NONPARAMETRIC MIXTURE MODELING ON CONSTRAINED SPACES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Putu Ayu G. Sudyanti

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Vinayak Rao, Chair

Department of Statistics, Purdue University

Dr. Hyonho Chun

Department of Mathematics and Statistics, Boston University

Dr. Bruce Craig

Department of Statistics, Purdue University

Dr. Anindya Bhadra

Department of Statistics, Purdue University

Approved by:

Dr. Jun Xie

Graduate Chair, Department of Statistics, Purdue University

To my parents, Komang Dharmawan and Nyoman Mandali, and my late grandparents.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my academic advisor, Dr. Vinayak Rao, for his guidance, feedback, and immense knowledge in the four years that I have worked with him. He has shown me continued support, patience, and kindness while modeling to me how to be a good researcher. I could not have imagined having a better PhD advisor. I would like to thank my committee members, Dr. Hyonho Chun for providing me with the tumor datasets to play with, Dr. Bruce Craig, and Dr. Anindya Bhadra for their thoughtful comments and suggestions.

I have been very fortunate to have been funded by the Walther Cancer Foundation as a Research Assistant at the Purdue Center for Cancer Research during my PhD studies. Special thanks to Dr. Nadia Atallah for being a great manager, mentor, and friend, who has shown me the value of persistance and hard work. Thank you to Prof. Ratliff, Dr. Sagar Utturkar, and all my collaborators at PCCR. I really have learned so much while working there. I would also like to thank Dr. Arling from the Nursing Department whom I briefly had the chance to work with and the Fulbright Program for giving me the opportunity to pursue my higher education in the US.

I have met a lot of wonderful people during my time at Purdue. Thank you to my cohort friends who have made the first few years more enjoyable: Kara Keller, Zach Haas, Eric Gerber, Tracy Gonzalez, Barret Schloerke, Rongrong Zhang, Sophie Sun, Yixi Xu, Deborah Chen, and Will Eagan. To my fellow academic siblings with whom I have enjoyed many intellectual discussions: Jiasen Yang, Boqian Zhang, Qi Wang, Hanxi Sun, Bingjing Tang, and Guillherme Gomes. To other friends in the department whom I have grown close to especially Hakeem, Daniel Cardona, AMT, Tim Keaton, and Yumin Zhang. Thank you also to the staffs in the department: Doug, Mary, Patti, and Holly.

To my family away from home, the Indonesian graduate students community, Pam and Dave Krismartanto, Maya Fitryanti, Melati Putri, Utami Irawati, Priskilla Siahaya, and many others. Thank you for making my time at Purdue more pleasant through shared meals and conversations.

Finally, I would not be where I am without the love and support of my family. Thank you to my father who has been my biggest role model, my mother for showing me true strength, my brother who has given me many words of encouragement, my sister-in-law and nieces for making my days brighter with each video call. I would also like to thank Margot and Jan for their kind support over the past years. Lastly, to Emery, thank you for being my best friend throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

MoTG	Mixtures of truncated Gaussians
TMoG	Truncated mixtures of Gaussians
DP	Dirichlet process
DPMM	Dirichlet process mixture models
MCMC	Markov chain Monte Carlo
SNV	Single nucleotide variant
NIW	Normal-Inverse-Wishart
ESS	Effective sample size
CRP	Chinese Restaurant Process

ABSTRACT

Sudyanti, Putu Ayu G. Ph.D., Purdue University, August 2019. Nonparametric mixture modeling on constrained spaces. Major Professor: Vinayak Rao.

Mixture modeling is a classical unsupervised learning method with applications to clustering and density estimation. This dissertation studies two challenges in modeling data with mixture models. The first part addresses problems that arise when modeling observations lying on constrained spaces, such as the boundaries of a city or a landmass. It is often desirable to model such data through the use of mixture models, especially nonparametric mixture models. Specifying the component distributions and evaluating normalization constants raise modeling and computational challenges. In particular, the likelihood forms an intractable quantity, and Bayesian inference over the parameters of these models results in posterior distributions that are doubly-intractable. We address this problem via a model based on rejection sampling and an algorithm based on data augmentation. Our approach is to specify such models as restrictions of standard, unconstrained distributions to the constraint set, with measurements from the model simulated by a rejection sampling algorithm. Posterior inference proceeds by Markov chain Monte Carlo, first imputing the rejected samples given mixture parameters and then resampling parameters given all samples. We study two modeling approaches: mixtures of truncated Gaussians and truncated mixtures of Gaussians, along with Markov chain Monte Carlo sampling algorithms for both. We also discuss variations of the models, as well as approximations to improve mixing, reduce computational cost, and lower variance.

The second part of this dissertation explores the application of mixture models to estimate contamination rates in matched tumor and normal samples. Bulk sequencing of tumor samples are prone to contaminations from normal cells, which lead to difficulties and inaccuracies in determining the mutational landscape of the cancer genome. In such instances, a matched normal sample from the same patient can be used to act as a control for germline mutations. Probabilistic models are popularly used in this context due to their flexibility. We propose a hierarchical Bayesian model to denoise the contamination in such data and detect somatic mutations in tumor cell populations. We explore the use of a Dirichlet prior on the contamination level and extend this to a framework of Dirichlet processes. We discuss MCMC schemes to sample from the joint posterior distribution and evaluate its performance on both synthetic experiments and publicly available data.

1. INTRODUCTION

Nonparametric or infinite mixture models (Lo, 1984; Escobar and West, 1995; Rasmussen, 2000) are powerful modeling tools that are used to cluster observations or estimate complex densities. This dissertation addresses two challenges in modeling multimodal data using nonparametric mixture models: the first involves correctly estimating the density of observations on a constrained set; and the second deals with estimating contamination rates and clustering mutations in tumor samples. The motivations and our proposed methods to address each of the challenges are described in this chapter.

1.1 Density estimation for data on constrained spaces

Observations are often bounded within a specific domain. One example of such data is crime data, where measurements are restricted to within the complex boundaries of a geographical entity, either because none exist outside (due to topographical features like water bodies) or because measurements belong to another city/state/country. Another example is flow cytometry, where measurements with component-values outside some range (e.g. 0 to 1024) are discarded. Other instances include single cell RNA-sequencing data (Cao et al., 2017) (where count-measurements below some threshold does not exist), operational risk modeling (Luo et al., 2009) (where loss data only above a threshold are provided), stock price data (Aban et al., 2006), mortality (Alai et al., 2013), survival (Cain et al., 2011), capture-recapture data (Manning and Goldberg, 2010), where certain outcomes are truncated (Mandel, 2007), animal movement data (Patterson et al., