method is negatively but only mildly affected in terms of both ISE and ARI. The CTD-ISE and CTD-KL become much worse and less stable. The KLA remains non-competitive. When the order is over-specified at $K = 5$, the ISE remains well behaved. The computationally favored CTD-ISE and CTD-KL become statistically ineffective. Under the case of mixed orders, the ISE is still well behaved and other methods remain non-competitive.

Simulation Results under Distributions III and IV

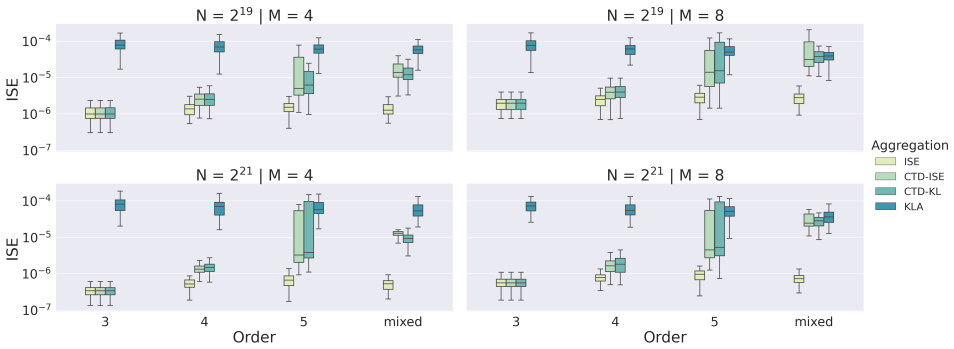
The simulation results under distributions III and IV are summarized in Figure 4.9 and Figure 4.10. The plots are arranged the same way as before.

Most inference methods have deteriorated performance on multidimensional data. It turns out that the performance of the ISE approach remains reasonable in all cases under distribution III. When the order is slightly over-specified with $K = 4$, the performances of CTD-ISE and CTD-KL also remain reasonable. When $k = 5$, these two approaches become unstable, just like their performance under distributions I and II. Under the case of mixed orders, the ISE approach is still well behaved. The other methods remain non-competitive.

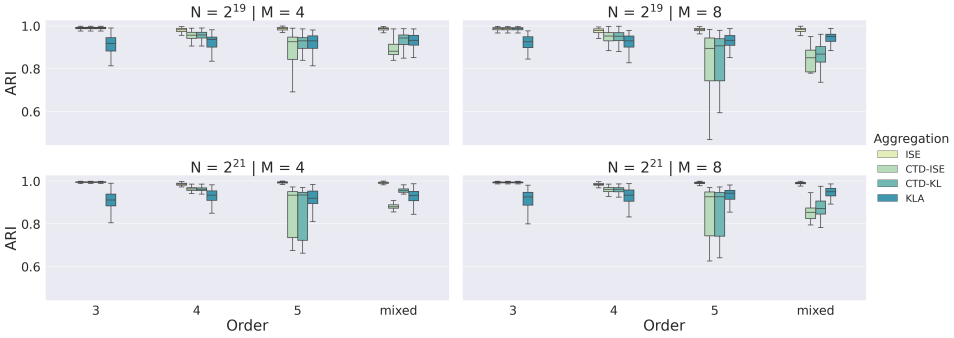
Unlike distributions I-III, the subpopulations in distribution IV are not well separated. We anticipate that all approaches do not perform too well. Indeed, ISE, CTD-ISE, and CTD-KL are all unstable even when the order is correctly specified with $K = 3$. We are surprised, however, that they recover from this failure when the order is over-specified at $K = 4$. The ISE approach has a comparable low ISE value in this case to the ISE value under distribution III, where the subpopulations are well separated.

Summary

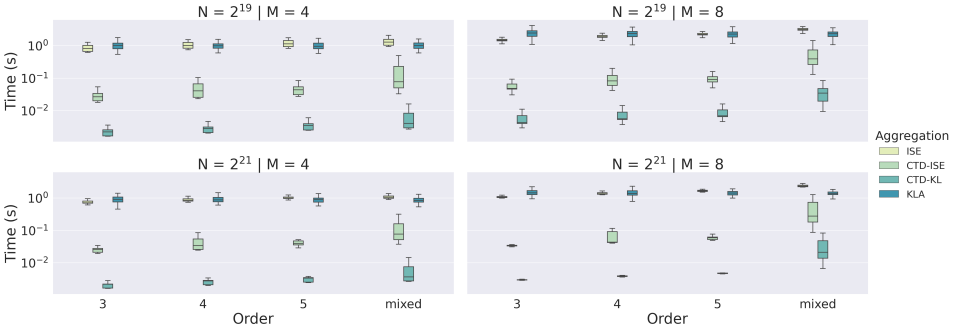
Our study reveals that there is a trade-off between robustness and computational efficiency: the computationally intensive approach is robust against over-specification, while the two computationally friendly approaches have compromised statistical performance when the order is over-specified. We believe that the information in the data on the true model is not lost in the split step of the learning. Hence, there is a good promise to develop computationally friendly aggregation strategies to ag-



(a) ISE

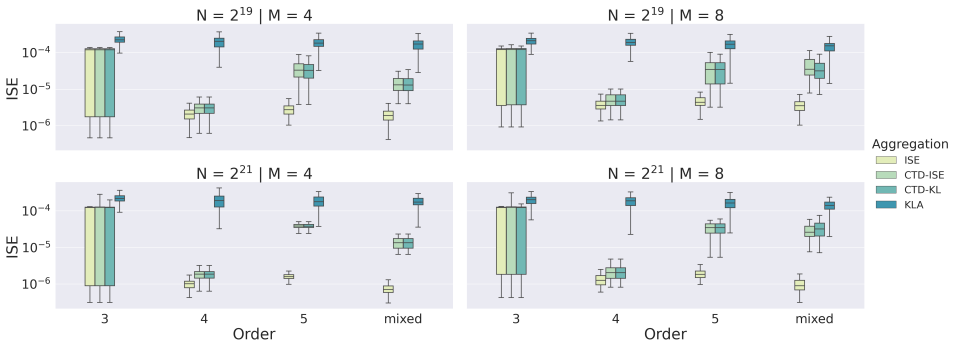


(b) ARI

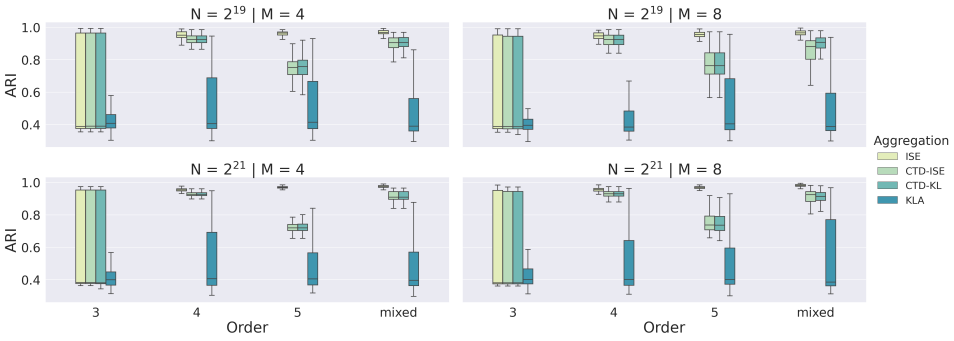


(c) Time

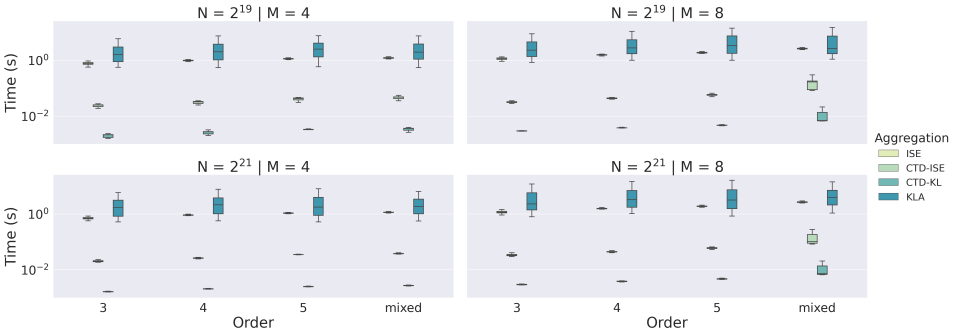
Figure 4.9: Performances of four split-and-conquer approaches for learning 2-dimensional 3-component mixture in III.



(a) ISE



(b) ARI



(c) Time

Figure 4.10: Performances of four split-and-conquer approaches for learning 2-dimensional 3-component mixture in IV.

gregate local estimates in a statistically efficient way. A full exploration of such remedies is left as future work.

Our simulation experiment only covers a small range of scenarios. It is dangerous to generalize what we have observed. It might be safe to say, the reasonable performance of the ISE approach in all situations indicates the local estimates effectively summarize the information contained in the data. The over-specification of the order may not always be devastating. Based on these results, one may decide to always use the ISE approach as it is least affected by over-specification. However, the drawback of this approach is its computational complexity. It becomes infeasible when either M or the dimension d becomes larger. The straightforward implementations of CTD-ISE and CTD-KL do not perform as well under order mis-specification. One is reminded that the motivation behind the CTD-ISE and CTD-KL is their computational efficacy, which is not shared by ISE. The superior performance of ISE in terms of the robustness against over-specification indicates that statistical efficacy is possible. With some effort, we believe robust as well as computationally efficient split-and-conquer approaches can be found. We aim to pursue this topic in the future.

4.8 Conclusion

In this chapter, we describe two potential aggregation approaches, namely the barycentre approach and the reduction approach, for the split-and-conquer learning of finite Gaussian mixtures. We show an example where the barycentre approach does not work and hence recommend using the reduction approach for aggregation. We also discuss the connection between these two approaches. Considering the computational complexity, we recommend using the CTD between mixtures as the objective function for the reduction approach. A numerically effective MM algorithm is designed for the computation. Our experiments show that our proposed GMR estimator has good performance both statistically and computationally. In the simulation study, our estimator is as good as the global estimator if the latter is feasible. We also investigate the robustness of the GMR estimator under the cases where the full dataset partition is not at random and where the locally fitted mixture is over-specified. We empirically show the GMR is as efficient as the global estima-

tor under non-random split. When locally fitted mixtures are over-specified, there is a trade-off between the statistical efficiency and computational complexity: a computationally intensive approach is robust against over-specification, while two computationally friendly approaches have compromised statistical performances. We leave this as future work to develop a computationally effective robust estimator under over-specification. We have focused on finite GMMs, but with some adjustment, our approach could be applied to learning mixtures with other subpopulation distributions such as Gamma and Poisson. We discuss the generalization to non-Gaussian mixtures in Chapter 6.

Chapter 5

Gaussian Mixture Reduction and Approximate Inference

It is often cited (Nguyen et al., 2020; Titterington et al., 1985) that there always exists a Gaussian mixture whose density function is arbitrarily close to any density function. For example, the kernel density estimate with Gaussian kernel and proper bandwidth is consistent for any continuous density function that vanishes at infinity (Wied and Weißbach, 2012). Based on this observation, GMMs are widely used to approximate distributions of complex shapes. In many applications, the target density function can be well approximated by a finite Gaussian mixture with proper order. The approximation makes the downstream data analysis computationally efficient.

In this chapter, we study the problem called Gaussian Mixture Reduction (GMR) in the machine learning community. Ignoring the application background, GMR is a procedure to approximate an order N Gaussian mixture

$$\Phi(x; G) = \sum_{n=1}^{N} w_n \Phi(x; \mu_n, \Sigma_n) := \sum_{n=1}^{N} w_n \Phi_n(x)$$

by an order M Gaussian mixture

$$\Phi(x; \widetilde{G}) = \sum_{m=1}^{M} \widetilde{w}_m \Phi(x; \widetilde{\mu}_m, \widetilde{\Sigma}_m) := \sum_{m=1}^{M} \widetilde{w}_m \widetilde{\Phi}_m(x)$$

with $M < N$. We refer to $\Phi(x; G)$ and $\Phi(x; \widetilde{G})$ as original mixture and reduced mixture respectively. Note that we use apparent notation for G, $\widetilde{G}$, μ_n, Σ_n, $\widetilde{\Sigma}_n$, and $\widetilde{\Sigma}_m$. For the rest of the dicussion, the order M is given or chosen by the user. We want to highlight that the GMR is not a statistical inference problem. It does not learn the mixing distribution G from data but instead approximates a target mixture with high order by a mixture with a lower order.

The GMR procedure is used in many machine learning applications involving recursive inference procedures (Manzar, 2017). In these cases, suppose the distribution of a statistics is a mixture of moderate order at one iteration. The corresponding distribution after another iteration can be a mixture whose order increases by a multiplication factor. Therefore, the sequence of mixtures from the recursive procedure are mixtures whose orders increase exponentially with the number of iterations. The computation of these distributions therefore quickly becomes intractable. To overcome this difficulty, the GMR procedure is applied after each iteration to approximate the mixture with a controlled order to stop the order of a mixture from increasing exponentially. As a result, the orders of the sequence of mixtures remain the same and the computation becomes feasible.

One example of using GMR is the belief propagation under a probabilistic graphical model (Sudderth et al., 2010). One task in inference under a probabilistic graphical model is to find the marginal distribution of a random variable given the joint distribution of the random variables whose dependency structure is specified by a graph. An example of a graphical model and dependency structure will be given more precisely in Section 5.1.1. Given the complicated dependency structure of these random variables, computationally and efficiently marginalizing over other random variables involves a recursive procedure called belief propagation. The messages, which are distributions that contain the "influence" that one variable exerts on another, are updated iteratively in the belief propagation. The flexibility of the Gaussian mixture makes it an ideal choice to approximate the initial message in belief propagation. After an iteration, the updated message remains a Gaussian mixture but its order is increased by some multiplication factor. Hence the order of the message increases exponentially and quickly becomes intractable. The GMR can be used after each iteration so that the message remains in a controlled order and manageable computational cost.

Another example of using GMR is in recursive Bayesian filtering under a hidden Markov model (HMM). In the Bayesian filtering of an HMM, the finite Gaussian mixtures can be used to approximate the transition distribution and the marginal distribution. Consequently, the posterior distribution (the distribution of hidden states given the observed value up to the current time point) is also a Gaussian mixture. However, the order of the mixture increases exponentially with time. To control the computation costs, some intermediate approximation steps are often introduced to prevent the order of the posterior mixture from exploding. Similarly, the GMR is helpful for this purpose.

There has been a rich literature on GMR and most approaches are one of three general types: *greedy algorithm-based* (Huber and Hanebeck, 2008; Runnalls, 2007; Salmond, 1990), *optimization-based* (Williams and Maybeck, 2006), and *clustering-based* (Assa and Plataniotis, 2018; Davis and Dhillon, 2007; Goldberger and Roweis, 2005; Schieferdecker and Huber, 2009; Vasconcelos and Lippman, 1999; Yu et al., 2018; Zhang and Kwok, 2010). The greedy algorithms either merge two components or prune a component at a time and repeat until the desired number of components (order) is obtained. These approaches are ad hoc and usually do not have an ultimate optimality target. The optimization-based approaches such as in Williams and Maybeck (2006) have an explicit optimality target but can be computationally difficult. The clustering-based approaches are motivated by the *k-means* algorithms in the Euclidean space and are computationally efficient in general. To the best of our knowledge, it is unsure in the literature whether the clustering-based algorithms should converge or whether they have attained some optimality target when they converge.

In this chapter, we propose an optimization-based approach for GMR. The proposed reduced mixture minimizes a Composite Transportation Divergence (CTD) between the original mixture and a mixture with the desired order. We develop an Majorization Maximization (MM) algorithm to efficiently solve the corresponding numerical optimization problem. Our MM algorithm resembles the existing clustering-based algorithm and is computationally efficient. Its clear optimality target enables us to show the algorithm converges under some conditions on the CTD. Since our MM algorithm includes the existing clustering-based algorithms as special cases, our results reveal the missing *optimality target*s of the existing

clustering-based algorithms and establish their algorithmic convergence in general.

The rest of the chapter is organized as follows. In Section 5.1, we present the details of the approximate inference with GMR in belief propagation and filtering under Hidden Markov Model (HMM). In Section 5.2, we review existing GMR approaches in the literature. The proposed method is given in Section 5.3. We show that existing clustering-based approaches are special cases of our proposed method. Numerical experiments comparing different approaches are given in Section 5.4. Conclusions and discussions are given in Section 5.5.

5.1 Application Examples of Gaussian Mixture Reduction

The density functions of Gaussian mixtures are often used to approximate density functions of complex shapes in statistical inference. In this section, we give some examples of using GMR. In Section 5.1.1, we explain the approximate inference in the belief propagation under probabilistic graphical models. In Section 5.1.2, we describe the tracking procedure under hidden Markov models.

5.1.1 Belief Propagation under Graphical Models

A graph consists of a node set $\mathcal{V}$ and an undirected edge set $\mathcal{E}$ made of pairs of nodes that are related. A probabilistic graphical model associates each node with a random variable, say X_i, and postulates that the joint density function of the random vector $X = \{X_i : i \in \mathcal{V}\}$ can be factorized into

$$f_X(x) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i) \tag{5.1}$$

for some non-negative valued functions $\psi_{ij}(\cdot,\cdot)$ and $\psi_i(\cdot)$. We call $\psi_{ij}(\cdot,\cdot)$ and $\psi_i(\cdot)$ local potential and local evidence potential respectively. Note that the factorization is determined by the dependency of these variables that is characterized by the graph.

One task in the inference under probabilistic graphical model is to evaluate the marginal density function of X_i for $i \in \mathcal{V}$ given the factorization in (5.1). For example, the graphical model is applied to kinematic tracking problems in computer

vision (Sudderth et al., 2010). The task of kinematic tracking is to estimate the motion of each part of an articulated object based on recorded video sequences. The tracking of human hand gestures, which is used as a natural human-computer interface device, is one of the applications of kinematic tracking (Wu and Huang, 2001). To simplify the problem, discretization is used and only several rigid bodies of the hand need to be tracked. For example, Sudderth et al. (2010) only tracks the position of little circles marked on the hand as shown in the left plot in Figure 5.1. In these applications, the random variable X_i is a 4-dimensional vector

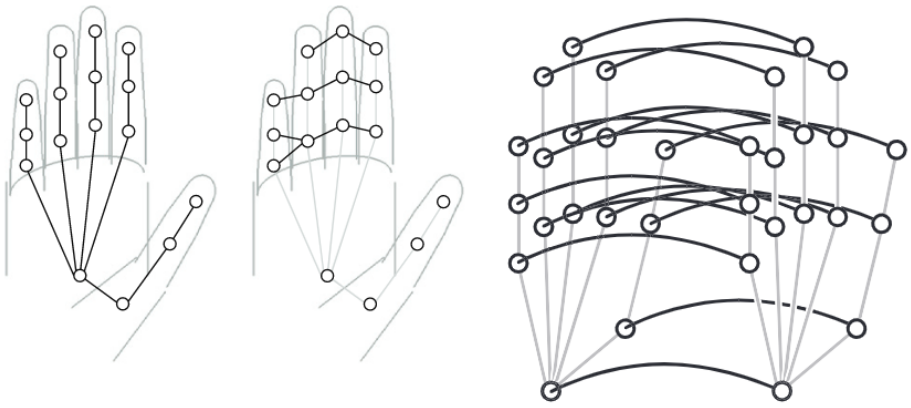


Figure 5.1: Graphical models capturing the kinematic, structural, and temporal constraints relating the hand's 16 rigid bodies. The images is taken from Sudderth et al. (2010).

that records the 3D spatial location and the angle of the rotation of each of the bodies of the hand. Figure 5.1 depicts the graphical model used in Sudderth et al. (2010). The left plot shows the pairwise potentials that capture kinematic constraints that phalanges are connected by revolute joints. The middle plot shows the pairwise potentials that capture structural constraints of different fingers. The right plot shows pairwise potentials that capture the Markov temporal dynamics of the hand at two consecutive frames of the video. Given the graphical model, the task of tracking is to find the marginal distribution of each body of the hand X_i and predict the spatial location of these bodies over time.

To find out the marginal distribution of X_i, a computationally efficient procedure called Belief Propagation (BP) is proposed by Yedidia et al. (2003). We

describe the BP algorithm and show the motivation of using GMR in the inference.

The BP algorithm works by passing density functions called messages along with the edges between the nodes in the graph. These messages contain the "influence" that one variable exerts on another. More precisely, let the neighbourhood of a node i be denoted as $\Gamma(i) = \{j : (i,j) \in \mathcal{E}\}$. The message from node i to node $j \in \Gamma(i)$ is a density function that is denoted as $m_{ji}(\cdot)$. Given messages $m_{ji}^{(t-1)}(\cdot)$ at the $(t-1)$th iteration, the BP algorithm updates them according to

$$m_{ji}^{(t)}(x) \propto \int \{\psi_{ij}(x, x_j)\psi_j(x_j) \prod_{k \in \Gamma(j)\setminus i} m_{kj}^{(t-1)}(x_j)\}\, dx_j \tag{5.2}$$

in the next iteration. A belief is the tentative marginal density function $q_i^{(t-1)}(x)$ of X_i up to some normalization constant. Given the messages, the BP algorithm updates the beliefs by

$$q_i^{(t)}(x) \propto \psi_i(x) \prod_{j \in \Gamma(i)} m_{ji}^{(t)}(x). \tag{5.3}$$

The messages and beliefs are iteratively updated until convergence. For acyclic or tree-structured graphs, the updates lead to a sequence of beliefs that converges to the density function of the marginal distribution. For graphs with loops, the BP sequence provides a useful approximation. The derivation and justification of the message passing are very complex, see Yedidia et al. (2003) for details.

Closed-form outcomes of the messages do not exist in general but with some exceptions. When all local potential ψ_i and local evidence potential ψ_{ij} are Gaussian densities, then by the property of Gaussian distribution in Section (2.7)

$$\int \phi(x; \mu_1, \Sigma_1)\phi(x; \mu_2, \Sigma_2)\, dx = \phi(\mu_1; \mu_2, \Sigma_1 + \Sigma_2),$$

the messages and the beliefs updated using (5.2) and (5.3) remain Gaussian with closed-form new parameter values. However, the Gaussian distributions are not flexible enough to capture the shape of the marginal densities. To take advantage of the property of the Gaussian distribution while permitting flexible density shapes, the Gaussian mixtures can be used to approximate local potential $\psi_{ij}(\cdot)$ and lo-

cal evidence potential $\psi_i(\cdot)$. Subsequently, the messages $m_{ji}^{(t)}$ and the beliefs $q_i^{(t)}$ are Gaussian mixtures whose parameter values have closed-form. However, the orders of the messages and beliefs increase exponentially with t and the inference quickly becomes intractable. To solve this issue, one may use the GMR technique to reduce the order of the mixture before the next update to stop it from increasing exponentially. We apply the proposed GMR to BP in Section 5.4.2.

5.1.2 Tracking under Hidden Markov Models

In this section, we show that the GMR is also used for tracking under hidden Markov models. We first introduce the tracking under hidden Markov models and then show how the GMR is used under tracking.

The HMM is defined as follows.

Definition 5.1. *A hidden Markov model characterizes two discrete time series* $\{X_t\}_{t\geq 0}$ *and* $\{Y_t\}_{t\geq 0}$.

1. *The unobserved state space* $\{X_t\}_{t\geq 0}$ *is a Markov process. The Markov process has the property that the conditional distribution of* X_t *given* $X_{0:t-1} = x_{0:t-1}$ *is the same as the conditional distribution of* X_t *given* $X_{t-1} = x_{t-1}$. *We denote the distribution of* X_0 *with density function* $p_0(x)$ *and the transition density of* X_{t+1} *given* $X_t = x_t$ *by* $p_{t+1}(x|x_t)$.

2. *The series* $\{Y_t\}_{t\geq 0}$ *is an observed series. We denote the conditional distribution of* Y_t *given* $X_t = x_t$ *by* $g_t(\cdot|x_t)$.

Based on the definition of an HMM, it can be noted that the conditional distribution of Y_t given $X_{0:t} = x_{0:t}$ is the same as its conditional distribution given $X_t = x_t$.

One HMM example is the *Linear Gaussian Model* specified by

$$X_{t+1} = F_t X_t + G_t W_t$$
$$Y_t = H_t^\top X_t + V_t$$

where $X_0 \sim \phi(x; \mu_0, P_0)$, $V_t \overset{\text{i.i.d.}}{\sim} \phi(x; 0, R_t)$, $W_t \overset{\text{i.i.d.}}{\sim} \phi(x; 0, Q_t)$, and all unspecified quantities are non-random matrices of appropriate nature and dimensions. The

linear Gaussian model is widely used for target tracking and signal processing (Anderson and Moore, 2012). Another HMM example is *Stochastic Volatility model*. This model and its generalizations are widely used in economics and mathematical finance (Taylor, 1994). The stochastic volatility model is specified by

$$\begin{aligned} X_{t+1} &= \alpha X_t + \sigma W_t \\ Y_t &= \beta \exp(X_t) V_t \end{aligned}$$

where $X_0 \sim \phi(x; 0, \sigma^2/(1-\alpha^2))$, $V_t \overset{\text{i.i.d.}}{\sim} \phi(x; 0, 1)$, $W_t \overset{\text{i.i.d.}}{\sim} \phi(x; 0, 1)$, and all unspecified quantities are non-random parameters. Compare to the linear Gaussian model, the relationship between Y_ts and the latent X_ts is nonlinear under the stochastic volatility model.

Under the HMMs, one general inference problem is *tracking*: inferring the value of the latent variable at the current moment given all the observations up to this moment. It is called "tracking" since we are interested in keeping track of the "location" of the system given noisy observations. Mathematically, it is to determine the conditional distribution of X_t given $Y_{0:t} = y_{0:t}$ namely $\{p_{X_t|Y_{0:t}}(x_t|y_{0:t})\}_{t\geq 0}$.

Under linear Gaussian models, the tracking is analytically tractable, and the procedure is usually named the Kalman filter (Meinhold and Singpurwalla, 1983). In general, the filtering is done by a recursive procedure described as follows (Doucet and Johansen, 2009, Page 4–5). The conditional joint distribution of all hidden variables $X_{0:t}$ given the observations $y_{0:t}$ can be written as

$$p_{X_{0:t}|Y_{0:t}}(x_{0:t}|y_{0:t}) = \frac{p_{X_{0:t}}(x_{0:t}) p_{Y_{0:t}|X_{0:t}}(y_{0:t}|x_{0:t})}{\int p_{X_{0:t}}(\tilde{x}_{0:t}) p_{Y_{0:t}|X_{0:t}}(y_{0:t}|\tilde{x}_{0:t})\, d\tilde{x}_{0:t}} \tag{5.4}$$

where the prior on the hidden variables are given by

$$p_{X_{0:t}}(x_{0:t}) = p_0(x_0) \prod_{k=1}^{t} p_k(x_k|x_{k-1})$$

and likelihood of $x_{0:t}$ is given by

$$p_{Y_{0:t}|X_{0:t}}(y_{0:t}|x_{0:t}) = \prod_{k=0}^{t} g_k(y_k|x_k)$$

based on Definition 5.1.

The joint density function $p_{X_{0:t},Y_{0:t}}(x_{0:t}, y_{0:t})$ of $(X_{0:t}, Y_{0:t})$ has decomposition

$$p_{X_{0:t},Y_{0:t}}(x_{0:t}, y_{0:t}) = p_{X_{0:t-1},Y_{0:t-1}}(x_{0:t-1}, y_{0:t-1}) p_t(x_t|x_{t-1}) g_t(y_t|x_t).$$

Along with (5.4), the posterior $p_{X_{0:t}|Y_{0:t}}(x_{0:t}|y_{0:t})$ consequently satisfies the recursion relationship

$$p_{X_{0:t}|Y_{0:t}}(x_{0:t}|y_{0:t}) = p_{X_{0:t-1}|Y_{0:t-1}}(x_{0:t-1}|y_{0:t-1}) \frac{p_t(x_t|x_{t-1}) g_t(y_t|x_t)}{p_{Y_t|Y_{0:t-1}}(y_t|y_{0:t-1})}$$

where

$$p_{Y_t|Y_{0:t-1}}(y_t|y_{0:t-1}) = \int p_{X_{t-1}|Y_{0:t-1}}(x_{t-1}|y_{0:t-1}) p_t(x_t|x_{t-1}) g_t(y_t|x_t)\, dx_{t-1}\, dx_t.$$

Integrating out $x_{0:t-1}$, we find the filtering satisfies the recursion

$$p_{X_t|Y_{0:t}}(x_t|y_{0:t}) = \frac{p_{X_t|Y_{0:t-1}}(x_t|y_{0:t-1}) g_t(y_t|x_t)}{p_{Y_t|Y_{0:t-1}}(y_t|y_{0:t-1})} \tag{5.5}$$

where

$$p_{X_t|Y_{0:t-1}}(x_t|y_{0:t-1}) = \int p_t(x_t|x_{t-1}) p_{X_t|Y_{1:t-1}}(x_t|y_{1:t-1})\, dx_{t-1}. \tag{5.6}$$

Equation (5.6) is called the prediction step and (5.5) is called the update step.

The prediction and update steps are used iteratively for tracking under HMMs. For general conditional density $g_t(\cdot)$ and conditional density $p_t(\cdot)$, both steps do not have closed-form outcomes and numerical approaches are needed. One numerical approach is functional approximation: approximating $p_t(\cdot)$ and $g_t(\cdot)$ by Gaussian mixtures. The choice of GMMs is due to their ability to approximate

any density function to arbitrary precision and having closed-form outputs in the prediction and update steps. Suppose

$$p_t(x_t|x_{t-1}) = \sum_{i=1}^{n_{ti}} w_{ti}\phi(x_t; A_{ti}x_{t-1} + \mu_{ti}, \Sigma_{ti}),$$
$$g_t(y_t|x_t) = \sum_{j=1}^{m_{tj}} \pi_{tj}\phi(y_t; B_{tj}x_t + u_{tj}, \Lambda_{tj})$$

for some non-random matrices, vectors, constants A_{ti}, Σ_{ti}, B_{tj}, Λ_{tj}, μ_{ti}, u_{tj}, w_{ti}, and π_{tj}. The numbers n_{ti} and m_{tj} are the orders of the mixtures of the conditional densities.

Let the tth step of the recursion be

$$p_{X_{t-1}|Y_{0:t-1}}(x_{t-1}|y_{0:t-1}) = \sum_{k=1}^{N_{tk}} \omega_k \phi(x_{t-1}; \mu_k^\dagger, \Sigma_k^\dagger)$$

for some $\mu_k^\dagger$, $\Sigma_k^\dagger$, and $k \in [N_{tk}]$ where N_{tk} is a known integer. Then the prediction step and update step are given as follows.

Prediction step:

$$p_{X_t|Y_{0:t-1}}(x_t|y_{0:t-1}) = \sum_{i=1}^{n_{ti}} \sum_{k=1}^{N_{tk}} w_{ti}\omega_k \phi(x_t; \mu_{tik}, \Sigma_{tik}) \tag{5.7}$$

where $\mu_{tik} = A_{ti}\mu_k^\dagger + \mu_{ti}$ and $\Sigma_{tik} = \Sigma_{ti} + A_{ti}\Sigma_k^\dagger A_{ti}^\top$.

Update step:

$$p_{X_t|Y_{0:t}}(x_t|y_{0:t}) = \sum_{i=1}^{n_{ti}} \sum_{j=1}^{m_{tj}} \sum_{k=1}^{N_{tk}} w_{tijk}\phi(x_t; \mu_{tijk}, \Sigma_{tijk}) \tag{5.8}$$

where $w_{tijk} = \pi_{tijk} / \sum_{i,j,k} \pi_{tijk}$, $\pi_{tijk} = w_{ti}\pi_{tj}\omega_k C_{tijk}$, and

$$C_{tijk} = \phi(y_t; B_{tj}\mu_{tik} + u_{tj}, \Lambda_{tj} + B_{tj}\Sigma_{tik}B_{tj}^\top).$$

This is a Gaussian mixture and its subpopulation parameters are

$$\Sigma_{tijk}^{-1} = \Sigma_{tik}^{-1} + B_{tj}^{\top}\Lambda_{tj}^{-1}B_{tj}$$

and

$$\mu_{tijk} = \Sigma_{tijk}\{\Sigma_{tik}^{-1}\mu_{tik} + B_{tj}^{\top}\Lambda_{tj}^{-1}(y_t - u_{tj})\}.$$

It is seen from (5.7) and (5.8), the order of both mixtures will increase exponentially with updating. Use GMR to reduce the order of the mixture before each update can be helpful to make the inference tractable. This example is used to illustrate the use of GMR in tracking under HMMs. We do not include a numerical example in this chapter, interested readers can refer to Yu et al. (2018, Section 5.3).

5.2 Existing GMR approaches

As we mentioned before, there are three general types of GMR approach: greedy algorithm-based, clustering-based, and optimization-based approaches. We provide additional details in three subsections respectively. The details on clustering-based approaches are needed later when we show that this type of approach is a special case of our proposed method.

5.2.1 Greedy Algorithms

The greedy algorithm-based approaches (Assa and Plataniotis, 2018; Runnalls, 2007; Salmond, 1990; Williams and Maybeck, 2006) start with the original mixture and usually either prune one component or merge two components at each step. The number of components in the mixture is hence reduced by one after each step. The procedure is repeated until the desired order is achieved.

The pruning is done by discarding the component that either has the smallest weight or the cost of its discarding is the lowest according to some metric. The weights of the remaining components are then renormalized to obtain a proper density function. Instead of pruning, one may merge two components that are most similar. One uses some metric c_{ij} between ith component and jth component

for the similarity. For example, Salmond (1990) considers

$$c_{ij} = \frac{w_i w_j}{w_i + w_j}(\mu_i - \mu_j)^\top \Sigma^{-1}(\mu_i - \mu_j)$$

where Σ is the covariance matrix of the original mixture $\Phi(x; G)$. In comparison, Runnalls (2007) considers

$$c_{ij} = w_{ij} \log \det(\Sigma_{ij}) - w_i \log \det(\Sigma_i) - w_j \log \det(\Sigma_j)$$

where

$$\begin{aligned} w_{ij} &= w_i + w_j \\ \mu_{ij} &= (w_i/w_{ij})\mu_i + (w_j/w_{ij})\mu_j \\ \Sigma_{ij} &= \sum_{k=i,j} (w_k/w_{ij})\{\Sigma_k + (\mu_k - \mu_{ij})(\mu_k - \mu_{ij})^\top\}. \end{aligned} \tag{5.9}$$

In this approach, two components have smallest c_{ij} value are merged into a single component via moment matching. Let us say the ith and the jth components of the current mixture are chosen to be merged. Then these two components along with their mixing weights can be viewed as an unnormalized mixture. This approach replaces these two components with a weighted Gaussian component that can also be viewed as an unnormalized mixture. The corresponding parameter values of the weighted Gaussian component are decided based on matching its first two moments with that of the unnormalized 2-component mixture. As a result, the corresponding parameters are given by w_{ij}, μ_{ij}, and Σ_{ij} in (5.9). By merging two components that minimize c_{ij} at each step, it lacks an ultimate optimality target. Such greedy algorithms are clearly sub-optimal since the low loss at the first step may lead to substantial losses in future steps.

5.2.2 An Optimization-based Algorithm

Williams and Maybeck (2006) formulates the GMR as an optimization problem. The reduced mixture minimizes some divergence to the original mixture. The mix-

ing distribution of the reduced mixture is simply

$$\widetilde{G} = \arg\min_{\widetilde{G}\in\mathbb{G}_M} D(\Phi(\cdot;G), \Phi(\cdot;\widetilde{G}))$$

for some divergence $D(\cdot,\cdot)$ and $\mathbb{G}_M$ is the space of all mixing distributions of order M. If successfully implemented, it obtains the best possible approximation according to $D(\cdot,\cdot)$ by definition.

Williams and Maybeck (2006) study this approach with $D(\cdot,\cdot)$ chosen to be the Integrated Squared Error (ISE): the squared L_2 distance between two Gaussian mixtures given by

$$\begin{aligned} D_{\text{ISE}}(\Phi(\cdot;G), \Phi(\cdot;\widetilde{G})) = &\sum_{n=1}^{N}\sum_{n'=1}^{N} w_n w_{n'} \phi(\mu_n; \mu_{n'}, \Sigma_n + \Sigma_{n'}) \\ &- 2\sum_{n=1}^{N}\sum_{m=1}^{M} w_n \widetilde{w}_m \phi(\mu_n; \widetilde{\mu}_m, \Sigma_n + \widetilde{\Sigma}_m) \\ &+ \sum_{m=1}^{M}\sum_{m'=1}^{M} \widetilde{w}_m \widetilde{w}_{m'} \phi(\widetilde{\mu}_m; \widetilde{\mu}_{m'}, \widetilde{\Sigma}_m + \widetilde{\Sigma}_{m'}). \end{aligned}$$

Although the above function has a closed-form, complicated numerical algorithms seem inevitable for minimization. Williams and Maybeck (2006) recommends the generic Quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm for optimization. Recall N is the order of the original mixture, M is the order of the reduced mixture, and d is the dimension. Using the BFGS algorithm, the cost for evaluating the gradient at each step is $\mathcal{O}(NMd^3)$ and the cost for approximating the Hessian is $\mathcal{O}(M^2d^4)$. As the cost is quartic in dimension d, directly minimizing ISE is very expensive when the dimension d is very large.

5.2.3 Clustering-based Algorithms

The clustering-based approach such as the one in Schieferdecker and Huber (2009) for GMR mimics the *k-means* algorithm. This approach permits a simple implementation and has a low computational cost. Recall we use N for the order of the original mixture and M for the order of the reduced mixture. The algorithm starts

with M initial cluster centres. It then iterates between the following two steps.

1. *Assignment step*: split N components of the original mixture into M groups based on their closeness to the proposed cluster centres;
2. *Update step*: obtain a new centre of the components in each group according to some criterion.

The iteration continues until the centres do not change meaningfully between iterations. The reduced Gaussian mixture has M components formed by these centres and the corresponding mixing weight as the sum of the weights of the original components belong to this cluster.

As in the case of clustering in the vector space, one can design different assignment schemes in this algorithm for GMR. If the algorithm assigns each component in the original mixture entirely to a single cluster, it is a hard clustering-based algorithm. If the algorithm split a component in the original mixture into several parts and assign them to different clusters, this leads to a soft clustering-based algorithm. The existing hard clustering-based and soft clustering-based approaches are described below.

Hard Clustering-based The hard clustering-based approaches partition the original mixture components into M groups according to some closeness measure. For example, Schieferdecker and Huber (2009) uses the KL divergence

$$\begin{aligned} D(\Phi_n, \widetilde{\Phi}_m) =& D_{\mathrm{KL}}(\Phi(\cdot; \mu_n, \Sigma_n) \| \Phi(\cdot; \widetilde{\mu}_m, \widetilde{\Sigma}_m)) \\ =& \frac{1}{2} \left(\log \frac{\det(\widetilde{\Sigma}_m)}{\det(\Sigma_n)} + \mathrm{tr}(\widetilde{\Sigma}_m^{-1} \Sigma_n) + (\widetilde{\mu}_m - \mu_n)^\top \widetilde{\Sigma}_m^{-1} (\widetilde{\mu}_m - \mu_n) - d \right). \end{aligned}$$

Assa and Plataniotis (2018) uses the squared Wasserstein distance between two Gaussians

$$\begin{aligned} D(\Phi_n, \widetilde{\Phi}_m) =& W_2^2(\Phi(\cdot; \mu_n, \Sigma_n), \Phi(\cdot; \widetilde{\mu}_m, \widetilde{\Sigma}_m)) \\ =& \|\mu_n - \widetilde{\mu}_m\|^2 + \mathrm{tr} \left(\Sigma_n + \widetilde{\Sigma}_m - 2(\Sigma_n^{1/2} \widetilde{\Sigma}_m \Sigma_n^{1/2})^{1/2} \right). \end{aligned}$$

Each component of the original mixture is assigned to its nearest cluster centre.

At the update step, the new cluster centre is formed by the components that are assigned to the same cluster. Suppose we assign the nth component of the original

mixture to cluster $C(n) \in [M]$. Schieferdecker and Huber (2009) creates the new cluster centres via moment matching:

$$\begin{aligned}
\widetilde{w}_m &= \sum_{\{n:C(n)=m\}} w_n \\
\widetilde{\mu}_m &= \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \mu_n \\
\widetilde{\Sigma}_m &= \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \{\Sigma_n + (\mu_n - \widetilde{\mu}_m)(\mu_n - \widetilde{\mu}_m)^\top\}.
\end{aligned} \tag{5.10}$$

Assa and Plataniotis (2018) forms the new cluster centres by local Wasserstein barycentre. Namely, the barycentre of the components assigned to this cluster.

Both of them iterate between these two steps until the change in the ISE between the original and reduced mixtures is below some threshold. A general description of the hard clustering-based algorithm is given in Algorithm 2.

Algorithm 2 A general description of hard clustering-based algorithms for GMR.

Input: $\Phi_1, \Phi_2, \ldots, \Phi_N, w_1, w_2, \ldots, w_N$
Initialize: $\widetilde{\Phi}_1, \widetilde{\Phi}_2, \ldots, \widetilde{\Phi}_M$
repeat
Assignment step:
Compute $d_{nm} = D_{\mathrm{KL}}(\Phi_n, \widetilde{\Phi}_m)$ in (2.5)
Assign component n to clusters $C(n) = \arg\min_j d_{nj}$
Update step: update cluster centre by moment matching
for $m \in [M]$ **do**

$$\begin{aligned}
\widetilde{w}_m &\leftarrow \sum_{\{n:C(n)=m\}} w_n \\
\widetilde{\mu}_m &\leftarrow \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \mu_n \\
\widetilde{\Sigma}_m &\leftarrow \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \{\Sigma_n + (\mu_n - \widetilde{\mu}_m)(\mu_n - \widetilde{\mu}_m)^\top\}.
\end{aligned}$$

end for
Let $\Phi(x; \widetilde{G}) = \sum_m \widetilde{w}_m \widetilde{\Phi}_m(x)$
until the change in the value of $D_{\mathrm{ISE}}(\Phi(\cdot; G), \Phi(\cdot; \widetilde{G}))$ is below some threshold

Soft Clustering-based Instead of assigning each component of the original mixture to one cluster, the soft clustering-based approach assigns z_{nm} fraction of component n to the mth clustering centre. Clearly, we require $z_{nm} \geq 0$ and $\sum_{m=1}^{M} z_{nm} = 1$ for $n = 1, 2, \ldots, N$.

Various forms of z_{nm} are considered in the literature. Let $\mathbb{E}_n\{h(X)\}$ be the expectation of $h(X)$ when $X \sim \Phi_n$ and define

$$\begin{aligned} E_{nm} =& \mathbb{E}_n\{\log \phi(X; \widetilde{\mu}_m, \widetilde{\Sigma}_m)\} \\ =& \int \phi(x; \mu_n, \Sigma_n) \log \phi(x; \mu_n, \Sigma_n)\, dx \\ & - D_{\mathrm{KL}}(\Phi(\cdot; \mu_n, \Sigma_n) \| \Phi(\cdot; \widetilde{\mu}_m, \widetilde{\Sigma}_m)). \end{aligned} \tag{5.11}$$

This quantity measures the similarity between components of the original mixture and the reduced mixture. Vasconcelos and Lippman (1999) recommends having

$$z_{nm} \propto \widetilde{w}_m \exp(w_n I E_{nm})$$

with some hyper-parameter $I > 0$. Yu et al. (2018) recommends having

$$z_{nm} \propto \widetilde{w}_m \exp(I E_{nm}). \tag{5.12}$$

The values of z_{nm} are computed in the assignment step of the clustering algorithm.

The update step in soft clustering-based algorithms uses the moment matching the same as the hard clustering-based algorithm but with adjusted fractions. We present the soft clustering-based algorithm of Yu et al. (2018) in Algorithm 3. We still use $C(n)$ to denote the cluster that the nth original component is assigned to in the hard clustering-based approach. When the hyper-parameter $I \to \infty$, we find $z_{nm} = 1$ if $C(n) = m$ and 0 otherwise. This shows that the soft clustering-based algorithm reduces to the hard clustering-based algorithm as $I \to \infty$.

The soft clustering-based algorithm in Yu et al. (2018) leads to an effective GMR procedure. However, Yu et al. (2018) derives the above procedure from a variational inference point of view. We find their derivation is wrong yet our finding invalidates their variational inference interpretation. Their GMR procedure itself is valid. We document this finding in the following remark.

Algorithm 3 Overview of soft clustering-based algorithms for GMR.

Input: $\Phi_1, \Phi_2, \ldots, \Phi_N, w_1, w_2, \ldots, w_N$, hyper-parameter $I > 0$
Initialize: $\widetilde{\Phi}_1, \widetilde{\Phi}_2, \ldots, \widetilde{\Phi}_M$, $\widetilde{w}_1, \widetilde{w}_2, \ldots, \widetilde{w}_m$
repeat
Assignment step:
Let

$$z_{nm} = \{\widetilde{w}_m \exp(IE_{nm})\} / \sum_{m'} \{\widetilde{w}_{m'} \exp(IE_{nm'})\}$$

where E_{nm} is defined in (5.11)
Update step:
for $m \in [M]$ **do**

$$\widetilde{w}_m \leftarrow \sum_n z_{nm} w_n$$

$$\widetilde{\mu}_m \leftarrow \widetilde{w}_m^{-1} \sum_n z_{nm} w_n \mu_n$$

$$\widetilde{\Sigma}_m \leftarrow \widetilde{w}_m^{-1} \sum_n z_{nm} w_n \left\{ \Sigma_n + (\mu_n - \widetilde{\mu}_m)(\mu_n - \widetilde{\mu}_m)^\top \right\}$$

end for
until the change in $\sum_n w_n \sum_m z_{nm} \left\{ \log \frac{\widetilde{w}_m}{z_{nm}} + IE_{nm} \right\}$ is below a threshold

Remark 5.1 (Technical errors in Yu et al. (2018))**.** *Let $X_1, X_2, \ldots, X_I$ be a set of* IID *pseudo-samples of size I from the original mixture $\phi(x; G)$. We denote by $X = (X_1, X_2, \ldots, X_I)^\top$. Yu et al. (2018) proposes to perform* GMR *by maximizing the expected likelihood of $\widetilde{G}$:*

$$\ell_I(\widetilde{G}) = \mathbb{E}\{\log \phi(X; \widetilde{G})\}.$$

The expectation is computed knowing $X \sim \prod_{i=1}^I \phi(x_i; G)$. Yu et al. (2018) apparently does not notice

$$\ell_I(\widetilde{G}) = I\, \mathbb{E}\{\log \phi(X_1; \widetilde{G})\}.$$

Instead, Yu et al. (2018) wrongfully claims that the expected log-likelihood function

$$\begin{aligned}\ell_I(\widetilde{G}) &= \mathbb{E}\{\log \phi(X; \widetilde{G})\} \\ &= \sum_{n=1}^{N} w_n \mathbb{E}_{Y \sim \prod_{i=1}^{I} \phi_n(y_i)} \left\{ \sum_{i=1}^{I} \log \phi(Y_i; \widetilde{G}) \right\}. \end{aligned} \tag{5.13}$$

Conceptually, this claim has regarded whole vector X having probability w_n to be an IID *sample from the subpopulation with distribution Φ_n, for $n \in [N]$. This is not true. One can also find* (5.13) *is false from integral expressions. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_I)$, and denote*

$$g(\boldsymbol{x}) = \phi(\boldsymbol{x}; \widetilde{G}) = \prod_{i=1}^{I} \phi(x_i; \widetilde{G}).$$

We have

$$\mathbb{E}\{\log \phi(X; \widetilde{G})\} = \int g(\boldsymbol{x}) \prod_{i=1}^{I} \left\{ \sum_{n=1}^{N} w_n \phi_n(x_i) \right\} d\boldsymbol{x}$$

while

$$\mathbb{E}_{Y \sim \prod_{i=1}^{I} \phi_n(y_i)} \left\{ \sum_{i=1}^{I} \log \phi(Y_i; \widetilde{G}) \right\} = \sum_{n=1}^{N} w_n \int g(\boldsymbol{x}) \prod_{i=1}^{I} \phi_n(x_i)\, d\boldsymbol{x}.$$

Two outcomes are clearly unequal because the summation and the product are not exchangeable. This misinterpretation carries in deriving a variational lower bound of their wrong objective function.

5.3 Proposed Reduction Approach

In this section, we present the novel GMR approach that minimizes the CTD from the reduced mixture to the original mixture. We present its formulation, the corresponding numerical algorithm, and its connection with some existing clustering-based algorithms.

Let $\Phi(x; G) = \sum_{n=1}^{N} w_n \Phi(x; \mu_n, \Sigma_n) := \sum_{n=1}^{N} w_n \Phi_n(x)$ be the original

mixture with order N. Our research problem is to search for a mixture

$$\Phi(x;\widetilde{G}) = \sum_{m=1}^{M} \widetilde{w}_m \Phi(x;\widetilde{\mu}_m,\widetilde{\Sigma}_m) := \sum_{m=1}^{M} \widetilde{w}_m \widetilde{\Phi}_m(x)$$

of order $M < N$ to approximate $\Phi(x;G)$.

Denote by $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^\top$ the mixing weights of the original mixture and $\widetilde{\boldsymbol{w}} = (\widetilde{w}_1, \widetilde{w}_2, \ldots, \widetilde{w}_m)^\top$ the mixing weights of the reduced mixture. Let $c(\Phi_n, \widetilde{\Phi}_m)$ be a non-negative bi-variate function on a space of Gaussian distributions. For example,

$$c(\Phi_n, \widetilde{\Phi}_m) = D_{\mathrm{KL}}(\Phi_n \| \widetilde{\Phi}_m)$$

could be a cost function. Let $\boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})$ be a transportation plan, and

$$\mathcal{H}(\boldsymbol{\pi}) = -\sum_{n,m} \pi_{nm}(\log \pi_{nm} - 1) \tag{5.14}$$

be the entropy of the transportation plan $\boldsymbol{\pi}$. We introduce an entropic regularized CTD

$$\begin{aligned} &\mathcal{T}_{c,\lambda}(\Phi(\cdot;G), \Phi(\cdot;\widetilde{G})) \\ &= \inf_{\boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})} \left\{ \sum_{n,m} \pi_{nm} c(\Phi_n, \widetilde{\Phi}_m) - \lambda \mathcal{H}(\boldsymbol{\pi}) \right\} \end{aligned} \tag{5.15}$$

for some regularization constant $\lambda \geq 0$.

Our proposed GMR reduces $\Phi(x;G)$ to $\Phi(x;\widetilde{G})$ with

$$\widetilde{G} = \operatorname*{arg\,inf}_{\widetilde{G} \in \mathbb{G}_M} \mathcal{T}_{c,\lambda}(\Phi(\cdot;G), \Phi(\cdot;\widetilde{G})) = \operatorname*{arg\,inf}_{\widetilde{G} \in \mathbb{G}_M} \mathcal{T}_{c,\lambda}(G, \widetilde{G}). \tag{5.16}$$

For simplicity, we also write $\mathcal{T}_{c,\lambda}(\Phi(\cdot;G), \Phi(\cdot;\widetilde{G}))$ as $\mathcal{T}_{c,\lambda}(G, \widetilde{G})$ as in (5.16).

We next introduce an MM algorithm in the spirit of Section 4.2 for the optimization in (5.16). We connect the generalized MM algorithm with the clustering-based algorithms introduced in Section 5.3.2. In Section 5.3.3, we discuss the choice of cost functions $c(\cdot,\cdot)$.

5.3.1 Numerical Algorithm

Let us define two functions of $\widetilde{G}$, with G hidden in the background:

$$\mathcal{J}_{c,\lambda}(\widetilde{G}) = \inf_{\boldsymbol{\pi}\in\Pi(\boldsymbol{w},\cdot)} \left\{ \sum_{n,m} \pi_{nm} c(\Phi_n, \widetilde{\Phi}_m) - \lambda \mathcal{H}(\boldsymbol{\pi}) \right\}, \tag{5.17}$$

$$\boldsymbol{\pi}^{\lambda}(\widetilde{G}) = \arg\inf_{\boldsymbol{\pi}\in\Pi(\boldsymbol{w},\cdot)} \left\{ \sum_{nm} \pi_{nm} c(\Phi_n, \widetilde{\Phi}_m) - \lambda \mathcal{H}(\boldsymbol{\pi}) \right\} \tag{5.18}$$

where $\mathcal{H}(\boldsymbol{\pi})$ is defined in (5.14). The optimization problem in (5.17) and (5.18) involves only one linear constraint in terms of $\boldsymbol{w}$. For a given cost function $c(\cdot,\cdot)$ and a level $\lambda > 0$, the optimal transportation plan $\boldsymbol{\pi}^{\lambda}(\widetilde{G})$ for this problem has an analytical solution

$$\pi^{\lambda}_{nm}(\widetilde{G}) = \frac{w_n \exp(-c(\Phi_n, \widetilde{\Phi}_m)/\lambda)}{\sum_{m'} \exp(-c(\Phi_n, \widetilde{\Phi}_{m'})/\lambda)}.$$

The derivation of the analytical form is given in Appendix C.1. The optimal transportation plan under the special case $\lambda = 0$ discussed in Section 4.2 is the limit of the optimal transportation plan $\pi^{\lambda}_{nm}(\widetilde{G})$ when $\lambda \to 0$. Introduce an index set

$$A_n = \{m' : c(\Phi_n, \widetilde{\Phi}_{m'}) = \min_m c(\Phi_n, \widetilde{\Phi}_m)\}$$

and let $\text{card}(A_n)$ be the cardinality of set A_n. It is seen that

$$\lim_{\lambda\downarrow 0} \pi^{\lambda}_{nm}(\widetilde{G}) = \begin{cases} w_n/\text{card}(A_n) & \text{if } m \in A_n \\ 0 & \text{otherwise} \end{cases}.$$

For the ease of notation, for $\lambda \geq 0$, we denote by

$$\pi^{\lambda}_{nm}(\widetilde{G}) = \begin{cases} w_n \dfrac{\exp(-c(\Phi_n, \widetilde{\Phi}_m)/\lambda)}{\sum_{m'} \exp(-c(\Phi_n, \widetilde{\Phi}_{m'})/\lambda)} & \lambda > 0, \\ w_n \dfrac{\mathbb{1}\{m \in \arg\min_{m'} c(\Phi_n, \widetilde{\Phi}_{m'})\}}{|\arg\min_{m'} c(\Phi_n, \widetilde{\Phi}_{m'})|} & \lambda = 0. \end{cases} \tag{5.19}$$

The conclusion in the following theorem simplifies the optimization problem in our propose GMR method.

Theorem 5.1. *Let G be the mixing distribution of the original mixture, $\mathcal{T}_{c,\lambda}(\cdot)$, $\mathcal{J}_{c,\lambda}(\cdot)$, and $\boldsymbol{\pi}^{\lambda}(\cdot)$ be defined in* (5.15)*,* (5.17)*, and* (5.18) *respectively. We have*

$$\inf\{\mathcal{T}_{c,\lambda}(G,\widetilde{G}) : \widetilde{G} \in \mathbb{G}_M\} = \inf\{\mathcal{J}_{c,\lambda}(\widetilde{G}) : \widetilde{G} \in \mathbb{G}_M\}.$$

The reduced mixture is hence given by

$$\widetilde{G} = \arg\inf\{\mathcal{J}_{c,\lambda}(\widetilde{G}) : \widetilde{G} \in \mathbb{G}_M\} \tag{5.20}$$

with mixing weights $\widetilde{\boldsymbol{w}}$ with its mth entry

$$\widetilde{w}_m = \sum_n \pi^{\lambda}_{nm}(\widetilde{G}). \tag{5.21}$$

Given this theorem, the MM algorithm in Section 4.2 and its convergence conclusion can be generalized straight-forwardly with the new transportation plan $\boldsymbol{\pi}^{\lambda}(\widetilde{G})$ in (5.19). With a minor level of redundancy, we describe the MM algorithm again to refresh our memory. The pseudo-code for the MM algorithm is given in Algorithm 4.

Algorithm 4 MM algorithm for GMR with CTD

Input: $\Phi_1, \Phi_2, \ldots, \Phi_N, w_1, w_2, \ldots, w_N$
Initialization: $\widetilde{\Phi}_m$, $m \in [M]$
repeat
  for $m \in [M]$ **do**
    Assignment step: compute π^{λ}_{nm} in (5.19)
    Update step:
    Let $\widetilde{w}_m = \sum_n \pi^{\lambda}_{nm}$
    Let $\widetilde{\Phi}_m = \arg\min_{\Phi} \sum_{n=1}^{N} \pi^{\lambda}_{nm} c(\Phi_n, \Phi)$
  end for
until $\sum_{n,m} \pi^{\lambda}_{nm} c(\Phi_n, \widetilde{\Phi}_m) - \lambda \mathcal{H}(\boldsymbol{\pi}^{\lambda})$ converges

The algorithm starts with some initial $\widetilde{G}^{(0)}$. Let $\widetilde{G}^{(t)}$ be the mixing distribution of the reduced mixture after t MM iterations. Define a majorization function of $\mathcal{J}_{c,\lambda}$

at $\widetilde{G}^{(t)}$ to be

$$\mathcal{K}_{c,\lambda}(\widetilde{G}|\widetilde{G}^{(t)}) = \left\{ \sum_{n,m} \pi_{nm}^{\lambda}(\widetilde{G}^{(t)}) c(\Phi_n, \widetilde{\Phi}_m) \right\} - \lambda \mathcal{H}(\boldsymbol{\pi}^{\lambda}(\widetilde{G}^{(t)})) \tag{5.22}$$

where $\pi_{nm}^{\lambda}(\widetilde{G}^{(t)})$ is the transportation plan defined by (5.19) and

$$\widetilde{w}_m^{(t+1)} = \sum_n \pi_{nm}^{\lambda}(\widetilde{G}^{(t)}),\ t = 1, 2, \ldots.$$

It can be seen that $\mathcal{K}_{c,\lambda}$ is a majorization function for $\mathcal{J}_{c,\lambda}$ meaning

$$\mathcal{K}_{c,\lambda}(\widetilde{G}|\widetilde{G}^{(t)}) \geq \mathcal{J}_{c,\lambda}(\widetilde{G}),\ t = 1, 2, \ldots.$$

The mean and covariance of $\widetilde{\Phi}_m$ are separated from mean and covariance of $\widetilde{\Phi}_{m'}$ when $m \neq m'$ in the majorization function (5.22). This allows us to update the subpopulation parameters of $\widetilde{\Phi}_m$ one subpopulation at a time and possibly in parallel, as the solutions to

$$\widetilde{\Phi}_m^{(t+1)} = \arg\inf_{\Phi} \left\{ \sum_n \pi_{nm}^{\lambda}(\widetilde{G}^{(t)}) c(\Phi_n, \Phi) \right\}. \tag{5.23}$$

The MM algorithm iterates between the majorization step (5.22) and the minimization step (5.23) until some user-selected convergence criterion is met.

Computational Complexity Analysis

The proposed MM algorithm is iterative. We assess its computational cost **at each iteration**. The cost function needs to be evaluated for $\mathcal{O}(NM)$ times. For most of the cost functions to be considered, the cost for their evaluation once is $\mathcal{O}(d^3)$ where d is the dimension of μ. Therefore, the total cost for evaluating the cost function is $\mathcal{O}(NMd^3)$. The cost for computing the transportation plan is $\mathcal{O}(NM)$. The computationally most expensive step is the updating step to obtain M local barycentres $\{\widetilde{\Phi}_m : m \in [M]\}$. This computation cost depends on the cost function we choose in the algorithm. When the cost function is Kullback-Leibler (KL) divergence, then the barycentre has a closed form and the cost for computing one

barycentre is $\mathcal{O}(d^2)$. Therefore, in this case, the total computation cost for finding the barycentres is $\mathcal{O}(Md^2)$. Hence, when the cost function is chosen to be the KL divergence, the total computation cost at each iteration is $\mathcal{O}(NMd^3)$. Compared to computation cost for the ISE reduction approach which is quartic in d, the proposed algorithm has a computation cost cubic in d. Therefore, our algorithm is computationally less expensive than the ISE reduction approach as described in Section 5.2.2 at each iteration.

5.3.2 Existing Algorithms as Special Cases

Our proposed GMR approach includes many clustering-based approaches in the literature as special cases. It also connects with many existing optimization-based approaches. We establish the connection in this section.

Clustering-based Algorithms

Schieferdecker and Huber (2009) argues that the clustering-based algorithms are computationally much cheaper than the optimization-based algorithm discussed in Williams and Maybeck (2006) and some greedy algorithms. Despite the computation efficiency, they do not establish the convergence of these algorithms nor identify their optimality targets when they converge.

We show that existing clustering-based algorithms are special cases of our proposed MM algorithm with specific cost functions in CTD as defined in (5.15). Because of this, our results provide important support to these approaches that were missing in the literature in the following aspects.

1. **Objective**: Because most existing clustering-based algorithms are special cases of our proposed MM algorithm, all of them are unknowingly minimizing an entropic regularized CTD.

2. **Convergence**: For the same reason, most existing clustering-based algorithms convergence when their corresponding entropic regularized CTD satisfy conditions in Theorem 4.2.

3. **Consistency**: Our proposed MM algorithm uses the same cost function $c(\cdot, \cdot)$ in the assignment and update steps. In the assignment step, we use the cost

function to measure the similarity between components in the original mixture and components in the proposed mixture $F(\cdot; G^{(t)})$. In the update step, we search for the barycentre of components in the original mixture assigned to the same cluster with respect to this same cost function. Our theory shows the MM algorithm produces a sequence with non-increasing entropic regularized CTD and therefore converges in this case. If different cost functions are used in these two steps, this guarantee may not be true. This happens, for example, when one assigns the components to clusters based on some divergence such as Wasserstein distance but nonetheless updates the cluster centres by moment matching. Since moment matching leads to barycentre under KL divergence, the convergence of the algorithm is not implied by our theory.

We now show the hard clustering-based algorithm of Schieferdecker and Huber (2009) is a special case of our algorithm with $\lambda = 0$ and the cost function

$$c(\Phi_n, \widetilde{\Phi}_m) = D_{\mathrm{KL}}(\Phi_n \| \widetilde{\Phi}_m)$$

According to our assignment step (5.19) in Algorithm 4, when $\lambda = 0$, the transportation plan becomes

$$\pi_{nm} = \begin{cases} w_n/|A_n| & \text{if } m \in A_n \\ 0 & \text{otherwise.} \end{cases}$$

where $A_n = \{m' : D_{\mathrm{KL}}(\Phi_n \| \Phi_{m'}) = \min_m D_{\mathrm{KL}}(\Phi_n \| \Phi_m)\}$. Then the mixing weights becomes

$$\widetilde{w}_m = \sum_{n=1}^{N} \pi_{nm} = \sum_{\{n:C(n)=m\}} w_n$$

and the mth subpopulation is updated via the KL barycentre given in Example 2.4

$$\widetilde{\Phi}_m = \arg\inf_{\Phi} \sum_{n=1}^{N} \pi_{nm} D_{\mathrm{KL}}(\Phi_n \| \Phi)$$

By substituting λ_n with π_{nm} above, the updated subpopulation parameters based

on our approach becomes

$$\widetilde{\mu}_m = \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \mu_n$$

and

$$\widetilde{\Sigma}_m = \widetilde{w}_m^{-1} \sum_{\{n:C(n)=m\}} w_n \{\Sigma_n + (\mu_n - \widetilde{\mu}_m)(\mu_n - \widetilde{\mu}_m)^\top\},$$

which are the same as the moment matching given in the hard clustering algorithm in Algorithm 2.

Initially, we also think the soft clustering based Algorithm in Yu et al. (2018) is a speical case of ours by letting

$$c(\Phi_n, \widetilde{\Phi}_m) = -\log \widetilde{w}_m - ID_{\mathrm{KL}}(\Phi_n \| \widetilde{\Phi}_m).$$

As we show later that this cost function leads to an MM algorithm that has exactly the same update as the soft clustering based algorithm in Yu et al. (2018). However, since this cost function depends on the mixing weights of the reduced mixture, the theoretical convergence of our proposed MM algorithm is not guaranteed.

With this cost function, in the assignment step (5.19) of Algorithm 1, the transportation plan becomes

$$\pi_{nm} = \frac{w_n \exp(-c(\Phi_n, \widetilde{\Phi}_m))}{\sum_{m'} \exp(-c(\Phi_n, \widetilde{\Phi}_{m'}))} = \frac{w_n \widetilde{w}_m \exp(IE_{nm})}{\sum_{m'} \widetilde{w}_{m'} \exp(IE_{nm'})} = w_n z_{nm}$$

with z_{nm} the same as in (5.12). The mixing weights of the reduced mixture becomes

$$\widetilde{w}_m = \sum_{n=1}^{N} \pi_{nm} = \sum_{n=1}^{N} w_n z_{nm} \tag{5.24}$$

and the mth subpopulation is updated via

$$\begin{aligned}
\widetilde{\Phi}_m &= \arg\inf_{\widetilde{\Phi}} \sum_{n=1}^{N} \pi_{nm} c(w_n \Phi_n, \widetilde{w}_m \widetilde{\Phi}) \\
&= \arg\inf_{\widetilde{\Phi}} \sum_{n=1}^{N} \pi_{nm} \{ -\log \widetilde{w}_m - I \int \phi(x; \mu_n, \Sigma_n) \log \phi(x; \widetilde{\mu}, \widetilde{\Sigma}) \, dx \} \\
&= \arg\inf_{\Phi} \sum_{n=1}^{N} \pi_{nm} \int \log \left\{ \frac{\phi(x; \mu_n, \Sigma_n)}{\phi(x; \widetilde{\mu}, \widetilde{\Sigma})} \right\} \phi(x; \mu_n, \Sigma_n) \, dx \\
&= \arg\inf_{\widetilde{\Phi}} \sum_{n=1}^{N} \pi_{nm} D_{\mathrm{KL}}(\Phi(\cdot; \mu_n, \Sigma_n) \| \Phi(\cdot; \widetilde{\mu}, \widetilde{\Sigma})).
\end{aligned}$$

Therefore, the components of the reduced mixture are the barycentres of the original Gaussian components assigned to the cluster under the KL divergence. By substituting λ_n with π_{nm} in Example 2.4, along with (5.24), the updated subpopulation parameters based on our approach becomes

$$\widetilde{\mu}_m = \widetilde{w}_m^{-1} \sum_{n=1}^{N} w_n z_{nm} \mu_n \tag{5.25}$$

and

$$\widetilde{\Sigma}_m = \widetilde{w}_m^{-1} \sum_{n=1}^{N} w_n z_{nm} \{ \Sigma_n + (\mu_n - \widetilde{\mu}_m)(\mu_n - \widetilde{\mu}_m)^\top \}. \tag{5.26}$$

It can be seen that (5.24) – (5.26) lead to the cluster centres given in the soft clustering algorithm in Algorithm 3.

Optimization-based Algorithms

There are two connected but slightly different optimization-based approaches for GMR. One is the proposed approach that minimizes the CTD $\mathcal{T}_{c,\lambda}$. The other is to directly minimize some cost function $\mathrm{cost}(\cdot, \cdot)$ between two mixtures for GMR. We use $\mathrm{cost}(\cdot, \cdot)$ here to highlight that it is a divergence between two general distributions. In comparison, the cost function $c(\cdot, \cdot)$ in CTD is a divergence between two subpopulation distributions.

Conceptually, one can choose the cost function $\text{cost}(\cdot,\cdot)$ as the cost function $c(\cdot,\cdot)$ in CTD. Namely, one can make

$$\text{cost}(\cdot,\cdot) = c(\cdot,\cdot).$$

In this case, we are interested in finding out the connection between the two targets $\text{cost}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})) = c(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G}))$ and $\mathcal{T}_{c,\lambda}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G}))$ in (2.16). We show in this section that when $c(\cdot,\cdot)$ satisfies "convexity", then the CTD between two mixtures is an upper bound for the cost between two mixtures.

Consider the special case where we wish to reduce the Gaussian mixture to a single Gaussian and the cost function in CTD is chosen to be ISE or the KL divergence between two Gaussians. If so, we have

$$\arg\min_{\widetilde{\Phi}} \sum_{n=1}^{N} w_n c(\Phi_n, \widetilde{\Phi}) = \arg\min_{\widetilde{\Phi}} c\left(\sum_{n=1}^{N} w_n \Phi_n, \widetilde{\Phi}\right). \tag{5.27}$$

The Left Hand Side (LHS) of (5.27) is the CTD between the original mixture and the reduced mixture and the Right Hand Side (RHS) of (5.27) is the divergence between two mixtures. The equality shows that when we reduce a mixture to a single Gaussian with some cost functions, these two approaches are equivalent. We give a proof in Appendix C.3. More generally, when the cost function $c(\cdot,\cdot)$ has a "convexity" property to be defined later, we have

$$\mathcal{T}_{c,0}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})) \geq c(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})).$$

That is to say our proposed GMR approach when $\lambda = 0$ in fact minimizes an upper bound of the direct cost function.

Theorem 5.2. *Let $c(\cdot,\cdot)$ be a non-negative bi-variate function on space of Gaussian mixture distributions with "convexity" property: for any $\alpha \in (0,1)$, and Gaussian distributions F_1, F_2, Φ_1, and Φ_2, we have*

$$c(\alpha F_1 + (1-\alpha)F_2, \alpha\Phi_1 + (1-\alpha)\Phi_2) \leq \alpha c(F_1,\Phi_1) + (1-\alpha)c(F_2,\Phi_2).$$

Then for all $\widetilde{G}$, we have

$$c(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})) \leq \mathcal{T}_{c,0}(\Phi(\cdot;G),\Phi(\cdot;\widetilde{G})).$$

The proof of this theorem is given in Appendix C.3.

Remark 5.2. *The conclusion in Theorem 5.2 under some special cost functions has been shown in the literature from different angles. Delon and Desolneux (2020, Section 4.2) obtains this conclusion for cost function*

$$c(\Phi_n,\widetilde{\Phi}_m) = W_2^2(\Phi_n,\widetilde{\Phi}_m)$$

where W_2 the 2-Wasserstein distance between two Gaussians with the ground distance being Euclidean distance in $\mathbb{R}^d$. Nguyen (2013, Lemma 1) obtains this conclusion when the cost function is the general class f-divergence between two densities.

Many divergence functions have the convexity property. Other than the whole family of f-divergence (Nguyen, 2013, Lemma 1) and the squared 2-Wasserstein distance (Villani, 2003, Chapter 7), we show that ISE is also convex as follows. Recall $\mathcal{F}$ is the distribution family of subpopulations in our context of finite mixture models.

Example 5.1 (Integrated Squared Error is Convex)**.** *Let F_1, F_2, Φ_1, Φ_2 be some subpopulation distributions with density functions f_1, f_2, ϕ_1, and $\phi_2 \in \mathcal{F}$ and $\alpha \in [0,1]$. Note that Φ_1 and Φ_2 are not necessarily Gaussian distributions. Let D_{ISE} be defined in* (2.6)*, we have*

$$D_{ISE}(\alpha F_1+(1-\alpha)F_2, \alpha\Phi_1+(1-\alpha)\Phi_2) \leq \alpha D_{ISE}(F_1,\Phi_1)+(1-\alpha)D_{ISE}(F_2,\Phi_2).$$

The proof is straightforward and is the same as showing Euclidean norm on a

vector space is convex. By the definition of D_{ISE}, we have

$$\begin{aligned}
&D_{\text{ISE}}(\alpha F_1 + (1-\alpha)F_2, \alpha\Phi_1 + (1-\alpha)\Phi_2) \\
=&\int \{\alpha\{F_1(x) - \Phi_1(x)\} + (1-\alpha)\{f_2(x) - \Phi_2(x)\}\}^2 dx \\
=&\alpha^2 D_{\text{ISE}}(F_1, \Phi_1) + (1-\alpha)^2 D_{\text{ISE}}(F_2, \Phi_2) + 2\alpha(1-\alpha)\langle f_1 - \phi_1, f_2 - \phi_2\rangle
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\alpha D_{\text{ISE}}(F_1, \Phi_1) + (1-\alpha) D_{\text{ISE}}(F_2, \Phi_2) \\
&- D_{\text{ISE}}(\alpha F_1 + (1-\alpha)F_2, \alpha\Phi_1 + (1-\alpha)\Phi_2) \\
=&\alpha(1-\alpha)\{D_{\text{ISE}}(F_1, \Phi_1) + D_{\text{ISE}}(F_2, \Phi_2) - 2\langle f_1 - \phi_1, f_2 - \phi_2\rangle\} \\
=&\alpha(1-\alpha) D_{\text{ISE}}(F_1 - \Phi_1, F_2 - \Phi_2) \geq 0.
\end{aligned}$$

The last inequality holds because the ISE is non-negative and $\alpha \in [0, 1]$. This completes the proof.

5.3.3 The Choice of Cost Functions in the Proposed GMR

The proposed MM algorithm and its implied GMR cover most known clustering-based GMR approaches as special cases. At the same time, our proposed approach goes far beyond. The proposed Algorithm 4 determines what fractions of a subpopulation in the original mixture will be assigned to a cluster in the reduced mixture based on a cost function. It also uses the same cost function to update the cluster centres. Hence, the choice of the cost function is very important in terms of computational cost and the property of the reduced mixture.

In this section, we investigate two choices of the cost function: the Cauchy-Schwarz (CS) divergence and the Integrate Squared Error (ISE). We find that both divergences have closed-form expressions when applied to Gaussian distributions. Their corresponding Gaussian barycentres do not have closed expressions. Therefore, we also discuss the corresponding numerical solutions for barycentres.

Cauchy-Schwarz Divergence

By straightforward calculation, we find the CS divergence between two Gaussians is given by

$$\begin{aligned} & D_{\mathrm{CS}}(\Phi(\cdot;\mu_1,\Sigma_1),\Phi(\cdot;\mu_2,\Sigma_2)) \\ =& -\log \frac{\int \phi(x;\mu_1,\Sigma_1)\phi(x;\mu_2,\Sigma_2)dx}{\sqrt{\int \phi^2(x;\mu_1,\Sigma_1)dx \int \phi^2(x;\mu_2,\Sigma_2)dx}} \\ =& -\log \phi(\mu_1;\mu_2,\Sigma_1+\Sigma_2) - \frac{1}{4}\left\{\log\det(\Sigma_1)+\log\det(\Sigma_2)\right\} - \frac{d}{2}\log(2\pi). \end{aligned}$$

Let $\Phi_n = \Phi(\cdot;\mu_n,\Sigma_n)$ be the components of the original mixture. When the CS divergence is the cost function in the proposed MM algorithm, the update step requires to find the barycentre that is defined as a Gaussian distribution Φ that minimizes

$$\sum_{n=1}^{N} \lambda_n D_{\mathrm{CS}}(\Phi_n,\Phi)$$

for some $\boldsymbol{\lambda} = (\lambda_1,\lambda_2,\ldots,\lambda_N)^\top \in \Delta_{N-1}$. Denote the solution by $\widetilde{\Phi}$. It is easy to find that its mean vector $\widetilde{\mu}$ and covariance matrix $\widetilde{\Sigma}$ satisfy

$$\widetilde{\mu} = \left\{\sum_n \lambda_n(\Sigma_n+\widetilde{\Sigma})^{-1}\right\}^{-1} \sum_n \lambda_n(\Sigma_n+\widetilde{\Sigma})^{-1}\mu_n. \tag{5.28}$$

Hence, the task of finding $\widetilde{\Phi}$ is reduced to find its covariance $\widetilde{\Sigma}$. With some straightforward algebra, we find $\widetilde{\Sigma}$ solves the matrix equation

$$\widetilde{\Sigma}^{-1} = 2\sum_n \lambda_n(\Sigma_n+\widetilde{\Sigma})^{-1}\{\mathbf{I}_d - (\mu_n-\widetilde{\mu})(\mu_n-\widetilde{\mu})^\top(\Sigma_n+\widetilde{\Sigma})^{-1}\} \tag{5.29}$$

where $\mathbf{I}_d$ is the identity matrix of size $d\times d$ and d is the dimension of the Gaussian distribution.

The technical details behind (5.28) and (5.29) are as follows. Denote by

$$L(\mu,\Sigma) = \sum_n \lambda_n D_{\mathrm{CS}}(\Phi_n\|\Phi)$$

which has the following expression:

$$L(\mu, \Sigma) = \frac{1}{2} \sum_n \lambda_n (\mu_n - \mu)^\top (\Sigma_n + \Sigma)^{-1} (\mu_n - \mu) + \frac{1}{2} \sum_n \lambda_n \log \left\{ \frac{\det(\Sigma_n + \Sigma)}{\det^{1/2}(\Sigma)} \right\} + C$$

for some constant C. Since $L(\mu, \Sigma)$ is a smooth function, the solution to the optimization problem is a stationary point. It is seen that

$$\frac{\partial L}{\partial \mu} = -\sum_n \lambda_n (\Sigma_n + \Sigma)^{-1} (\mu_n - \mu)$$

and

$$2\frac{\partial L}{\partial \Sigma} = -\frac{1}{2}\Sigma^{-1} + \sum_n \lambda_n (\Sigma_n + \Sigma)^{-1} \left\{ \mathbf{I}_d - (\mu_n - \mu)(\mu_n - \mu)^\top (\Sigma_n + \Sigma)^{-1} \right\}.$$

Setting both partial derivatives to $\mathbf{0}$, we get

$$\widetilde{\mu} = \left\{ \sum_n \lambda_n (\Sigma_n + \widetilde{\Sigma})^{-1} \right\}^{-1} \sum_n \lambda_n (\Sigma_n + \widetilde{\Sigma})^{-1} \mu_n$$

and

$$\widetilde{\Sigma}^{-1} = 2 \sum_n \lambda_n (\Sigma_n + \widetilde{\Sigma})^{-1} \{ \mathbf{I}_d - (\mu_n - \widetilde{\mu})(\mu_n - \widetilde{\mu})^\top (\Sigma_n + \widetilde{\Sigma})^{-1} \}$$

as claimed earlier.

To numerically obtain the mean and covariance of the CS barycentre, we iteratively update $\widetilde{\mu}$ and $\widetilde{\Sigma}$ via (5.28) and (5.29) from some initial value. The iterations stop when the change in the value of $\sum_{n=1}^N \lambda_n D_{\text{CS}}(\Phi_n, \Phi)$ is below a threshold. The KL Gaussian barycentre has a closed form solution and we use this solution as initial value of the covariance matrix $\widetilde{\Sigma}$ in our numerical implementation.

We show in Section 5.3.2 that the convexity property of a cost function according to Definition 5.2 leads to some good properties. Unfortunately, the CS divergence is not convex as shown below.

Example 5.2 (Cauchy-Schwarz Divergence is Non-Convex). *Let* $\alpha = 0.5$ *and* $\mu > 0$ *and* $\sigma > 0$ *be two constants. Let* $F_1(x) = \Phi(x; -1, \sigma^2)$, $F_2(x) = \Phi(x; -\mu, 1)$, $\Phi_1(x) = \Phi(x; 1, 1)$, *and* $\Phi_2(x) = \Phi(x; \mu, \sigma^2)$ *be four Gaussian distributions. If* CS *divergence has convexity property, we should have*

$$D_{CS}(\alpha F_1 + (1-\alpha)F_2, \alpha\Phi_1 + (1-\alpha)\Phi_2) > \alpha D_{CS}(F_1, \Phi_1) + (1-\alpha)D_{CS}(F_2, \Phi_2).$$

Using the closed-form of the CS *divergence between two Gaussian mixtures, we are able to obtains the closed-form for both* LHS *and* RHS *of this inequality. The difference of* LHS-RHS *as a function of* μ *and* σ *is shown in Figure 5.2. It shows*

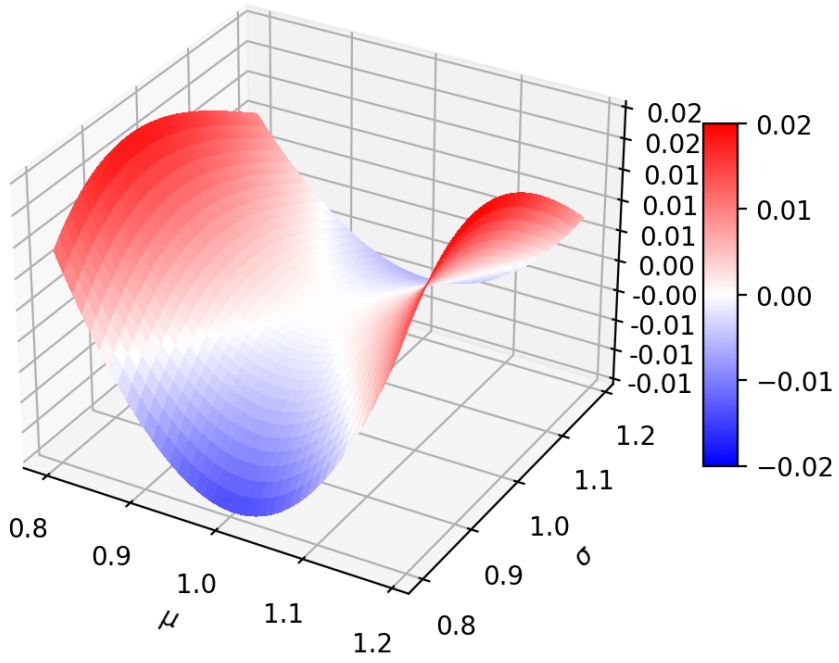


Figure 5.2: The function of the difference CS divergence between two Gaussians is non-convex.

that this function has a saddle surface and is not always positive. Therefore, the CS *divergence is not convex.*

Integrated Squared Error

The ISE between two Gaussians is another of our choice of the cost function in subsequent experiments. It has a closed form expression:

$$D_{\text{ISE}}(\Phi(\cdot;\mu_1,\Sigma_1),\Phi(\cdot;\mu_2,\Sigma_2)) = \det{}^{-1/2}(2\pi\Sigma_1) + \det{}^{-1/2}(2\pi\Sigma_2) - 2\phi(\mu_1;\mu_2,\Sigma_1+\Sigma_2).$$

The objective function $\sum_{n=1}^{N}\lambda_n D_{\text{ISE}}(\Phi_n,\Phi)$ for finding out the local barycentres when the ISE is the cost function does not have convexity property. Hence, the local barycentres may be some local minima when numerical algorithms are used for computation. In our experiment, we search for barycentre using the numerical algorithms such as BFGS or the Nelder-Mead algorithm (Nelder and Mead, 1965). We use the KL barycentre and the Wasserstein barycentre as initial values. To ensure the positive definiteness of the covariance matrix, we use the numerical trick to optimize over the Cholesky decomposition of the covariance matrix instead.

5.4 Experiments

In this section, we use numerical experiments to compare the performance of various reduction approaches. Recall $\Phi(\cdot;G)$ is the original mixture with order N and we want to approximate it by an order M mixture $\Phi(\cdot;\widetilde{G})$. We include the following reduction approaches:

1) directly minimize $D_{\text{ISE}}(\Phi(\cdot;G),\Phi(x;\widetilde{G}))$ and we refer to this approach as MISE;

2) the proposed CTD-based approach with cost functions: ISE, CS, KL, and squared 2-Wasserstein. We denote these methods as CTD-ISE, CTD-CS, CTD-KL and CTD-W2.

We set the level of entropy regularization at $\lambda = 0$ in all experiments. Unless otherwise specified, all the reduction algorithms are initialized by 10 values obtained by fitting a Gaussian mixture of M components by the penalized Maximum Likelihood Estimate (PMLE) using random samples of size 100 generated from the

original mixture. All experiments are implemented in `Python` 3.7.4 on Cedar cluster at Compute Canada with Intel E5 CPU with 2.1Ghz.

5.4.1 Simulated Experiment

Our first experiment considers reducing a bi-variate Gaussian mixture of order $N = 25$ to a mixture of order $M = 5, 10$, and 15. Rather than select an order $N = 25$ bi-variate Gaussian mixture arbitrarily, we create $R = 100$ of them with a certain structure and a large dose of uncertainty. We describe the procedure for generating the parameter values of the original mixture in each repetition. We let all the mixing weights $w_j = 0.04$, $j \in [25]$. Our first task is to generated $N = 25$ subpopulation mean vectors. We make these subpopulations roughly clustered around randomly selected 5 centres. For this reason, we generate a multinomial random vector $n_1, n_2, \ldots, n_5$ with event probabilities $(0.2, \ldots, 0.2)$ so that $\sum_{i=1}^{5} n_i = 25$. Each centre $\boldsymbol{\mu}_i^0, i = 1, 2, \ldots, 5$ is selected uniformly from $[-10, 10] \times [-10, 10]$. We then generate $\boldsymbol{\mu}_{ij}^0, j = 1, \ldots, n_i$ uniformly in the disk with radius 2.5. The subpopulation mean vectors are then chosen to be

$$\boldsymbol{\mu}_i^0 + \boldsymbol{\mu}_{ij}^0, \;\; j = 1, \ldots, n_i; \; i = 1, 2, \ldots, 5.$$

Clearly, these subpopulation means should be clustered around $\boldsymbol{\mu}_i^0, i = 1, 2, \ldots, 5$.

The next task is to generate $N = 25$ subpopulation covariance matrices. For $n \in [N]$, we generate σ_{11n}, σ_{22n} from Gamma distribution with shape parameter 8 and scale parameter 4, and generate a θ_n uniform in $[0.2, 0.8]$. We then let

$$\Sigma_n = \begin{pmatrix} \sigma_{11n} & \sqrt{\sigma_{11n}\sigma_{22n}}\cos(\theta_n \pi) \\ \sqrt{\sigma_{11n}\sigma_{22n}}\cos(\theta_n \pi) & \sigma_{22n} \end{pmatrix}$$

be the covariance matrix of the nth component of the original mixture. Note the π in the above expression is the mathematical constant whose value approximately equals to 3.14159, not a transportation plan. Figure 5.3 (a) shows an example of one generated subpopulation means and their cluster centres. Figure 5.3 (b) shows the heat-map of the corresponding original mixture of order $N = 25$.

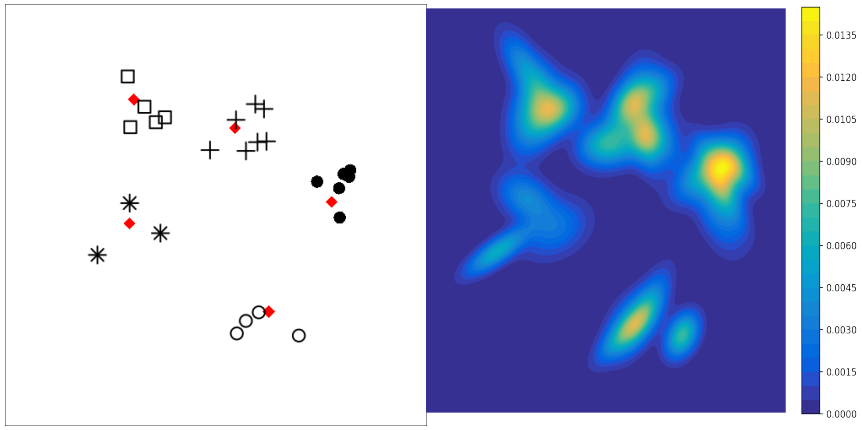
The above design has taken several factors into consideration. If the mean

vectors of the mixture are completely random, we suspect none of GMR approaches can find a substantially reduced mixture that approximates the original mixture well. This will render the comparison meaningless. At the same time, we must leave enough uncertainty in the original mixture. If all GMR approaches find a very precise approximation, the comparison will also lose its value. Our design also contains a few tuning constants. We can choose different values to use the one so that the outcome of the experiment is informative.

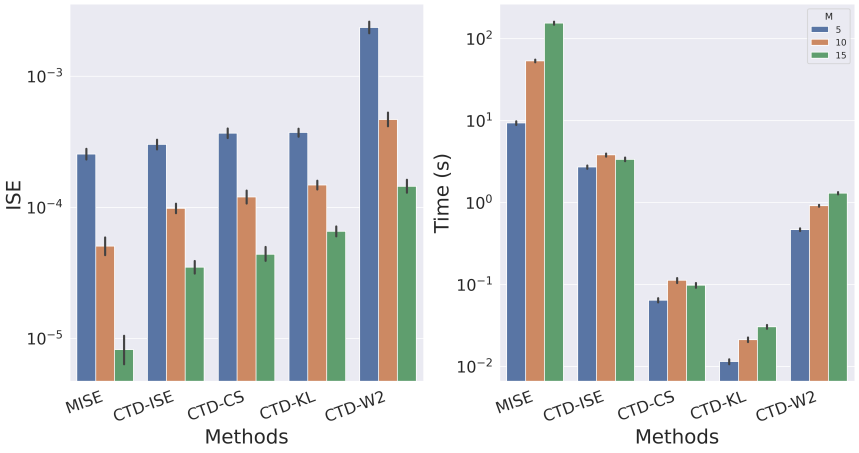
The reduction results for the original mixture in Figure 5.3 (a) are given in Figure 5.4 under different reduction approaches and values of M. Knowing how the original mixture is created, the reduction should be most satisfactory with $M = 5$. Without the knowledge of creation, we may attempt to reduce its order to $M = 10$ and $M = 15$.

We apply all GMR approaches to these $R = 100$ order $N = 25$ bivariate Gaussian mixtures. We compute the ISE values between the reduced and original mixtures. We visualize the average of these 100 ISE values and 95% error bar in Figure 5.3 (c). The average computational time and the 95% error bar of each reduction method is given in Figure 5.3 (d). The three attached bars are results for $M = 5, 10, 15$ respectively.

MISE reduction approach has the smallest ISE by definition. This method therefore serves as a baseline for comparison. The low ISE of MISE is at the cost of high computational time. The ISE decreases and the computation time increases when M increases from 5 to 10 and then to 15 for all reduction approaches. The performances of the proposed CTD-based GMR approaches vary with the choice of the cost function. Their relative performance is consistent regardless of the value of M. In terms of ISE, the preference for the cost function from high to low is ISE, CS, KL, and W2. In terms of the computational time, the preference for the cost function from high to low is KL, CS, W2, and ISE. If we use ISE as the cost function in the proposed approach, the result is nearly as good as the direct ISE reduction method (MISE) when $M = 5$, but at $1/10$ of the computation time. The CTD-KL approach only takes about $1/1000$ computation time of the MISE. Based on Figure 5.4, CTD-KL and CTD-W2 are still quite good. All CTD-based approaches for reduction are satisfactory.



(a) Cluster centres **(b)** Density heat-map



(c) Integrate Squared Error **(d)** Computation time

Figure 5.3: (a) The location of 25 mean vectors in one randomly generated mixture, (b) The density heat-map of the randomly generated mixture, (c) the average ISE between the reduced and original mixtures and the 95% error bar, and (d) the total computational time. Three attached bars are results for $M = 5, 10, 15$ respectively.

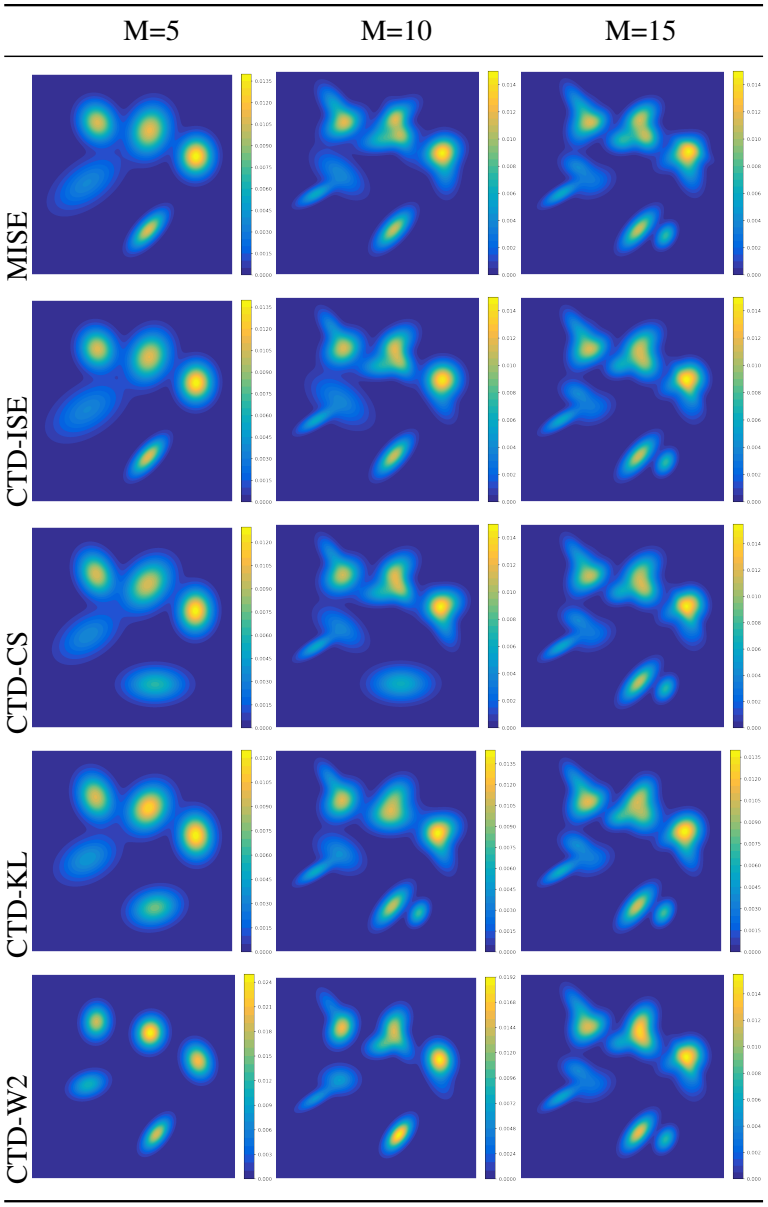


Figure 5.4: Heat-maps of density functions of reduced mixtures from one generated original mixture whose heat-map is in Figure 5.3.

5.4.2 Approximate Inference for Belief Propagation

In this section, we apply the proposed GMR to the graphical model represented by Figure 5.5 (a) following Yu et al. (2018). The goal of this experiment is to precisely approximate the marginal distribution of the random variable associated with each node. In this model (5.1), the local potential associated with the (i, j)th edge is given by $\psi_{ij}(x, y) = \phi(x; y, \phi_{ij}^{-1})$, where ϕ_{ij} values are marked alongside the graph edges in the figure. The local evidence potential associated with the ith node is a two-component Gaussian mixture

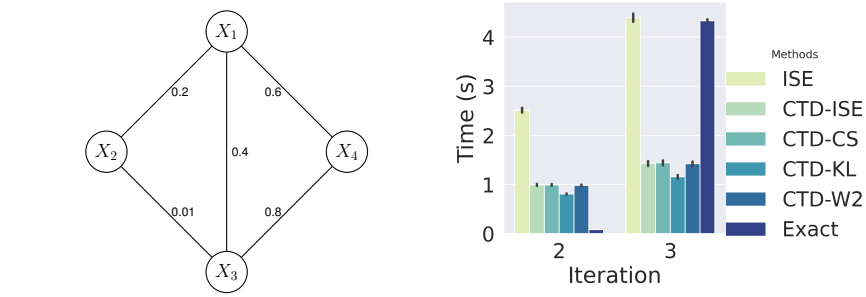
$$\psi_i(x) = w_i\phi(x; \mu_{i1}, 1) + (1 - w_i)\phi(x; \mu_{i2}, 1.5),\ i = 1, 2, 3, 4.$$

for some w_i, μ_{i1}, and μ_{i2} values. In this experiment, we create $R = 100$ graphic model specifications with these constant values in each specification generated as follows: $w_i \overset{\text{i.i.d.}}{\sim} U(0, 1)$, $\mu_{i1} \overset{\text{i.i.d.}}{\sim} U(-4, 0)$, and $\mu_{i2} \overset{\text{i.i.d.}}{\sim} U(0, 4),\ i = 1, 2, 3, 4$.

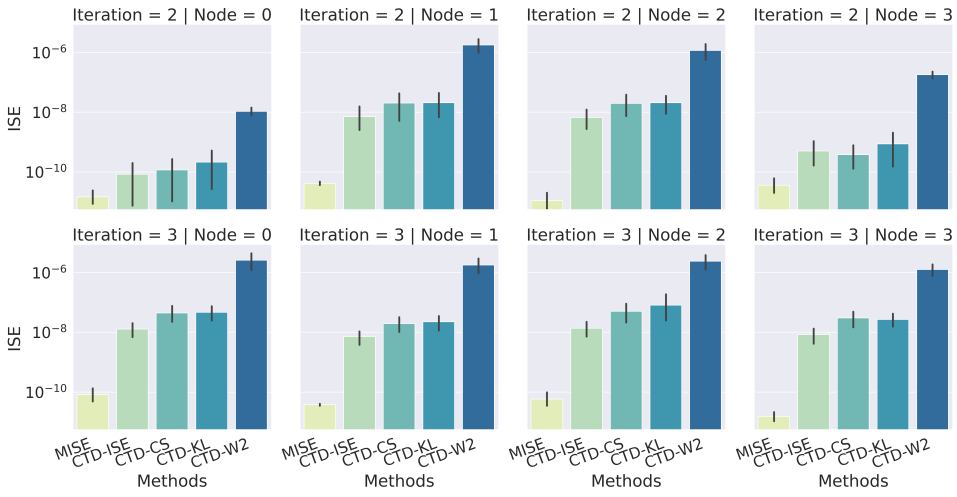
The marginal distributions of X_i are completely determined by these potentials. However, it is difficult to compute their density functions. The iterations in the BP introduced in Section 5.1.1 involves message mixtures whose orders grow exponentially. The *exact inference* hence becomes intractable after merely 4 iterations. One way to overcome this difficulty is to reduce the order of the message mixture after each iteration before it is used for updating the belief in the next iteration. This is so called *approximate inference*. The proposed GMR can be used here to keep the order at a manageable size. In this experiment, we reduce the order of the message mixture to $M = 4$ whenever its order exceeds 4 after an iteration following Yu et al. (2018).

We evaluate the performance of the GMR approaches by ISE between the exact belief and the approximate beliefs. The comparison is computationally feasible only for the first 3 iterations due to limited computer memory. Since no reduction is applied in the first iteration, we only obtain the result for the 2nd and 3rd iterations. The results are averaged over 100 trials with the corresponding 95% error bars.

Figure 5.5 (c) gives the distance of the belief based on the approximate inference to the true belief mixture based on the exact inference at four nodes. As the iteration increases, the ISE gets larger. It can be also seen from Figure 5.5 (c) that the approximate inference based on ISE is most accurate. For all minimum



(a) Graphical model **(b)** Time



(c) ISE

Figure 5.5: (a) The structure of the graphical model, (b) computation time for belief update versus number of iteration, and (c) the ISE between the exact and approximate beliefs.

CTD-based reduction approaches, when the cost function is the ISE, the approximate inference has the best results. In terms of the computational time, the MISE approach that is the closest to the exact inference does not save the computational time. In the 3rd iteration, the order of the message mixture is very large in the exact inference, the CTD-based approaches save the computational time and the beliefs obtained based on the approximate inference are very close to those based on the exact inference.

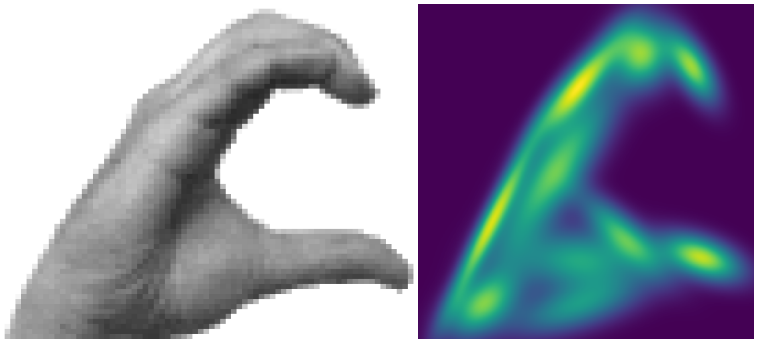
5.4.3 Hand Gesture Recognition

We apply the GMR for static hand gesture recognition in this section. For static hand gesture recognition, a set of labelled images of hand gestures are given as the training set. A classifier is trained to classify unseen images of the same set of hand gestures.

Dataset & Pre-processing We use the Jochen Triesch static hand posture database in Triesch and Von Der Malsburg (1996) that is publicly available online. This dataset contains 128×128 grey-scale images of 10 hand postures forming the alphabetic letters: A, B, C, D, G, H, I, L, V, and Y by 24 persons with 3 different backgrounds. To remove additional noise caused by the background, in our experiment, we use the same set of images as described in Kampa et al. (2011) whose backgrounds are removed. To reduce the classification error caused by the misalignment of the hands, Kampa et al. (2011) centres these hands by cropping. They manually crop each image into the smallest rectangle that only contains the hand and whose centre is the centre of the hand. After this step, all hands are centred in the image but with different sizes due to the difference in the hand sizes in the original images. To make the classifiers less dependent on the size of the hand, they resize the images into a square whose most top-left pixel and most bottom-right pixel have coordinates $(0, 0)$ and $(1, 1)$ respectively. After these pre-processing steps, there are 168 images in total with around $16 - 20$ images for each hand posture.

Gaussian Mixture & Hand Gesture Recognition Kampa et al. (2011) view the intensity of each pixel as a function of the location. They approximate this function by the density function of a Gaussian mixture up to some normalizing constant. They therefore obtain a 10-component Gaussian mixture from each image with the non-background pixels. Each image is then represented by a 2-dimensional Gaussian mixture. An example of the original image and the heat-map of the density function of the corresponding fitted mixture model is given in Figure 5.6.

Kampa et al. (2011) classify new image based on CS divergence between a test image and all training images. The test image is classified by the nearest neighbour method. A test image of hand gesture is classified as gesture "A" if there is a training image with hand gesture "A" that is closest to this test image. This approach



(a) Pre-processed image **(b)** Heat-map of GMM

Figure 5.6: An example of (a) a pre-processed image of hand posture "C"; (b) the heat-map of the order 10 Gaussian mixture of a pre-processed image.

achieves a classification accuracy of 95.2%.

We use a slightly different strategy from Kampa et al. (2011). We first combine the training images of the same hand gesture into a single training image. Since each image is a GMM, the combined image of the same hand gesture is also a GMM except for a much higher order. We then use proposed GMR methods to reduce its order to 10. We call the resultant GMM as class prototypes of the hand gestures. The nearest neighbour classifier is then applied with respect to prototypes. When the number of training examples is very large, our approach is worthwhile for its efficiency in computation, perhaps at some loss in classification accuracy. The purpose of this example is to demonstrate the feasibility, not to have a superior classifier immediately. Figure 5.7 gives the prototypes of the hand gestures based on different reduction approaches.

Results The quality of the class prototypes must have an effect on the classification accuracy. We compare the classification accuracy and computation time of various reduction methods. Since the training set is relatively small, we perform 5-fold cross validation that is repeated 100 times to estimate the classification accuracy.

We consider two schemes during the test:

1. We use the same divergence for classification and for reduction. That is, we minimize the ISE to obtain the class prototype, and use ISE to measure

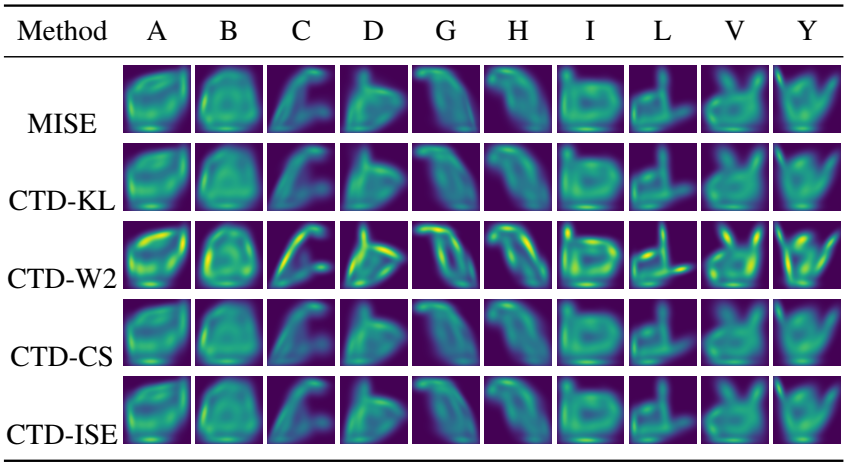


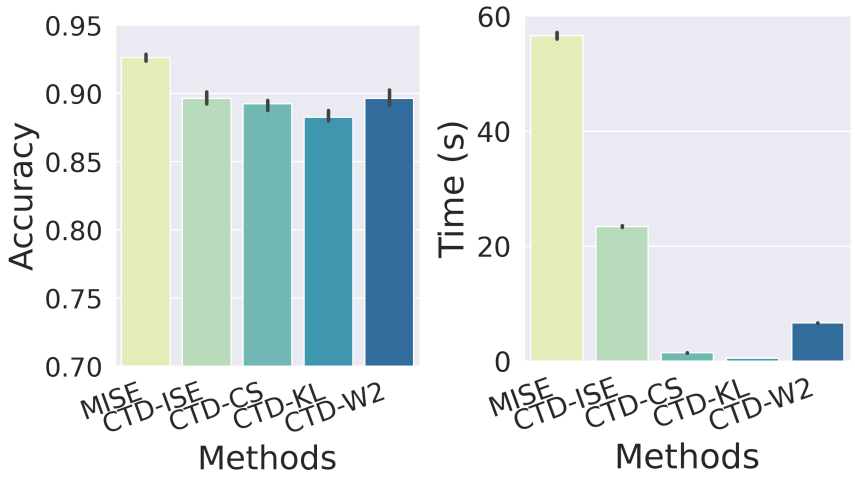
Figure 5.7: The class prototypes of hand gestures obtained by different reduction approaches.

the similarity between the test images and the class prototypes. Similarly for other CTD-based divergences. Figure 5.8 (a) depicts the classification accuracy applying this strategy based on several divergences.

2. We also obtain results by using different divergences for classification and for reduction. For example, we can minimize the ISE to obtain the class prototype, but use CS to measure the similarity between the test images and the class prototypes. Figure 5.8 (b) depicts the classification accuracy applying this strategy based on different combinations of divergences.

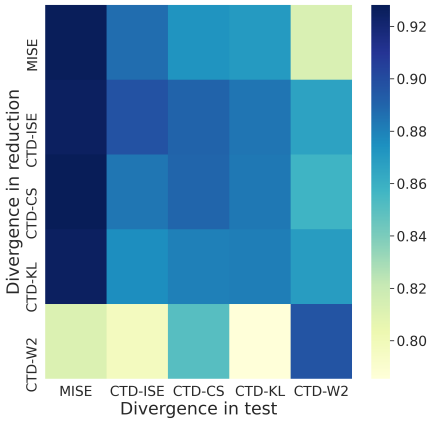
When the same divergence is used in the reduction and test, the MISE approach attains the highest classification accuracy. However, it takes most computation time. Using CTD-based divergences leads to slightly lower classification accuracy. However, CTD-CS and CTD-ISE beat CTD-KL while CTD-KL is most efficient in computation time.

When different divergences are used during the reduction and test, it is best to use ISE for the test. All reduction approaches except for CTD-W2 lead to high classification accuracy. Combining the considerations in computation time and classification accuracy, we recommend to use the CTD-KL to perform reduction and ISE for test.



(a) Classification Accuracy

(b) Computation Time



(c) Classification Accuracy

Figure 5.8: The (a) classification accuracy when the same divergence is used in the reduction and test, (b) computational time based on different reduction approaches, (c) classification accuracy when different divergences are used in the reduction and test on the hand gesture dataset.

5.5 Conclusion

In this chapter, we propose a new optimization-based GMR approach through composite transportation divergence (CTD). We establish the convergence of the accompanying iterative MM algorithm. Our approach includes the clustering-based and optimization-based approaches for GMR in the literature as our special cases. Therefore, our results provide optimality targets to many existing clustering-based approaches, support their usefulness, as well as establish their algorithm convergence.

The proposed GMR methods with various CTD experimented with so far do not achieve as high an efficiency as the MISE-based optimization approach. However, the proposed GMR methods have the computational simplicity of the clustering-based algorithms. In addition, our GMR approach allows flexible cost function choices in its CTD. This flexibility opens up a big room for potential improvement of the proposed GMR approach. We leave it a future project to search for a near perfect cost function.

Chapter 6

Beyond Gaussian Mixture

The finite Gaussian mixtures are by far the most studied finite mixture model (Chen and Li, 2009; Chen et al., 2012; Lo et al., 2001; Scrucca et al., 2016; Xu and Jordan, 1996). Other mixtures such as the mixture of Binomial distributions, Poisson distributions, and Gamma distributions are also broadly investigated in the literature and used in applications. For the split-and-conquer learning of mixtures in Chapter 4 and mixture reduction in Chapter 5, we only focus on the finite Gaussian mixtures though the methods can be generalized for mixture of distributions from other distribution families. We extend our proposed method to mixtures of distributions from exponential families in this chapter.

For both split-and-conquer learning of mixtures and the mixture reduction, the fundamental task is the approximation of a high order mixture by one with a lower order. The approximation problem is formulated as an optimization problem where the lower order mixture is found by minimizing the entropic regularized Composite Transportation Divergence (CTD) between two mixtures and an Majorization Maximization (MM) algorithm is developed accordingly. Chapter 5 shows that there are various choices for the cost function such as the Kullback-Leibler (KL) divergence, Integrated Squared Error (ISE), and Cauchy-Schwarz (CS) divergence. To use these divergences as the cost function in the proposed MM algorithm, the key is to have an easy-to-use form of the divergence for the assignment step and the corresponding barycentres for the update step. We therefore study the form of KL divergence, ISE, and CS divergence for mixtures of distributions in exponential families and

their corresponding barycentres.

Recall that the density function of a distribution from exponential family can be written as

$$f(x;\theta) = \exp(\theta^\top T(x) - A(\theta)) \tag{6.1}$$

with respect to some reference measure $\nu(\cdot)$. We call $\theta = (\theta_1, \theta_2, \ldots, \theta_m)^\top$ the natural parameter and $T(x) = (T_1(x), T_2(x), \ldots, T_m(x))^\top$ the natural sufficient statistics. The natural parameters and natural sufficient statistics of some widely used distributions in the exponential family are given in Table 1.1. An exponential family is *regular* if the parameter space Θ is an open set. The exponential family is *minimal* if the functions in $T(x)$ are linearly independent. Without loss of generality, we assume the exponential families in our discussion are minimal.

Based on the parameterization of the exponential family, it is easy to see that

$$A(\theta) = \log \int \exp\{\theta^\top T(x)\}\, \nu(dx). \tag{6.2}$$

Moreover, as shown in Wainwright and Jordan (2008), the log partition function $A(\cdot)$ is a convex function of θ. When exponential family is minimal, then $A(\cdot)$ is strictly convex. We cite some properties of the exponential family in Wainwright and Jordan (2008) that is useful in our discussion.

Lemma 6.1 (Properties of the Exponential Family)**.** *Let the log partition function $A(\theta)$ be defined above and $p(x)$ be a density function with respect to ν that is not necessarily in the exponential family. Let*

$$\mathcal{M} = \{\mu \in \mathbb{R}^d : \exists\, p(x) \text{ s.t. } \mathbb{E}_{X\sim p}\{T(X)\} = \mu\},$$

then

1. *$\nabla A(\theta) = \mathbb{E}_{X\sim f(x;\theta)}\{T(X)\} := \mathbb{E}_\theta\{T(X)\}$ is a one-to-one mapping between Θ and ν if and only if the exponential family is minimal.*

2. *In a minimal exponential family, for each μ that is an interior point of $\mathcal{M}$, there exists some $\theta \in \Theta$ so that $\mathbb{E}_\theta\{T(X)\} = \mu$.*

With these properties of exponential family, we discuss the assignment step and

the update step in the MM algorithm for mixture reduction when $\mathcal{F}$ is an exponential family.

Assignment Step

In the assignment step of the proposed MM algorithm in Algorithm 4, the optimal transportation plan at each iteration is a function of the cost function, and the plan can be expressed as a matrix made of the pairwise distance between one subpopulation of the original mixture and another subpopulation of the candidate mixture. Therefore, to have an easy-to-use transportation plan, it is critical to get the closed-form cost function under the mixture of distributions for a given exponential family. In this section, we show the pairwise KL divergence, ISE, and CS divergence have closed-forms under any exponential families.

Let $F(\cdot;\theta_1)$ and $F(\cdot;\theta_2)$ be two distributions from the same exponential family. Then the KL divergence between these two distributions is

$$\begin{aligned} &D_{\mathrm{KL}}(F(\cdot;\theta_1)\|F(\cdot;\theta_2)) = \int f(x;\theta_1)\log\frac{f(x;\theta_1)}{f(x;\theta_2)}\,\nu(dx) \\ =&\int f(x;\theta_1)\left\{(\theta_1-\theta_2)^\top T(x) - (A(\theta_1)-A(\theta_2))\right\}\nu(dx) \\ =&(\theta_1-\theta_2)^\top\nabla A(\theta_1) - (A(\theta_1)-A(\theta_2)) \end{aligned} \tag{6.3}$$

where

$$\nabla A(\theta) = \mathbb{E}_\theta\{T(X)\} = \int T(x)f(x;\theta)\,\nu(dx).$$

The form of ∇A for widely used exponential family is given in Table 6.1. Therefore, the pairwise KL divergence between subpopulations has a closed-form.

To find the CS divergence and ISE between two distributions from the same exponential family, we make use of the following Lemma.

Lemma 6.2. *Let $f(x;\theta_1)$ and $f(x;\theta_2)$ be two distributions from the same exponential family, we have*

$$\int f(x;\theta_1)f(x;\theta_2)\,\nu(dx) = \exp\{A(\theta_1+\theta_2) - A(\theta_1) - A(\theta_2)\}.$$

Applying this result, the CS divergence between $F(\cdot;\theta_1)$ and $F(\cdot;\theta_2)$ is given

by

$$\begin{aligned} & D_{\text{CS}}(F(\cdot;\theta_1), F(\cdot;\theta_2)) \\ =& -\log \frac{\int f(x;\theta_1) f(x;\theta_2)\, \nu(dx)}{\sqrt{\int f^2(x;\theta_1)\, \nu(dx)} \sqrt{\int f^2(x;\theta_2)\, \nu(dx)}} \\ =& \{A(\theta_1) + A(\theta_2)\}/2 - A(\theta_1 + \theta_2) \end{aligned} \tag{6.4}$$

and the ISE between $F(\cdot;\theta_1)$ and $F(\cdot;\theta_2)$ is given by

$$\begin{aligned} & D_{\text{ISE}}(F(\cdot;\theta_1), F(\cdot;\theta_2)) = \int (f(x;\theta_1) - f(x;\theta_2))^2\, \nu(dx) \\ =& \left\{ \sum_{i=1}^{2} \exp\{A(2\theta_i) - 2A(\theta_i)\} \right\} \\ & - 2\exp\{A(\theta_1 + \theta_2) - A(\theta_1) - A(\theta_2)\}. \end{aligned} \tag{6.5}$$

The form of $A(\theta)$ for commonly used exponential families is given in Table 6.1. It can be seen that all three divergences between these types of subpopulations have closed-forms. Therefore, the optimal transportation plan (5.19) in the assignment step also has a closed-form.

Update Step

In the update step of the MM algorithm, we need to work on the barycentre of distributions with respect to these divergences. We discuss the barycentre of distributions from the same exponential family under different divergences in this section.

Theorem 6.1 (Barycentres of Distributions from an Exponential Family)**.** *Denote* $\{f(\cdot;\theta_n) \in \mathcal{F} : n \in [N]\}$ *as a set of density functions in the exponential family and* $\{F(\cdot;\theta_n) : n \in [N]\}$ *as their corresponding* CDF*s. The barycentre of* $\{F(\cdot;\theta_n) : n \in [N]\}$ *with weights* $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)^\top \in \Delta_{N-1}$ *under divergence* $D(\cdot,\cdot)$ *is* $F(\cdot;\bar{\theta})$ *such that*

$$\bar{\theta} = \arg\min_{\theta \in \Theta} \sum_{n=1}^{N} \lambda_n D(F(\cdot;\theta_n), F(\cdot;\theta)).$$

Table 6.1: Parameter specification and statistics of widely used density functions in full exponential family. In all cases, the base measure ν is Lebesgue measure that is modulated by a factor $h(\cdot)$. The $\psi(\cdot)$ is the digamma function.

$\mathcal{F}$	$T(x)$	$A(\theta)$
	Univariate distribution	
Exponential	x	$-\log(-\theta)$
Weibull (known k)	x^k	$-\log(-\theta)$
Laplace (known μ)	$\lvert x-\mu\rvert$	$\log(-2/\theta)$
Rayleigh	x^2	$-\log(-2\theta)$
Log-normal	$(\log x, \log^2 x)^\top$	$-\theta_1^2/\theta_2 - 1/\sqrt{2\theta_2}$
Gamma	$(\log x, x)^\top$	$\log\Gamma(\theta_1+1) - (\theta_1+1)\log(-\theta_2)$
Inverse Gamma	$(\log x, 1/x)^\top$	$\log\Gamma(-\theta_1-1)) + (\theta_1+1)\log(-\theta_2)$
	Multivariate distribution	
Gaussian Gamma	$(\log\tau, \tau, \tau x, \tau x^2)^\top$	$\log\Gamma(\theta_1+\frac{1}{2}) - \frac{1}{2}\log(-2\theta_4) - (\theta_1+\frac{1}{2})\log(-\theta_2+\frac{\theta_3^2}{4\theta_4})$
Dirichlet	$\log \boldsymbol{x}$	$\{\mathbb{1}_K^\top \log\Gamma(\boldsymbol{\theta}+1) - \log\{\mathbb{1}_K^\top\Gamma(\boldsymbol{\theta}+1)\}$

$\mathcal{F}$	$\nabla A(\theta)$
	Univariate distribution
Exponential	$-1/\theta$
Weibull (known k)	$k(-1/\theta)^{\frac{k-1}{k}}$
Laplace (known μ)	$-1/\theta$
Rayleigh	$-\frac{1}{\theta}$
Log-normal	$(-\frac{\theta_1}{2\theta_2}, \frac{\theta_1^2}{4\theta_2^2} - \frac{1}{2\theta_2})^\top$
Gamma	$(\psi(\theta_1+1) - \log(-\theta_2), -\frac{\theta_1+1}{\theta_2})^\top$
Inverse Gamma	$(\log(-\theta_2) - \psi(-(\theta_1+1)), \frac{\theta_1+1}{\theta_2})^\top$
	Multivariate distribution
Gaussian Gamma	$(\psi(\theta_1+\frac{1}{2}) - \log(\frac{\theta_3^2}{4\theta_4} - \theta_2), \frac{4\theta_1\theta_4+2\theta_4}{\theta_3^2-4\theta_2\theta_4}, -\frac{\theta_3(2\theta_1+1)}{\theta_3^2-4\theta_2\theta_4}, \frac{\theta_3^2(2\theta_1+1)}{2\theta_4(\theta_3^2-4\theta_2\theta_4)} - \frac{1}{2\theta_4})^\top$
Dirichlet	$\psi(\boldsymbol{\theta}+1) - \psi(\mathbb{1}_K^\top\boldsymbol{\theta}+K)$

We have the following results:

(a) when $D(F(\cdot;\theta_n), F(\cdot;\theta)) = D_{\mathrm{KL}}(F(\cdot;\theta_n)\|F(\cdot;\theta))$*, then* $\bar{\theta}$ *is the solution to*

$$\sum_{n=1}^{N} \lambda_n \nabla A(\theta_n) = \nabla A(\theta).$$

(b) when $D(F(\cdot;\theta_n), F(\cdot;\theta)) = D_{CS}(F(\cdot;\theta_n), F(\cdot;\theta))$*, then* $\bar{\theta}$ *is a solution to*

$$\nabla A(\theta) = 2\sum_{n=1}^{n} \lambda_n \nabla A(\theta_n + \theta).$$

(c) when $D(F(\cdot;\theta_n), F(\cdot;\theta)) = D_{ISE}(F(\cdot;\theta_n), F(\cdot;\theta))$*, then* $\bar{\theta}$ *is a minimizer of*

$$\exp\{A(2\theta) - 2A(\theta)\} - 2\sum_{n} \lambda_n \exp\left\{A(\theta_n + \theta) - A(\theta_n) - A(\theta)\right\}.$$

With these simple characterizations of the barycentre under these divergences, we can easily update the cluster centres in the corresponding MM algorithms. Their MM algorithms are therefore easy to carry out. We now give more details of the derivation of the barycentres in Theorem 6.1.

KL Barycentre of Distributions in an Exponential Distribution Family By (6.3), the KL barycentre of $\{F(\cdot;\theta_n) : n \in [N]\}$ is $F(\cdot;\bar{\theta})$ such that $\bar{\theta}$ minimizes

$$\mathcal{L}(\theta) = A(\theta) - \theta^\top \left\{\sum_{n=1}^{N} \lambda_n \nabla A(\theta_n)\right\} + C$$

where C is a constant that does not depend on θ. By the property of exponential family, $A(\theta)$ is convex in θ. Hence, $\mathcal{L}(\theta)$ is also convex in θ. As a result, $\bar{\theta}$ is the solution of $\nabla\mathcal{L}(\theta) = 0$, which is simplified to

$$\sum_{n=1}^{N} \lambda_n \nabla A(\theta_n) = \nabla A(\theta). \tag{6.6}$$

This result shows that the KL barycentre can be obtained easily. Since $\nabla A(\theta) = \mathbb{E}_\theta\{T(X)\}$, equation (6.6) lines up the expectation of $T(X)$ of the barycentre with the convex combination of those of original distributions. The second property of Lemma 6.1 ensures the existence of the solution.

CS Barycentre of Distributions in an Exponential Distribution Family By (6.4),

the CS barycentre of $\{F_n : n \in [N]\}$ is $F(\cdot; \bar{\theta})$ such that $\bar{\theta}$ minimizes

$$\frac{1}{2}A(\theta) - \sum_{n=1}^{N} \lambda_n A(\theta_n + \theta)$$

up to some additive constant. To find the CS barycentre, we look for the stationary point of the objective function, which is the solution to

$$\frac{1}{2}\nabla A(\theta) = \sum_{n=1}^{n} \lambda_n \nabla A(\theta_n + \theta).$$

We therefore have $\bar{\theta}$ be the solution to

$$\nabla A(\theta) = 2\sum_{n=1}^{n} \lambda_n \nabla A(\theta_n + \theta).$$

ISE Barycentre of Distributions from Exponential Family By (6.5), the barycentre $F(\cdot; \bar{\theta})$ under ISE has its $\bar{\theta}$ minimizes

$$\exp\{A(2\theta) - 2A(\theta)\} - 2\sum_{n} \lambda_n \exp\left\{A(\theta_n + \theta) - A(\theta_n) - A(\theta)\right\}.$$

This function is non-convex in general and numerical methods can be used to find a local minimum.

We have presented all ingredients needed to carry out our proposed MM algorithm, which makes it possible to perform reduction and distributed learning under finite mixtures of distributions from an exponential distribution family. We limit our investigation in this Chapter to theoretical discussion without going into numerical experiments. Clearly, empirical comparison of various approaches for reduction and distributed learning under finite mixtures of distributions from an exponential family may lead to a lot of repetition of contents in previous chapters. Considering the effort and the length of the thesis, we leave the empirical comparison as future work.

Chapter 7

Conclusions

The major contributions of this thesis are to develop novel methods for (a) the distributed learning of finite Gaussian mixtures, and (b) Gaussian Mixture Reduction (GMR) for approximate inference. The Composite Transportation Divergence (CTD) between two mixtures, which is a byproduct of the Optimal Transport (OT) theory, is used as an effective tool for both developments.

For split-and-conquer learning of finite Gaussian mixture models with the distributed dataset, we propose a novel aggregation method by reduction. We show that the proposed estimator is both statistically and computationally efficient. It also outperforms existing approaches for split-and-conquer learning of finite Gaussian mixtures. The proposed aggregation approach is also empirically shown to be robust against the non-random partition of the dataset. We also investigate its robustness when the order of the mixture on the local machine is over-specified. Empirical experiments show that the information of the mixture retains in local estimators, but more effective reduction procedures are yet to be discovered.

We also propose a general framework for GMR by minimizing the CTD between two mixtures. We show the proposed framework connects the existing clustering-based and optimization-based algorithms for GMR. Existing clustering-based algorithms are special cases of our proposed framework with specific cost functions in the CTD. We also show that the proposed method optimizes an upper bound of the objective function in the existing optimization-based algorithms for GMR. With so many potential cost functions to choose from, we see the potential to improve the

performance of the existing clustering-based algorithms for GMR. Although our discussion focuses on finite Gaussian mixtures, we show that the proposed framework can be easily adapted to mixtures of distributions from other distribution families.

The storage of datasets in a distributed fashion brings a lot of challenges to effective statistical inference. For example, there might be bandwidth constraints for the amount of information that can be sent across machines (Parras and Zazo, 2020), data heterogeneity (Jeong et al., 2018), and the well-known Byzantine failures problem (Chen et al., 2017; Tu et al., 2021) where erroneous information is communicated due to hardware breakdowns, data crashes, or communication failures. Many of these new challenges have not been studied in the context of mixtures yet, which are left as the future work of this thesis. In our proposed method, we assume of the order of the mixture is known, which may not always be true in applications. There are various potential approaches to tackle this problem. For example, over-specified mixtures can be fitted on local machines as in our preliminary study. The existing order selection approach can also be first applied to local machines. If the aggregation approach with known order assumption is used in these scenarios, do these estimators perform? Which estimator has better performance? The above two candidate approaches when the order is not known still assume the order is known on the central machine. Under the split-and-conquer learning setting where only the summary statistics are allowed to transmit to the central machine, how can one decide the order for aggregation on the central machine given only the summary statistics? We intend to investigate these problems in the future.

Bibliography

Agrawal, R., Evfimievski, A., and Srikant, R. (2003). Information sharing across private databases. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 86–97. ACM. → pages 10, 73

Agueh, M. and Carlier, G. (2011). Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924. → pages 41, 42

Alistarh, D., Allen-Zhu, Z., and Li, J. (2018). Byzantine stochastic gradient descent. *arXiv preprint arXiv:1803.08917*. → page 75

Anderson, B. D. and Moore, J. B. (2012). *Optimal Filtering*. Courier Corporation. → page 124

Ardeshiri, T., Özkan, E., and Orguner, U. (2013). On reduction of mixtures of the exponential family distributions. Technical report, Linköping University. → page 12

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*. → page 10

Assa, A. and Plataniotis, K. N. (2018). Wasserstein-distance-based Gaussian mixture reduction. *IEEE Signal Processing Letters*, 25(10):1465–1469. → pages 12, 119, 127, 130, 131

Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120. → page 21

Baldry, I. K., Balogh, M. L., Bower, R., Glazebrook, K., and Nichol, R. C. (2004). Color bimodality: implications for galaxy evolution. In *AIP Conference Proceedings*, volume 743, pages 106–119. AIP. → page 2

Baldwin, S. (2012). Compute Canada: advancing computational research. In *Journal of Physics: Conference Series*, volume 341, page 012001. IOP Publishing. → page 92

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2015). Distributed estimation and inference with statistical guarantees. *arXiv preprint arXiv:1509.05457*. → page 11

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353. → page 3

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. → pages 3, 70

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128. → page 75

Blum, J. and Susarla, V. (1977). Estimation of a mixing distribution function. *The Annals of Probability*, 5(2):200–209. → pages 8, 9, 48

Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers Inc. → page 89

Brubaker, M. A., Geiger, A., and Urtasun, R. (2015). Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665. → page 4

Bučar, T., Nagode, M., and Fajdiga, M. (2004). Reliability approximation using finite Weibull mixture distributions. *Reliability Engineering & System Safety*, 84(3):241–251. → page 4

Cappé, O. and Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 71(3):593–613. → page 89

Carta, J. and Ramirez, P. (2007). Analysis of two-component mixture Weibull statistics for estimation of wind speed distributions. *Renewable Energy*, 32(3):518–531. → page 7

Chang, X., Lin, S.-B., and Wang, Y. (2017). Divide and conquer local average regression. *Electronic Journal of Statistics*, 11(1):1326–1350. → pages 11, 74

Chen, J. (1995). Optimal rate of convergence for finite mixture models. *The Annals of Statistics*, 23(1):221–233. → page 25

Chen, J. and Li, P. (2009). Hypothesis test for normal mixture models: the EM approach. *The Annals of Statistics*, 37(5A):2523–2542. → pages 5, 161

Chen, J., Li, P., and Fu, Y. (2012). Inference on the order of a normal mixture. *Journal of the American Statistical Association*, 107(499):1096–1105. → pages 5, 161

Chen, J., Li, P., and Liu, G. (2020). Homogeneity testing under finite location-scale mixtures. *Canadian Journal of Statistics*, 48(4):670–684. → page 58

Chen, J., Li, S., and Tan, X. (2016). Consistency of the penalized MLE for two-parameter Gamma mixture models. *Science China Mathematics*, 59(12):2301–2318. → page 23

Chen, J. and Tan, X. (2009). Inference for multivariate normal mixtures. *Journal of Multivariate Analysis*, 100(7):1367–1383. → pages 22, 23, 24, 25, 61

Chen, J., Tan, X., and Zhang, R. (2008). Inference for normal mixtures in mean and variance. *Statistica Sinica*, 18(2):443–465. → pages 22, 24

Chen, W.-C., Ostrouchov, G., Pugmire, D., Prabhat, and Wehner, M. (2013). A parallel EM algorithm for model-based clustering applied to the exploration of large spatio-temporal data. *Technometrics*, 55(4):513–523. → pages 101, 102, 103

Chen, X. and Xie, M. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 24(4):1655–1684. → page 11

Chen, Y., Lin, Z., and Muller, H.-G. (2021). Wasserstein regression. *Journal of the American Statistical Association*, 0(0):1–14. → page 30

Chen, Y. and Muller, H.-G. (2021). Wasserstein gradients for the temporal evolution of probability distributions. *Electronic Journal of Statistics*, 15(2):4061–4084. → page 30

Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25. → page 169

Chernoff, H. and Lander, E. (1995). Asymptotic distribution of the likelihood ratio test that a mixture of two binomials is a single binomial. *Journal of Statistical Planning and Inference*, 43(1-2):19–40. → page 5

Choi, K. (1969). Estimators for the parameters of a finite mixture of distributions. *Annals of the Institute of Statistical Mathematics*, 21(1):107–116. → pages 8, 9, 48

Choi, K. and Bulgren, W. (1968). An estimation procedure for mixtures of distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(3):444–460. → pages 8, 9, 48

Clark, A. (2015). Pillow (pil fork) documentation. → page 70

Clarke, B. and Heathcote, C. (1994). Robust estimation of k-component univariate normal mixtures. *Annals of the Institute of Statistical Mathematics*, 46(1):83–93. → pages 8, 9, 48

Constantinopoulos, C. and Likas, A. (2007). Unsupervised learning of Gaussian mixtures based on variational component splitting. *IEEE Transactions on Neural Networks*, 18(3):745–755. → page 5

Corbett, J. C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J. J., Ghemawat, S., Gubarev, A., Heiser, C., and Hochschild, P. (2013). Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):Article 8. → pages 10, 73

Crouse, D. F., Willett, P., Pattipati, K., and Svensson, L. (2011). A look at Gaussian mixture reduction algorithms. In *14th International Conference on Information Fusion*, pages 1–8. IEEE. → page 12

Cutler, A. and Cordero-Brana, O. I. (1996). Minimum Hellinger distance estimation for finite mixture models. *Journal of the American Statistical Association*, 91(436):1716–1723. → pages 8, 9, 48, 61

Cuturi, M. (2013). Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, pages 2292–2300. → page 34

Cuturi, M. and Doucet, A. (2014). Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693. → page 41

Dara, S. and Tumma, P. (2018). Feature extraction by using deep learning: a survey. In *2018 Second International Conference on Electronics,*

Communication and Aerospace Technology (ICECA), pages 1795–1801. IEEE. → page 199

Davis, J. V. and Dhillon, I. S. (2007). Differential entropic clustering of multivariate Gaussians. In *Advances in Neural Information Processing Systems 19*, pages 337–344. → page 119

Dedecker, J. and Merlevede, F. (2017). Behavior of the wasserstein distance between the empirical and the marginal distributions of stationary α-dependent sequences. *Bernoulli*, 23(3):2083–2127. → page 30

Delon, J. and Desolneux, A. (2020). A Wasserstein-type distance in the space of Gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970. → page 144

Deng, Y. and Du, W. (2009). The Kantorovich metric in computer science: a brief survey. *Electronic Notes in Theoretical Computer Science*, 253(3):73–82. → page 29

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3. → page 124

Dwivedi, R., Ho, N., Khamaru, K., Wainwright, M., Jordan, M., and Yu, B. (2020). Sharp analysis of expectation-maximization for weakly identifiable models. In *International Conference on Artificial Intelligence and Statistics*, pages 1866–1876. → page 25

Eguchi, S. and Copas, J. (2006). Interpreting Kullback–Leibler divergence with the Neyman–Pearson lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040. → pages 25, 48

Evans, S. N. and Matsen, F. A. (2012). The phylogenetic Kantorovich–Rubinstein metric for environmental sequence samples. *Journal of the Royal Statistical Society: Series B (Methodological)*, 74(3):569–592. → pages 10, 36

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2019). Distributed estimation of principal eigenspaces. *Annals of Statistics*, 47(6):3009–3031. → page 11

Feldman, D., Faulkner, M., and Krause, A. (2011). Scalable training of mixture models via coresets. In *Advances in Neural Information Processing Systems*, pages 2142–2150. → page 87

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631. → pages 3, 98

Fréchet, M. (1948). Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310. → page 41

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics New York. → page 98

Goldberger, J. and Roweis, S. T. (2005). Hierarchical clustering of a mixture model. In *Advances in Neural Information Processing Systems 17*, pages 505–512. → page 119

Grother, P. and Hanaoka, K. (2016). NIST special database 19 handprinted forms and characters 2nd edition. Technical report, National Institute of Standards and Technology. → page 99

Gumbel, E. J. (1954). *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*, volume 33. US Government Printing Office. → page 58

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13(2):795–800. → page 21

He, M. and Chen, J. (2021). Strong consistency of the MLE under two-parameter Gamma mixture models with a structural scale parameter. *Advances in Data Analysis and Classification*, pages 1–30. → page 6

Heinrich, P. and Kahn, J. (2018). Strong identifiability and optimal minimax rates for finite mixture estimation. *The Annals of Statistics*, 46(6A):2844–2870. → page 25

Hernández, J.-A. and Phillips, I. W. (2006). Weibull mixture model to characterise end-to-end internet delay at coarse time-scales. *IEE Proceedings-Communications*, 153(2):295–304. → page 7

Hershey, J. R. and Olsen, P. A. (2007). Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE. → page 27

Holzmann, H., Munk, A., and Stratmann, B. (2004). Identifiability of finite mixtures–with applications to circular distributions. *Sankhyā: The Indian Journal of Statistics*, pages 440–449. → page 52

Huber, M. F. and Hanebeck, U. D. (2008). Progressive Gaussian mixture reduction. In *2008 11th International Conference on Information Fusion*, pages 1–8. IEEE. → page 119

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37. → page 81

Jaini, P. and Poupart, P. (2016). Online and distributed learning of Gaussian mixture models by Bayesian moment matching. *arXiv preprint arXiv:1609.05881*. → pages 88, 89, 95

Jensen, J. H., Ellis, D. P., Christensen, M. G., and Jensen, S. H. (2007). Evaluation of distance measures between Gaussian mixture models of MFCCs. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 107–108. Austrian Computer Society. → page 27

Jenssen, R., Principe, J. C., Erdogmus, D., and Eltoft, T. (2006). The Cauchy–Schwarz divergence and Parzen windowing: connections to graph theory and Mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629. → page 28

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., and Kim, S.-L. (2018). Communication-efficient on-device machine learning: federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*. → page 169

Joe, H. and Zhu, R. (2005). Generalized Poisson distribution: the property of mixture of Poisson and comparison with negative binomial distribution. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 47(2):219–229. → page 6

Kampa, K., Hasanbelliu, E., and Principe, J. C. (2011). Closed-form Cauchy-Schwarz pdf divergence for mixture of Gaussians. In *The 2011 International Joint Conference on Neural Networks*, pages 2578–2585. IEEE. → pages 156, 157

Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401. → page 75

Liesenfeld, R. (2001). A generalized bivariate mixture model for stock price volatility and trading volume. *Journal of Econometrics*, 104(1):141–178. → page 2

Liu, C. and Rubin, D. B. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–648. → page 21

Liu, Q. and Ihler, A. T. (2014). Distributed estimation, information loss and exponential families. In *Advances in Neural Information Processing Systems 27*, pages 1098–1106. → pages 11, 74, 79, 85, 86, 91, 107

Lo, Y., Mendell, N. R., and Rubin, D. B. (2001). Testing the number of components in a normal mixture. *Biometrika*, 88(3):767–778. → pages 5, 161

Lucic, M., Faulkner, M., Krause, A., and Feldman, D. (2017). Training Gaussian mixture models at scale via coresets. *The Journal of Machine Learning Research*, 18(1):5885–5909. → pages 88, 91, 92, 95, 97

Macdonald, P. (1971). Comment on "An estimation procedure for mixtures of distributions" by Choi and Bulgren. *Journal of the Royal Statistical Society. Series B (Methodological)*, 33:326–329. → pages 8, 9, 48

Maitra, R. and Melnykov, V. (2010). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376. → page 92

Manzar, A. (2017). Recursive Bayesian filtering through a mixture of Gaussian and discrete particles. Master's thesis, Queen's University. → pages 12, 118

Marín, J. M., Rodriguez-Bernal, M., and Wiper, M. P. (2005). Using Weibull mixture distributions to model heterogeneous survival data. *Communications in Statistics–Simulation and Computation*, 34(3):673–684. → page 7

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models*. John Wiley & Sons. → pages xiii, 4

Meilijson, I. (1989). A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(1):127–138. → page 21

Meinhold, R. J. and Singpurwalla, N. D. (1983). Understanding the Kalman filter. *The American Statistician*, 37(2):123–127. → page 124

Melnykov, V., Chen, W.-C., and Maitra, R. (2012). MixSim: an R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25. → page 92

Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80(2):267–278. → page 21

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. → page 88

Naya, S., Cao, R., de Ullibarri, I. L., Artiaga, R., Barbadillo, F., and García, A. (2006). Logistic mixture model versus Arrhenius for kinetic study of material degradation by dynamic thermogravimetric analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 20(3-4):158–163. → page 7

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer. → page 87

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313. → page 149

Nguyen, T. T., Nguyen, H. D., Chamroukhi, F., and McLachlan, G. J. (2020). Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics*, 7(1):1750861. → pages 4, 11, 117

Nguyen, X. (2013). Convergence of latent mixing measures in finite and infinite mixture models. *The Annals of Statistics*, 41(1):370–400. → pages 25, 30, 36, 40, 80, 144

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media. → page 56

Nowak, R. D. (2003). Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2245–2253. → pages 86, 87

Panaretos, V. M. and Zemel, Y. (2019). Statistical aspects of Wasserstein distances. *Annual Review of Statistics and its Application*, 6:405–431. → page 30

Parras, J. and Zazo, S. (2020). A graph network model for distributed learning with limited bandwidth links and privacy constraints. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3907–3911. IEEE. → page 169

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. → page 199

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110. → pages 2, 19

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. → page 91

Pernkopf, F. and Bouchaffra, D. (2005). Genetic-based EM algorithm for learning Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1344–1348. → page 5

Peyré, G. and Cuturi, M. (2019). Computational optimal transport: with applications to data science. *Foundations and Trends® in Machine Learning*, 11(5–6):355–607. → pages 29, 33, 35, 39, 83

Rabin, J., Peyré, G., Delon, J., and Bernot, M. (2011). Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer. → page 42

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850. → page 17

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239. → page 19

Ridolfi, A. and Idier, J. (2001). Penalized maximum likelihood estimation for univariate normal mixture distributions. In *AIP Conference Proceedings*, volume 568, pages 229–237. AIP. → page 22

Rousseau, J. and Mengersen, K. (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 73(5):689–710. → page 25

Runnalls, A. R. (2007). Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999. → pages 12, 119, 127, 128

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pixelcnn++: improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*. → page 7

Salmond, D. J. (1990). Mixture reduction algorithms for target tracking in clutter. In *Signal and Data Processing of Small Targets 1990*, volume 1305, page 434. International Society for Optics and Photonics. → pages 12, 119, 127, 128

Santosh, D. H. H., Venkatesh, P., Poornesh, P., Rao, L. N., and Kumar, N. A. (2013). Tracking multiple moving objects using Gaussian mixture model. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(2):114–119. → pages 4, 12

Schieferdecker, D. and Huber, M. F. (2009). Gaussian mixture reduction via clustering. In *2009 12th International Conference on Information Fusion*, pages 1536–1543. IEEE. → pages 12, 78, 119, 129, 130, 131, 139, 140

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464. → page 98

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, 8(1):289. → pages 5, 161

Solomon, J., Peyré, G., Kim, V. G., and Sra, S. (2016). Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):72. → page 42

Srivastava, S., Li, C., and Dunson, D. B. (2018). Scalable Bayes via barycenter in Wasserstein space. *The Journal of Machine Learning Research*, 19(1):312–346. → page 42

Sudderth, E. B., Ihler, A. T., Isard, M., Freeman, W. T., and Willsky, A. S. (2010). Nonparametric belief propagation. *Communications of the ACM*, 53(10):95–103. → pages xv, 12, 118, 121

Taylor, S. J. (1994). Modeling stochastic volatility: a review and comparative study. *Mathematical finance*, 4(2):183–204. → page 124

Teicher, H. (1961). Identifiability of mixtures. *The Annals of Mathematical Statistics*, 32(1):244–248. → page 52

Titterington, D. M., Afm, S., Smith, A. F., and Makov, U. (1985). *Statistical Analysis of Finite Mixture distributions*, volume 198. John Wiley & Sons Incorporated. → pages 4, 11, 117

Triesch, J. and Von Der Malsburg, C. (1996). Robust classification of hand postures against complex backgrounds. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 170–175. IEEE. → page 156

Tu, J., Liu, W., Mao, X., and Chen, X. (2021). Variance reduced median-of-means estimator for Byzantine-robust distributed inference. *Journal of Machine Learning Research*, 22(84):1–67. → pages 75, 169

Van der Vaart, A. W. (2000). *Asymptotic Statistics*, volume 3. Cambridge University Press. → pages 53, 191

Vasconcelos, N. and Lippman, A. (1999). Learning mixture hierarchies. In *Advances in Neural Information Processing Systems 11*, pages 606–612. → pages 119, 132

Villani, C. (2003). *Topics in Optimal Transportation*, volume 58. American Mathematical Society. → pages 16, 29, 30, 33, 36, 37, 144

Vlassis, N. and Likas, A. (2002). A greedy EM algorithm for Gaussian mixture learning. *Neural Processing Letters*, 15(1):77–87. → page 5

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc. → page 162

Wang, P., Cockburn, l. M., and Puterman, M. L. (1998). Analysis of patent data–a mixed-Poisson-regression-model approach. *Journal of Business & Economic Statistics*, 16(1):27–41. → page 6

West, M. (1993). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(2):409–422. → page 12

Wied, D. and Weißbach, R. (2012). Consistency of the kernel density estimator: a survey. *Statistical Papers*, 53(1):1–21. → pages 4, 117

Williams, J. L. (2003). Gaussian mixture reduction of tracking multiple maneuvering targets in clutter. Master's thesis, Air Force Institute of Technology. → pages 12, 28

Williams, J. L. and Maybeck, P. S. (2006). Cost-function-based hypothesis control techniques for multiple hypothesis tracking. *Mathematical and Computer Modelling*, 43(9–10):976–989. → pages 78, 119, 127, 128, 129, 139

Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103. → page 21

Wu, Y. and Huang, T. S. (2001). Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine*, 18(3):51–60. → page 121

Xie, C., Koyejo, O., and Gupta, I. (2018). Generalized Byzantine-tolerant SGD. *arXiv preprint arXiv:1802.10116*. → page 75

Xu, L. and Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8(1):129–151. → pages 5, 161

Yedidia, J. S., Freeman, W. T., Weiss, Y., et al. (2003). Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8(236-239):0018–9448. → pages 121, 122

Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. (2018). Byzantine-robust distributed learning: towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*. → page 75

Yu, L., Yang, T., and Chan, A. B. (2018). Density-preserving hierarchical EM algorithm: simplifying Gaussian mixture models for approximate inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1323–1337. → pages 4, 12, 78, 119, 127, 132, 133, 134, 141, 154

Zangwill, W. I. (1969). *Nonlinear Programming: A Unified Approach*, volume 52. Prentice-Hall Englewood Cliffs, New Jersey. → page 83

Zhang, K. and Kwok, J. T. (2010). Simplifying mixture models through function approximation. *IEEE Transactions on Neural Networks*, 21(4):644–658. → page 119

Zhang, L. and Liu, C. (2006). Fitting irregular diameter distributions of forest stands by Weibull, modified Weibull, and mixture Weibull models. *Journal of Forest Research*, 11(5):369–372. → page 7

Zhang, Q. and Chen, J. (2021a). Minimum Wasserstein distance estimator under finite location-scale mixtures. *arXiv preprint arXiv:2107.01323*. → page vi

Zhang, Q. and Chen, J. (2021b). Robustness of Gaussian mixture reduction for split-and-conquer learning of finite Gaussian mixtures. In *Proceedings of the 3rd International Conference on Statistics: Theory and Applications (ICSTA'21)*. Avestia. → page vi

Zhang, Q. and Chen, J. (2022). Distributed learning of finite Gaussian mixtures. *Journal of Machine Learning Research*, 0(0):0. → page vi

Zhang, Q., Zhang, A. G., and Chen, J. (2020). Gaussian mixture reduction with composite transportation distance. *arXiv preprint arXiv:2002.08410*. → page vi

Zhang, Y., Duchi, J., and Wainwright, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340. → pages 11, 74

Zhu, D. (2016). A two-component mixture model for density estimation and classification. *Journal of Interdisciplinary Mathematics*, 19(2):311–319. → page 61

Appendix A

Appendix for Chapter 3

Numerically friendly Expression of the Objective Function

We present the numerically friendly expression of the objective function of the Minimum Wasserstein Distance Estimator (MWDE) under mixture of location-scale families. To learn the finite mixture distribution through MWDE, we need to compute

$$\mathbb{W}_N(G) = W_2^2(F_N(\cdot), F(\cdot; G)) = \int_0^1 \{F_N^{-1}(t) - F^{-1}(t; G)\}^2 \, dt \tag{A.1}$$

for finite location-scale mixture

$$F(\cdot; G) = \sum_{k=1}^{K} \pi_k F(\cdot; \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \sigma_k^{-1} F_0((x - \mu_k)/\sigma_k).$$

We write $\mathbb{E}_k(\cdot)$ as expectation under distribution $F(\cdot; \boldsymbol{\theta}_k)$. For instance,

$$\mathbb{E}_k\{X^2\} = \mu_k^2 + \sigma_k^2(\mu_0^2 + \sigma_0^2) + 2\mu_k\sigma_k\mu_0.$$

Define intervals $I_n = ((n-1)/N, n/N]$ for $n = 1, 2, \ldots, N$ so that $F_N^{-1}(t) = x_{(n)}$ when $t \in I_n$, where $x_{(n)}$ is the nth order statistic. For ease of notation, we write

$x_{(n)}$ as x_n. Over this interval, we have

$$\begin{aligned} &\int_{I_n} \{F_N^{-1}(t) - F^{-1}(t;G)\}^2 \, dt \\ &= \int_{I_n} \{x_n^2 - 2x_n F^{-1}(t;G) + \{F^{-1}(t;G)\}^2\} \, dt. \end{aligned} \tag{A.2}$$

The integration of the first term in (A.2), after summing over n, is given by

$$\sum_{n=1}^{N} \int_{I_n} x_n^2 \, dt = N^{-1} \sum_n x_n^2.$$

Denote $\overline{x^2} = N^{-1} \sum_n x_n^2$, the integration of the third term in (A.2) is

$$\sum_{n=1}^{N} \int_{I_n} \{F^{-1}(t;G)\}^2 \, dt = \int_{-\infty}^{\infty} x^2 f(x;G) \, dx = \sum_{k=1}^{K} w_k \mathbb{E}_k\{X^2\}.$$

Let $\xi_0 = -\infty$, $\xi_{N+1} = \infty$, and $\xi_n = F^{-1}(n/N;G)$ for $n = 1, \ldots, N$. Denote

$$\Delta F_{nk} = F(\xi_n; \boldsymbol{\theta}_k) - F(\xi_{n-1}; \boldsymbol{\theta}_k)$$

and

$$T(x) = \int_{-\infty}^{x} t f_0(t) \, dt, \quad \Delta T_{nk} = T((\xi_n - \mu_k)/\sigma_k) - T(\xi_{n-1} - \mu_k)/\sigma_k).$$

Then

$$\begin{aligned} \int_{I_n} F^{-1}(t;G) \, dt &= \sum_k w_k \int_{\xi_{n-1}}^{\xi_n} x f(x; \mu_k, \sigma_k) \, dx \\ &= \sum_k w_k \{\mu_k \Delta F_{nk} + \sigma_k \Delta T_{nk}\}. \end{aligned}$$

These lead to numerically convenient expression

$$\mathbb{W}_N(G) = \overline{x^2} + \sum_k w_k \mathbb{E}_k\{X^2\} - 2 \sum_n \sum_k w_k \{\mu_k \Delta F_{nk} + \sigma_k \Delta T_{nk}\}.$$

Gradient of the Objective Function

To most effectively use Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, it is best to provide gradients of the objective function. Below is the numerically friendly expressions of gradients of $\mathbb{W}_N$ in (A.1).

Lemma A.1. *Let $\delta_{jk} = 1$ when $j = k$ and $\delta_{jk} = 0$ when $j \neq k$. For $n = 1, \ldots, N$ and $j = 1, 2, \ldots, K$, we have*

$$\begin{aligned}
\frac{\partial F(\xi_n; \boldsymbol{\theta}_k)}{\partial w_j} &= f(\xi_n; \boldsymbol{\theta}_k) \frac{\partial \xi_n}{\partial w_j}, \\
\frac{\partial F(\xi_n; \boldsymbol{\theta}_k)}{\partial \mu_j} &= f(\xi_n; \boldsymbol{\theta}_k) \left(\frac{\partial \xi_n}{\partial \mu_j} - \delta_{jk} \right), \\
\frac{\partial F(\xi_n; \boldsymbol{\theta}_k)}{\partial \sigma_j} &= f(\xi_n; \boldsymbol{\theta}_k) \left(\frac{\partial \xi_n}{\partial \sigma_j} - \left\{ \frac{\xi_n - \mu_k}{\sigma_k} \right\} \delta_{jk} \right),
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial}{\partial w_j} T\left(\frac{\xi_n - \mu_k}{\sigma_k} \right) &= f(\xi_n; \boldsymbol{\theta}_k) \left(\frac{\xi_n - \mu_k}{\sigma_k} \right) \frac{\partial \xi_i}{\partial w_j}, \\
\frac{\partial}{\partial \mu_j} T\left(\frac{\xi_n - \mu_k}{\sigma_k} \right) &= f(\xi_n; \boldsymbol{\theta}_k) \left(\frac{\xi_n - \mu_k}{\sigma_k} \right) \left(\frac{\partial \xi_n}{\partial \mu_j} - \delta_{jk} \right), \\
\frac{\partial}{\partial \sigma_j} T\left(\frac{\xi_n - \mu_k}{\sigma_k} \right) &= f(\xi_n; \boldsymbol{\theta}_k) \left(\frac{\xi_n - \mu_k}{\sigma_k} \right) \left\{ \frac{\partial \xi_i}{\partial \sigma_j} - \left(\frac{\xi_n - \mu_k}{\sigma_k} \right) \delta_{jk} \right\}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
\frac{\partial \xi_n}{\partial \mu_k} &= \frac{w_k f(\xi_i; \boldsymbol{\theta}_k)}{f(\xi_n; G)}, \\
\frac{\partial \xi_n}{\partial \sigma_k} &= \frac{w_k f(\xi_n; \boldsymbol{\theta}_k)}{f(\xi_i; G)} \left(\frac{\xi_n - \mu_k}{\sigma_k} \right), \\
\frac{\partial \xi_n}{\partial w_k} &= -\frac{F(\xi_n; \boldsymbol{\theta}_k)}{f(\xi_n; G)}.
\end{aligned}$$

Based on this lemma, it is seen that

$$\begin{aligned}\frac{\partial \mathbb{W}_N}{\partial \mu_j} =& 2w_j(\mu_j + \sigma_j\mu_0) - 2w_j \sum_{n=1}^{N} x_{(n)} \Delta F_{nj} \\ & - 2\sum_{n=1}^{N}\sum_{k} w_k\mu_k x_{(n)} \left\{ \frac{\partial F_0(\xi_n;\boldsymbol{\theta}_k)}{\partial \mu_j} - \frac{\partial F_0(\xi_{n-1};\boldsymbol{\theta}_k)}{\partial \mu_j} \right\} \\ & - 2\sum_{n=1}^{N}\sum_{k} w_k\sigma_k x_{(n)} \frac{\partial}{\partial \mu_j} \left\{ T\left(\frac{\xi_n - \mu_k}{\sigma_k}\right) - T\left(\frac{\xi_{n-1} - \mu_k}{\sigma_k}\right) \right\}\end{aligned}$$

with $F_0(\xi_0;\theta_k) = 0$, $F_0(\xi_{N+1};\theta_k) = 1$, $T\big(\frac{\xi_0-\mu_k}{\sigma_k}\big) = 0$, and $T\big(\frac{\xi_{N+1}-\mu_k}{\sigma_k}\big) = \int_{-\infty}^{\infty} t f_0(t)dt$ is a constant that does not depend on any parameters. Substituting the partial derivatives in Lemma A.1, we then get

$$\begin{aligned}\frac{\partial \mathbb{W}_N}{\partial \mu_j} =& 2w_j(\mu_j + \sigma_j\mu_0) - 2w_j \sum_{n=1}^{N} x_{(n)} \Delta F_{nj} \\ & - 2\sum_{n=1}^{N-1} x_{(n)}\xi_n \sum_{k} w_k f(\xi_n;\mu_k,\sigma_k) \left(\frac{\partial \xi_n}{\partial \mu_j} - \delta_{jk}\right) \\ & + 2\sum_{n=1}^{N-1} x_{(n)}\xi_{n-1} \sum_{k} w_k f(\xi_{n-1};\mu_k,\sigma_k) \left(\frac{\partial \xi_{n-1}}{\partial \mu_j} - \delta_{jk}\right) \\ =& 2w_j \left\{ \mu_j + \sigma_j\mu_0 - \sum_{n=1}^{N} x_{(n)} \Delta F_{nj} \right\}\end{aligned}$$

Similarly, we have

$$\begin{aligned}\frac{\partial \mathbb{W}_N}{\partial \sigma_j} &= 2w_j \left\{ \sigma_j(\mu_0^2 + \sigma_0^2) + \mu_j\mu_0 - \sum_{n=1}^{N} x_{(n)} \Delta \mu_{nj} \right\}, \\ \frac{\partial \mathbb{W}_N}{\partial w_k} &= \{\mu_k^2 + \sigma_k^2(\mu_0^2 + \sigma_0^2) + 2\mu_k\sigma_k\mu_0\} - 2\sum_{n=1}^{N-1} \{x_{(n+1)} - x_{(n)}\} \xi_i F(\xi_n;\boldsymbol{\theta}_k) \\ & \quad - 2\left\{ \mu_k \sum_{n=1}^{N} x_{(n)} \Delta F_{nk} + \sigma_k \sum_{n=1}^{N} x_{(n)} \Delta T_{nk} \right\}.\end{aligned}$$

These gradients are used in the BFGS algorithm to compute the MWDE under

location-scale mixtures in Chapter 3.

Compute Quantiles of Mixtures

Computing the quantiles of the mixture distribution $F(\cdot; G)$ for each G is one of the most demanding tasks in the computation of the gradients of $\mathbb{W}_N$ in (A.1). The property stated in the following lemma allows us to develop a bi-section algorithm for computing the quantiles of the mixture.

Lemma A.2. *Let $F(x; G) = \sum_{k=1}^{K} F(x; \mu_k, \sigma_k)$ be a K-component mixture, $\xi(t) = F^{-1}(t; G)$ and $\xi_k(t) = F^{-1}(t; \boldsymbol{\theta}_k)$ respectively the t-quantile of the mixture and its kth subpopulation. For any $t \in (0, 1)$,*

$$\min_k \xi_k(t) \leq \xi(t) \leq \max_k \xi_k(t). \tag{A.3}$$

Proof. Since $F(x; \boldsymbol{\theta})$ has a continuous CDF, we must have $F(\xi_k(t); \boldsymbol{\theta}_k) = t$. By the monotonicity of the CDF $F(\cdot; \boldsymbol{\theta}_k)$, we have

$$F\left(\min_k \xi_k(t); \boldsymbol{\theta}_k\right) \leq F(\xi_k(t); \boldsymbol{\theta}_k) \leq F\left(\max_k \xi_k(t); \boldsymbol{\theta}_k\right).$$

Multiplying by w_k and summing over k lead to

$$F\left(\min_k \xi_k(t); G\right) \leq t \leq F\left(\max_k \xi_k(t); G\right).$$

This implies (A.3) and completes the proof. □

In view of this lemma, we can easily find the quantiles of $F(\cdot; \boldsymbol{\theta}_k)$ to form an interval containing the targeting quantile of $F(\cdot; G)$. We can effectively find $F^{-1}(t; G)$ value through a bi-section algorithm.

Appendix B

Appendix for Chapter 4

The appendix for Chapter 4 is organized as follows. Section B.1 contains all left over technical details and proofs and Section B.2 provides additional details.

B.1 Technical Proofs

Example 4.1: Technical Details

We will show that $\mathbb{D}(\overline{G}^C) < \mathbb{D}(\overline{G})$ where

$$\mathbb{D}(G) = 0.5\mathbb{W}_{D,2}^2(G_1, G) + 0.5\mathbb{W}_{D,2}^2(G_2, G).$$

This result implies that $\overline{G}$ is not a barycenter. We stop short of proving that $\overline{G}^C$ is. The latter task is so tedious that we have it omitted.

Note that all transportation plans from G_1 and G_2 to the presumed barycentre $\overline{G}^C$ have the form

$$\begin{pmatrix} p & 0.4-p \\ 0.4-p & 0.2+p \end{pmatrix} \text{ and } \begin{pmatrix} p & 0.6-p \\ 0.4-p & p \end{pmatrix},$$

respectively, for some p between 0 and 0.4. These two matrices are bivariate probability mass functions with the marginal probability masses $(0.4, 0.6)^\top$ and

$(0.6, 0.4)^\top$ as required. The cost functions may be presented as

$$\begin{pmatrix} c(-1,-1) & c(-1,2/3) \\ c(1,-1) & c(1,2/3) \end{pmatrix} = \begin{pmatrix} 0 & 25/9 \\ 4 & 1/9 \end{pmatrix}.$$

It is clear that $p = 0.4$ gives the optimal plans for transporting G_1 to $\overline{G}^C$ and G_2 to $\overline{G}^C$. With these plans in place, we can see that

$$\mathbb{D}(\overline{G}^C) = 0.5\mathbb{W}^2_{D,2}(G_1, \overline{G}^C) + 0.5\mathbb{W}^2_{D,2}(G_2, \overline{G}^C) = 1/3.$$

In comparison, the optimal transportation plan from G_1 to $\overline{G}$ is to move 0.1 mass from ϕ_1 to ϕ_{-1} with a total cost of $0.1 \times 4 = 0.4$. Hence,

$$\mathbb{D}(\overline{G}) = 0.5\mathbb{W}^2_{D,2}(G_1, \overline{G}) + 0.5\mathbb{W}^2_{D,2}(G_2, \overline{G}) = 0.4 > 1/3.$$

That is, $\overline{G}$ is not a barycenter.

Proof of Theorem 4.1

Let $G^* = \arg\inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}$. Let the mixing weights of any $G \in \mathbb{G}_K$ be $\widetilde{\boldsymbol{w}}(G)$, and let the subpopulations prescribed by G be Φ_k. According to (4.12), we have

$$\widetilde{w}_k(G^*) = \sum_i \pi_{ik}(G^\star),$$

which implies that $\boldsymbol{\pi}(G^*) \in \Pi(\cdot, \widetilde{\boldsymbol{w}}(G^*))$. Since $\boldsymbol{\pi}(G^*) \in \Pi(\boldsymbol{w}, \cdot)$ by (4.7), we also have that $\boldsymbol{\pi}(G^*) \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}}(G^*))$ or it is a valid transportation plan from $\overline{G}$ to G^*. Consequently,

$$\begin{aligned} &\inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\} \\ &\leq \sum_{i,k} \pi_{ik}(G^*) c(\overline{\Phi}_i, \Phi^*_k) = \mathcal{J}_c(G^*) = \inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}, \end{aligned}$$

with the last equality implied by the definition of G^*. This inequality implies that the left-hand side of (4.10) is less than its right-hand side.

Next, we prove that the inequality holds in the other direction. Let $G^\dagger = \inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\}$, the solution to the optimization on the left-hand side

of (4.10). We denote the subpopulations prescribed by $G^\dagger$ as $\Phi^\dagger_\gamma$. Let

$$\boldsymbol{\pi}^\dagger = \arg\inf \left\{ \sum_{i,k} \boldsymbol{\pi}_{ik} c(\overline{\Phi}_i, \Phi^\dagger_k) : \boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}}(G^\dagger)) \right\},$$

which is the optimal transportation plan from $\overline{G}$ to this $G^\dagger$. Because of this, we have

$$\begin{aligned} \inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\} = \mathcal{T}_c(G^\dagger) &= \sum_{i,k} \boldsymbol{\pi}^\dagger_{ik} c(\overline{\Phi}_i, \Phi^\dagger_k) \\ &\geq \mathcal{J}_c(G^\dagger) \geq \inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}. \end{aligned}$$

The last step holds because $\boldsymbol{\pi}^\dagger \in \Pi(\boldsymbol{w}, \cdot)$. This completes the proof.

Proof of Theorem 4.2

(i). Clearly, we have $\mathcal{K}_c(G|G_t) \geq \mathcal{J}_c(G)$ for all G with equality holds at $G = G_t$. Hence,

$$\begin{aligned} \mathcal{J}_c(G_t) &\geq \mathcal{J}_c(G_t) - \{\mathcal{K}_c(G_{t+1}|G_t) - \mathcal{J}_c(G_{t+1})\} \\ &= \mathcal{J}_c(G_{t+1}) - \{\mathcal{K}_c(G_{t+1}|G_t) - \mathcal{J}_c(G_t)\} \\ &\geq \mathcal{J}_c(G_{t+1}) - \{\mathcal{K}_c(G_t|G_t) - \mathcal{J}_c(G_t)\} \\ &= \mathcal{J}_c(G_{t+1}). \end{aligned}$$

This is the property that an Majorization Maximization (MM) algorithm must have. **(ii)**. Suppose $G^{(t)}$ has a convergent subsequence leading to a limit G^*. Let this subsequence be $G^{(t_k)}$. By Helly's selection theorem (Van der Vaart, 2000), there is a subsequence s_k of t_k such that $G^{(s_k+1)}$ has a limit, say G^{**}. These limits, however, could be subprobability distributions. That is, we cannot rule out the possibility that the total probability in the limit is below 1 by Helly's theorem.

This is not the case under the theorem conditions. Let $\Delta > 0$ be large enough such that

$$A_1 = \{\Phi : c(\overline{\Phi}_i, \Phi) \leq \Delta, \text{for all subpopulations } \overline{\Phi}_i \text{ of } \overline{G}\}$$

is not empty. With this Δ, we define

$$A_2 = \{\Phi : c(\overline{\Phi}_i, \Phi) > \Delta, \text{for all subpopulations } \Phi_i \text{ of } \overline{G}\}.$$

Suppose $G^\dagger$ has a subpopulation $\Phi^\dagger$ such that $c(\overline{\Phi}_i, \Phi^\dagger) > \Delta$ for all i. Replacing this subpopulation in $G^\dagger$ by any $\Phi^{\dagger\dagger} \in A_1$ to form $G^{\dagger\dagger}$, we can see that for any t,

$$\mathcal{K}_c(G^\dagger | G^{(t-1)}) > \mathcal{K}_c(G^{\dagger\dagger} | G^{(t-1)}).$$

This result shows that none of the subpopulations of $G^{(t)}$ are members of A_2. Otherwise, $G^{(t)}$ does not minimize $\mathcal{K}_c(G|G^{(t-1)})$ at the tth iteration.

Note that the complement of A_2 is compact by condition (4.15). Consequently, the subpopulations of $G^{(t)}$ are confined to a compact subset. Hence, all limit points of $G^{(t)}$, including both G^* and G^{**}, are proper distributions. By the monotonicity of the iteration:

$$\mathcal{J}_c(G^{(s_{k+1})}) \leq \mathcal{J}_c(G^{(s_k+1)}) \leq \mathcal{J}_c(G^{(s_k)}).$$

Let $k \to \infty$, we get

$$\mathcal{J}_c(G^{**}) = \mathcal{J}_c(G^*). \tag{B.1}$$

By the definition of the MM iteration, we have

$$\mathcal{K}_c(G^{(s_k+1)} | G^{(s_k)}) \leq \mathcal{K}_c(G|G^{(s_k)}).$$

Let $k \to \infty$ and by the continuity of $\mathcal{K}_c(\cdot|\cdot)$, we get

$$\mathcal{K}_c(G^{**}|G^*) \leq \mathcal{K}_c(G|G^*).$$

Hence, G^{**} is a solution to $\min \mathcal{K}_c(G|G^t)$ when $G^{(t)} = G^*$. Namely, we have $\mathcal{K}_c(G^{**}|G^*) = \mathcal{K}_c(G^{(t+1)}|G^*)$. With the help of (B.1), it further implies

$$\mathcal{J}_c(G^{**}) = \mathcal{J}_c(G^{(t+1)}) = \mathcal{J}_c(G^*)$$

when $G^{(t)} = G^*$. This shows that iteration from $G^{(t)} = G^*$ does not make

$\mathcal{J}_c(G^{(t+1)})$ smaller than $\mathcal{J}_c(G^{(t)})$. Hence, G^* is a stationary point of the MM iteration. This is conclusion (ii) and we have completed the proof.

Proof of Theorem 4.3

Recall that the local estimators $\widehat{G}_m, m \in [M]$ are strongly consistent for G^* when the order K of G^* is known. Clearly, this implies that the aggregate estimator $\overline{G} \to G^*$ and $\mathcal{T}_c(\overline{G}, G^*) \to 0$ almost surely. That is, other than a probability 0 event in the probability space Ω on which the random variables are defined, convergence holds. Furthermore, each support point of $\overline{G}$ must converge to one of those of G^*. The total weights of the support points of $\overline{G}$ converging to the same support of G^* must converge to the corresponding weight of G^*. Without loss of generality, assume $\mathcal{T}_c(\overline{G}, G^*) \to 0$ holds at all $\omega \in \Omega$ without a zero-probability exception.

By definition, $\overline{G}^R$ has K support points. We also notice that

$$\mathcal{T}_c(\overline{G}, \overline{G}^R) \leq \mathcal{T}_c(\overline{G}, G^*) \to 0. \tag{B.2}$$

Suppose that $\overline{G}^R$ does not converge to G^* at some $\omega \in \Omega$. One possibility is that the smallest mixing weight of $\overline{G}^R$ (or a subsequence thereof) goes to zero as $N \to \infty$. In this case, $\overline{G}^R$ has $K-1$ or fewer meaningful support points. Since the support points of $\overline{G}$ are in an infinitesimal neighborhood of those of G^*, one of them must be a distance away from any of the support points of $\overline{G}^R$. Therefore, by Condition 4, the transportation cost of this support point is larger than a positive constant not depending on N. The positive transportation cost implies that $\mathcal{T}_c(\overline{G}, \overline{G}^R) \not\to 0$, which contradicts (B.2).

The next possibility is that the smallest mixing weight of $\overline{G}^R$ does not go to zero. In this case, there is a subsequence such that all the mixing weights converge to positive constants. Without loss of generality, all the mixing weights simply converge to positive constants as $N \to \infty$. If there is a subsequence of support points of $\overline{G}^R$ that is at least ϵ-distance away from any of the support points of G^*, then the transportation cost from $\overline{G}$ to this support point will be larger than a positive constant not depending on N. This again leads to a contradiction to (B.2).

The final possibility is that $\overline{G}^R$ (or a subsequence thereof) has a proper limit, say $G^{**} \neq G^*$. If so, $\mathcal{T}_c(\overline{G}, \overline{G}^R) \to \mathcal{T}_c(G^*, G^{**}) \neq 0$, contradicting (B.2).

We have exhausted all the possibilities. Hence, the consistency claim is true.

Proof of Theorem 4.4

We start with a few rate conclusions. Let Φ_{mk} be the kth subpopulation learned at local machine m and w_{mk} be its mixing weight. Note that we do not put a "hat" on them for notation simplicity. According to Lemma 2.1 on the rate of convergence of the PMLE at local machines, these subpopulations can be arranged so that for all $m \in [M]$ and $k \in [K]$, we have

$$\|\Phi_{mk} - \Phi_k^*\| = O_p(N^{-1/2}), \quad \sum_m \lambda_m w_{mk} - w_k^* = O_p(N^{-1/2}).$$

By C5, the first rate conclusion above implies

$$\max\{c(\Phi_{mk}, \Phi_k^*) : k \in [K]\} = O_p(N^{-1}).$$

For each k, let $\widetilde{w}_k = \sum_{m=1}^M \lambda_m w_{mk}$ and $\widetilde{\Phi}_k$ be the local barycentre of Φ_{mk}, $m \in [M]$:

$$\widetilde{\Phi}_k = \arg\min \left\{ \Phi : \sum_{m=1}^M \lambda_m w_{mk} c(\Phi_{mk}, \Phi) \right\}.$$

By the rate conclusions given earlier, we have $\widetilde{w}_k = w_k^* + O_p(N^{-1/2})$ for $k \in [K]$. By C5, we must also have

$$\|\widetilde{\Phi}_k - \Phi_k^*\| = O_p(N^{-1/2})$$

and $\mathcal{T}_c(\overline{G}, \widetilde{G}) = o_p(N^{-1})$. This $\overline{G}^R$ is given by $\widetilde{G}$, then the rate conclusion of the theorem is proved.

Next, we show that the GMR $\overline{G}^R$ is given by $\widetilde{G}$ asymptotically. By theorem conditions, the true subpopulations Φ_k^* are all distinct. Hence, by condition C4, we have

$$\min\{c(\Phi_k^*, \Phi_{k'}^*) : k \neq k' \in [K]\} > 0.$$

Thus, if the subpopulations of $\overline{G}^R$ is not in an $o_p(1)$ neighbourhood of one of Φ_k^* even though everyone of $\overline{G}$ is, the transport cost to this subpopulation from any

subpopopulation of $\overline{G}$ exceeds a positive constant in probability. This contradicts

$$\mathcal{T}_c(\overline{G}, \overline{G}^R) \leq \mathcal{T}_c(\overline{G}, \widetilde{G}) = o_p(N^{-1}). \tag{B.3}$$

This implies, all subpopulations of $\overline{G}^R$ are within $o_p(1)$ neighbourhood of one of Φ_k^*. Denote these subpopulations as $\overline{\Phi}_k^R$ The optimal plan must transport Φ_{mk} to $\overline{\Phi}_k^R$, otherwise the total transport cost exceeds a positive constant in probability which again contradicts (B.3). Since $\overline{G}^R$ minimizes the transport cost, we must have $\overline{\Phi}_k^R = \widetilde{\Phi}_k$, the local barycenter. These conclusions imply that GMR $\overline{G}^R = \widetilde{G}$ with probability approaching to 1. Consequently, the rates of convergence of $\widetilde{w}_k, \widetilde{\Phi}_k$ extend to those of $\overline{G}^R$ and this completes the proof.

KL Divergence Satisfies C5

Let μ_1, Σ_1 and μ_2, Σ_2 be the parameters of Φ_1 and Φ_2. It is known that

$$2D_{\text{KL}}(\Phi_1 \| \Phi_2) = \log \frac{\det(\Sigma_2)}{\det(\Sigma_1)} + \text{tr}(\Sigma_2^{-1}\Sigma_1 - \mathbf{I}_d) + (\mu_2 - \mu_1)^\top \Sigma_2^{-1}(\mu_2 - \mu_1).$$

Assume both Φ_1 and Φ_2 are in a small neighborhood of Φ whose covariance matrix Σ is positive definite. Hence, eigenvalues of Σ_2 are in small neighborhood of these of Σ. Thus, there exists a positive constant A_1 such that the second term in $2D_{\text{KL}}(\Phi_1 \| \Phi_2)$ satisfies

$$A_1^{-1}\|\mu_2 - \mu_1\|^2 \leq (\mu_2 - \mu_1)^\top \Sigma_2^{-1}(\mu_2 - \mu_1) \leq A_1\|\mu_2 - \mu_1\|^2 \tag{B.4}$$

Let $\lambda_1, \ldots, \lambda_d$ be eigenvalues of $\Sigma_2^{-1/2}\Sigma_1\Sigma_2^{-1/2}$. Since both Σ_1 and Σ_2 are in a small neighborhood of Σ, we have $\lambda_1, \ldots, \lambda_d$ all close to 1.

$$\log\{\det(\Sigma_2)/\det(\Sigma_1)\} + \text{tr}(\Sigma_2^{-1}\Sigma_1 - \mathbf{I}_d) = \sum_{j=1}^{d}\{(\lambda_j - 1) - \log\lambda_j\}.$$

Note that $(\lambda - 1) - \log\lambda$ is a convex function with its minimum attained at $\lambda = 1$ at which point its second derivative equals 1. Hence, there exists an $A_2 > 0$ such that

$$A_2^{-1}(\lambda - 1)^2 \leq (\lambda - 1) - \log\lambda \leq A_2(\lambda - 1)^2.$$

We have therefore shown that

$$A_2^{-1}\sum_{j=1}^{d}(\lambda_j-1)^2 \le \log\{\det(\Sigma_2)/\det(\Sigma_1)\}+\mathrm{tr}(\Sigma_2^{-1}\Sigma_1-\mathbf{I}_d) \le A_2\sum_{j=1}^{d}(\lambda_j-1)^2. \tag{B.5}$$

We now connect the bound with Frobenius norm.

For a positive definite matrix Σ, it is easy to see that $\|\Sigma-\mathbf{I}_d\|_F^2=\sum_{j=1}^{d}(\sigma_j-1)^2$, where $\sigma_1,\sigma_2,\ldots,\sigma_d$ are eigenvalues of Σ. Frobenius norm also has sub-multiplicative property

$$\|\Sigma_1\Sigma_2\|_F \le \|\Sigma_1\|_F\|\Sigma_2\|_F.$$

Applying the sub-multiplicative property in our context, we get

$$\begin{aligned}\|\Sigma_1-\Sigma_2\|_F^2 &\le \|\Sigma_2^{1/2}\|_F^4\,\|\Sigma_2^{-1/2}\Sigma_1\Sigma_2^{-1/2}-\mathbf{I}_d\|_F^2\\ &\le A_3\,\|\Sigma_2^{-1/2}\Sigma_1\Sigma_2^{-1/2}-\mathbf{I}_d\|_F^2\\ &= A_3\sum_{j=1}^{d}(\lambda_j-1)^2\end{aligned}$$

for some local positive constant $A_3 > \|\Sigma^{1/2}\|_F^4$, as both matrices are in a small neighborhood of Σ. Similarly, we have

$$\begin{aligned}\sum_{j=1}^{d}(\lambda_j-1)^2 &= \|\Sigma_2^{-1/2}\Sigma_1\Sigma_2^{-1/2}-\mathbf{I}_d\|_F^2\\ &= \|\Sigma_2^{-1/2}\{\Sigma_1-\Sigma_2\}\Sigma_2^{-1/2}\|_F^2\\ &\le \|\Sigma_2^{-1/2}\|_F^4\,\|\Sigma_1-\Sigma_2\|_F^2\\ &\le A_4\|\Sigma_1-\Sigma_2\|_F^2.\end{aligned}$$

for some positive constant A_4. This leads to

$$\|\Sigma_1-\Sigma_2\|_F^2 \ge A_4^{-1}\sum_{j=1}^{d}(\lambda_j-1)^2.$$

Let $A = 2\max\{A_1, A_2, A_3, A_4\}$. Applying (B.4) and (B.5), we have

$$A^{-1}\{\|\mu_1-\mu_2\|^2+\|\Sigma_1-\Sigma_2\|_F^2\} \leq D_{\text{KL}}(\Phi_1\|\Phi_2) \leq A\{\|\mu_1-\mu_2\|^2+\|\Sigma_1-\Sigma_2\|_F^2\}$$

when Φ_1, Φ_2 are in a small neighborhood of Φ, with A being a positive constant depends on Φ. This shows that the KL-divergence has property C5.

B.2 Additional Details

Additional Simulation Results

In this section, we present additional simulation results for $K = 5, 10, 50$ and $d = 10, 50$. All the settings are as in Section 4.5.1. Figures B.1–B.2 show the results for $N = 2^{19}$ and $M = 4$ with various combinations of K and d. The panels in each figure are arranged so that the order of the mixture increases from left to right, and the dimension of the mixture increases from top to bottom.

Comparing panels within the same row in Figure B.1, we note that the performance of all the estimators becomes worse as the order of the mixture increases in terms of W_1 distance. The panels within the same column in Figures B.1 show that all the estimators become worse as the dimension of the mixture increases in terms of both performance measures.

Regardless, our estimator has performance comparable to that of the global estimator. In terms of the misclassification error, for the same degree of overlapping, the superiority of our estimator increases compared to the KL-averaging as the number of components and the dimension increase.

The computational costs of the local estimators are typically low, and this gives our method an added computational advantage. However, this advantage is not guaranteed: see the bottom right panel in Figure B.2, where $d = 50$, $K = 50$, and the degree of overlapping is above 1%. There are many other factors at play. A more skillful implementation may lead to different conclusions on the computational time.

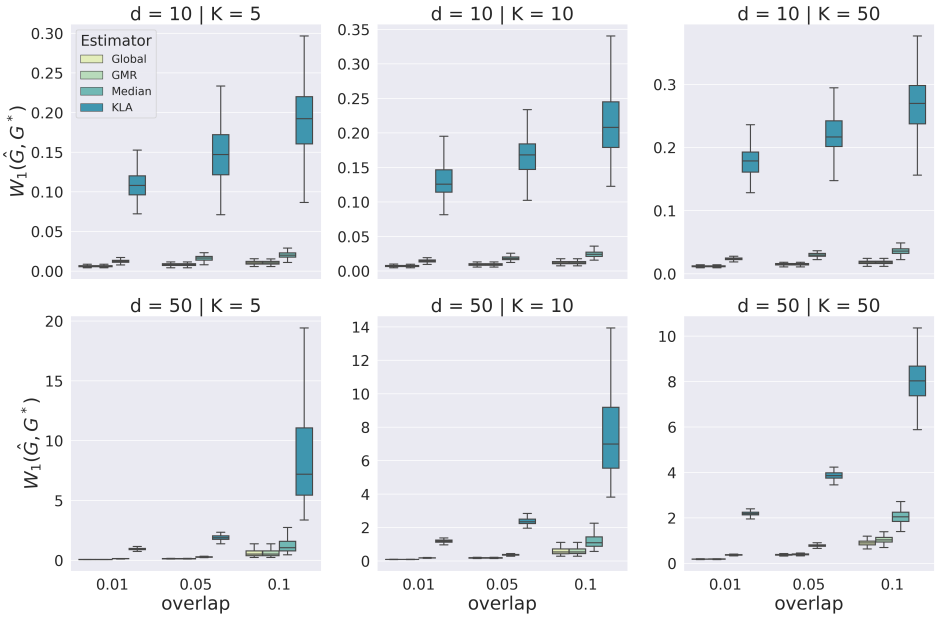


Figure B.1: W_1 distances of estimators for learning mixtures.

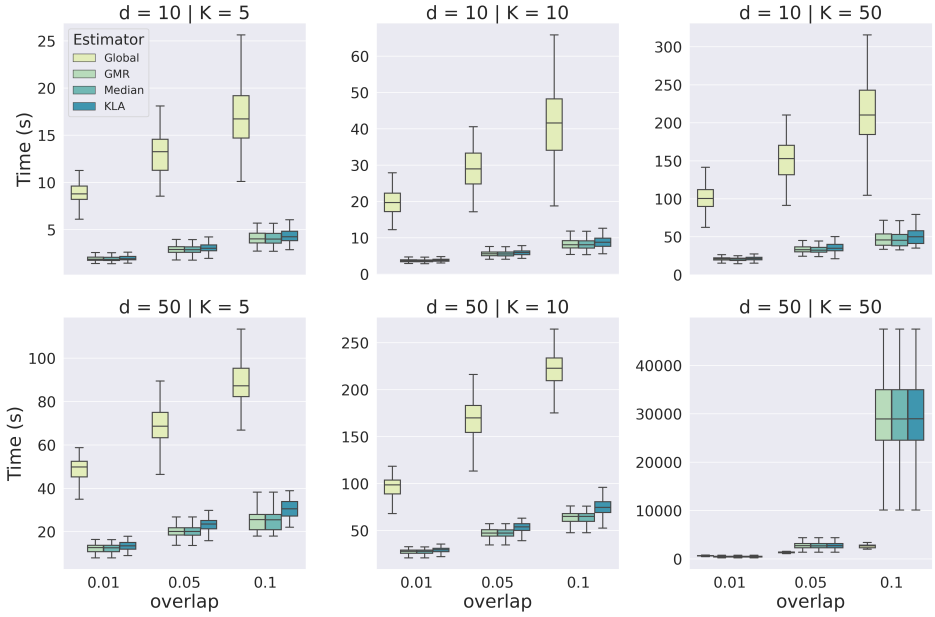


Figure B.2: Computational times for learning mixtures.

Convolutional Neural Network in NIST Example

Deep convolutional neural networks (CNNs) are commonly used to reduce the complex structure of a dataset to informative rectangle data. CNNs effectively perform dimension reduction and classification in an end-to-end fashion (Dara and Tumma, 2018). The final soft-max layer in a CNN can be viewed as fitting a multinomial logistic regression model on the reduced feature space. We use a CNN for dimension reduction in the NIST experiment; its architecture is specified in Table B.1. We implement the CNN in `pytorch 1.5.0` (Paszke et al., 2019) and train it for 10 epochs on the NIST training dataset. We use the SGD optimizer with learning rate 0.01, momentum 0.9, and batch size 64. After the training, we drop the final layer and use the resulting CNN to reduce the dimension of the images in the training and test sets to 50.

Table B.1: Architecture and layer specifications of CNN for dimension reduction in NIST example.

Layer	Layer specification	Activation function
Conv2d	$C_{\text{in}} = 1$, $C_{\text{out}} = 20$, $H = W = 5$	Relu
MaxPool2d	$k = 2$	–
Conv2d	$C_{\text{in}} = 20$, $C_{\text{out}} = 50$, $H = W = 5$	Relu
MaxPool2d	$k = 2$	–
Flatten	–	–
Linear	$H_{\text{in}} = 800$, $H_{\text{out}} = 50$	Relu
Linear	$H_{\text{in}} = 50$, $H_{\text{out}} = 10$	Softmax

Appendix C

Appendix for Chapter 5

C.1 Optimal Transportation Plan with One Marginal Constraint

In this section, we derive the closed-form of the optimal transportion plan when there is only one marginal constraint.

Let $\Pi(\boldsymbol{w}, \cdot) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{N\times M} : \sum_{m=1}^{M} \pi_{nm} = w_n\}$, $\mathcal{H}(\boldsymbol{\pi}) = -\sum_{n,m} \pi_{nm}(\log \pi_{nm} - 1)$, and

$$\boldsymbol{\pi}^{\lambda} = \underset{\boldsymbol{\pi}\in\Pi(\boldsymbol{w},\cdot)}{\arg\inf} \left\{ \sum_{n,m} \pi_{nm} C_{nm} - \lambda \mathcal{H}(\boldsymbol{\pi}) \right\}$$

for some C_{nm} that does not depend on $\boldsymbol{\pi}$. We show in this section that

$$\pi_{nm}^{\lambda} = \begin{cases} w_n \frac{\exp(-C_{nm}/\lambda)}{\sum_{m'} \exp(-C_{nm'}/\lambda)} & \lambda > 0 \\ w_n \frac{\mathbb{1}\{m = \arg\min_{m'} C_{nm}\}}{|\arg\min_{m'} C_{nm'}|} & \lambda = 0 \end{cases} \tag{C.1}$$

and

$$\boldsymbol{\pi}^{0} = \lim_{\lambda \downarrow 0} \boldsymbol{\pi}^{\lambda}.$$

Proof. Let

$$\ell_{\mathbf{C}}(\boldsymbol{\pi}) = \sum_{nm} \pi_{nm} C_{nm} - \lambda \mathcal{H}(\boldsymbol{\pi}). \tag{C.2}$$

We prove the results under the following two cases.

Case I ($\lambda > 0$) The Lagrangian associated with (C.2) is

$$\mathcal{L}(\pi, \xi_1, \cdots, \xi_N) = \ell_{\mathbf{C}}(\pi) - \sum_{n=1}^{N} \xi_n \left\{ \sum_{m=1}^{M} \pi_{nm} - w_n \right\}.$$

Then for $n \in [N]$ and $m \in [M]$, the first order conditions yield

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \pi_{nm}} = C_{nm} - \lambda \log \pi_{nm} - \xi_n = 0, \\ \frac{\partial \mathcal{L}}{\partial \xi_n} = \sum_{m=1}^{M} \pi_{nm} - w_n = 0. \end{cases}$$

Then the optimal transportation plan is given by

$$\pi^{\lambda}_{nm} = w_n \frac{\exp(-C_{nm}/\lambda)}{\sum_{m'} \exp(-C_{nm'}/\lambda)}. \tag{C.3}$$

Case II ($\lambda = 0$) The objective function becomes

$$\ell_C(\boldsymbol{\pi}) = \sum_{nm} C_{nm} \pi_{nm}$$

which is linear in $\boldsymbol{\pi}$ under the constraints that $\sum_m \pi_{nm} = w_n$ for $n \in [N]$. Therefore, by the linearity and the fact that $C_{nm} \geq 0$, it is clear that the objective function is smallest when

$$\pi_{nm} = \begin{cases} w_n & m = \arg\min_{m'} C_{nm'} \\ 0 & \text{otherwise} \end{cases}$$

When there are ties in $\arg\min_{m'} C_{nm'}$, we could evenly split the weight w_n and the optimal transportation plan becomes

$$\pi_{nm} = \begin{cases} w_n / |\arg\min_{m'} C_{nm'}| & m \in \arg\min_{m'} C_{nm'} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we show that the optimal transportation plan is (C.1) when there is only one marginal constraint on the transportation plan.

We now show that $\boldsymbol{\pi}^0 = \lim_{\lambda \downarrow 0} \boldsymbol{\pi}^\lambda$. According to (C.3), we have

$$\begin{aligned}\lim_{\lambda \downarrow 0} \pi_{nm}^\lambda &= \lim_{\lambda \downarrow 0} w_n \frac{\exp(-C_{nm}/\lambda)}{\sum_{m'} \exp(-C_{nm'}/\lambda)} \\ &= \lim_{\lambda \downarrow 0} \frac{w_n}{\sum_{m'} \exp\{-(C_{nm'} - C_{nm})/\lambda\}}\end{aligned}$$

We discuss the limit under the following two cases:

- Let $A_n = \arg\min_{m'} C_{nm'}$, then When $m \in A_n$, then there are $|A_n|$ terms in the denominator equals 1 and $\exp\{-(C_{nm^\ddagger} - C_{nm})/\lambda\} \to 0$ as $\lambda \downarrow 0$ for any $m^\ddagger \notin A_n$. Therefore, in this case, we have

$$\lim_{\lambda \downarrow 0} \pi_{nm}^\lambda = w_n / |\arg\min_{m'} C_{nm'}|.$$

- When there exists an m^* so that $C_{nm^*} < C_{nm}$, then $\exp\{-(C_{nm^*} - C_{nm})/\lambda\} \to \infty$ as $\lambda \downarrow 0$, hence

$$\lim_{\lambda \downarrow 0} \pi_{nm}^\lambda \to 0.$$

In conclusion, we then have

$$\lim_{\lambda \downarrow 0} \pi_{nm}^\lambda = w_n \frac{\mathbb{1}\{m \in \arg\min_{m'} C_{nm'}\}}{|\arg\min_{m'} C_{nm'}|} = \pi_{nm}^0.$$

□

C.2 Gaussian Baycentre under KL Divergence

We show the conclusion of Example 2.4 in this section. Let $\{\Phi_n(x) = \Phi(x; \mu_n, \Sigma_n) : n \in [N]\}$ be a set of Gaussian distributions. The Kullback-Leibler (KL) barycentre of $\{\Phi_n : n \in [N]\}$ with weights $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)^\top$, when confined on the space of Gaussian distributions, has its mean

$$\overline{\mu} = \left\{ \sum_{n=1}^N \lambda_n \right\}^{-1} \sum_{n=1}^N \lambda_n \mu_n$$

and the covariance

$$\overline{\Sigma} = \left\{ \sum_{n=1}^{N} \lambda_n \right\}^{-1} \sum_{n=1}^{N} \lambda_n \{ \Sigma_n + (\mu_n - \overline{\mu})(\mu_n - \overline{\mu})^\top \}.$$

Compare the barycentre with the moment matching in (5.10), we can see the barycentre of subpopulations belonging to the M cluster is the same as that from moment matching.

Proof. With KL-divergence, the barycentre confined on the space of Gaussians is a Gaussian with its mean and covariance minimize the function

$$\begin{aligned} L(\mu, \Sigma) =& \sum_{n=1}^{N} \lambda_n D_{\mathrm{KL}}(\Phi_n \| \Phi) \\ =& \frac{1}{2} \sum_n \lambda_n \left\{ \log \det(\Sigma) + \mathrm{tr}(\Sigma^{-1} \Sigma_n) \right\} \\ &+ \frac{1}{2} \sum_n \lambda_n (\mu - \mu_n)^\top \Sigma^{-1} (\mu - \mu_n) + C \end{aligned}$$

for some constant C. We now use the following linear algebra formulas

$$\frac{\partial \log \det(\Sigma)}{\partial \Sigma} = (\Sigma^{-1})^\top = (\Sigma^\top)^{-1},$$

$$\frac{\partial \mathrm{tr}(A \Sigma^{-1} B)}{\partial \Sigma} = -(\Sigma^{-1} B A \Sigma^{-1})^\top,$$

and

$$\frac{\partial}{\partial x} (x - \mu)^\top \Sigma^{-1} (x - \mu) = 2 \Sigma^{-1} (x - \mu)$$

to work out partial derivatives of L with respect to μ and Σ. They are given by

$$\begin{aligned} \frac{\partial L}{\partial \mu} =& 2 \sum_n \lambda_n \Sigma^{-1} (\mu - \mu_n), \\ \frac{\partial L}{\partial \Sigma} =& \Sigma^{-1} - \Sigma^{-1} \sum_n \lambda_n \left\{ \Sigma_n + (\mu - \mu_n)(\mu - \mu_n)^\top \right\} \Sigma^{-1}. \end{aligned}$$

Setting both partial derivatives to $\mathbf{0}$, we obtain

$$\overline{\mu} = \left\{ \sum_n \lambda_n \right\}^{-1} \sum_{n=1}^{N} \lambda_n \mu_n$$

and the covariance

$$\overline{\Sigma} = \left\{ \sum_n \lambda_n \right\}^{-1} \sum_{n=1}^{N} \lambda_n \{ \Sigma_n + (\mu_n - \overline{\mu})(\mu_n - \overline{\mu})^\top \}.$$

This completes the proof. □

C.3 Connection With Optimization Based Algorithms

In this section, we provide the derivation details for establishing the connection of existing optimization based GMR approach and our proposed approach in Section 5.3.2.

We first show that when the cost function is Integrated Squared Error (ISE) and KL divergence, our proposed approach is the same as the existing optimization based approach under the special case when $M = 1$. When $M > 1$, we show that the composite transportation divergence is an upper bound of the divergence between two mixtures when the cost function satisfies the "convexity".

Proof for Equation (5.27)

Proof. We give the proof under the following two situations.
ISE When $c(\cdot, \cdot) = D_{\text{ISE}}(\cdot, \cdot)$, the objective function on the LHS of (5.27) is

$$\sum_{n=1}^{N} w_n D_{\text{ISE}}(\Phi_n, \widetilde{\Phi}) = \sum_{n=1}^{N} w_n \left\{ \int \phi_n^2(x)\, dx + \int \widetilde{\phi}^2(x)\, dx - 2 \int \phi_n(x) \widetilde{\phi}(x)\, dx \right\}$$

The RHS of (5.27) is

$$
\begin{aligned}
& D_{\mathrm{ISE}}\left(\sum_{n=1}^{N} w_n \Phi_n, \widetilde{\Phi}\right) \\
=& \int \left\{\sum_{n=1}^{N} w_n \phi_n(x) - \widetilde{\phi}(x)\right\}^2 dx \\
=& \int \left\{\sum_{n=1}^{N} w_n \phi_n(x)\right\}^2 dx + \int \widetilde{\phi}^2(x)\, dx - 2\sum_{n} w_n \int \phi_n(x)\widetilde{\phi}(x)\, dx \\
=& C + \sum_{n=1}^{N} w_n D_{\mathrm{ISE}}(\Phi_n, \widetilde{\Phi})
\end{aligned}
$$

where C is some constant that does not depend on $\widetilde{\Phi}$. This relationship implies that (5.27) holds when the cost function is the ISE.

KL divergence

$$
\begin{aligned}
D_{\mathrm{KL}}\left(\sum_{n=1}^{N} w_m \Phi_n \| \widetilde{\Phi}\right) =& \int \left\{\sum_{n=1}^{N} w_n \phi_n(x)\right\} \log\left\{\frac{\sum_n w_n \phi_n(x)}{\widetilde{\phi}(x)}\right\} dx \\
=& C_1 - \sum_{n} w_n \int \phi_n(x) \log \widetilde{\phi}(x)\, dx \\
=& C_2 + \sum_{n} w_n \int \phi_n(x) \log \frac{\phi_n(x)}{\widetilde{\phi}(x)}\, dx \\
=& C_2 + \sum_{n} w_n D_{\mathrm{KL}}(\Phi_n \| \widetilde{\Phi})
\end{aligned}
$$

where C_1 and C_2 are constants not dependent on $\widetilde{\Phi}$. This relationship implies that (5.27) holds when the cost function is the KL divergence. □

Proof for Theorem 5.2

Proof. Let $\boldsymbol{\pi} \in \Pi(\boldsymbol{w}, \widetilde{\boldsymbol{w}})$ be a transportation plan. We can always write $\Phi(\cdot; G) = \sum_n w_n \Phi_n = \sum_n \sum_m \pi_{nm} \Phi_n$. Similarly $\Phi(\cdot; \widetilde{G}) = \sum_{n,m} \pi_{nm} \widetilde{\Phi}_m$. Therefore,

$$\begin{aligned} c(\Phi(\cdot; G), \Phi(\cdot; \widetilde{G})) &= c\left(\sum_n w_n \Phi_n, \sum_m \widetilde{w}_m \widetilde{\Phi}_m\right) \\ &= c\left(\sum_{n,m} \pi_{nm} \Phi_n, \sum_{n,m} \pi_{nm} \widetilde{\Phi}_m\right) \\ &\leq \sum_{n,m} \pi_{nm} c(\Phi_n, \widetilde{\Phi}_m) \end{aligned}$$

The last inequality holds because of the "convexity" property of the cost function. That is for any $\alpha \in (0, 1)$, and component distributions F_1, F_2, Φ_1, and Φ_2, we have

$$c(\alpha F_1 + (1-\alpha) F_2, \alpha \Phi_1 + (1-\alpha)\Phi_2) \leq \alpha c(F_1, \Phi_1) + (1-\alpha) c(F_2, \Phi_2).$$

Since this inequality holds for any transportation plan $\boldsymbol{\pi}$, therefore taking the infimum with respect to $\boldsymbol{\pi}$ on the right hand side, we then have

$$c(\Phi(\cdot; G), \Phi(\cdot; \widetilde{G})) \leq \mathcal{T}_{c,0}(\Phi(\cdot; G), \Phi(\cdot; \widetilde{G}))$$

which completes the proof. □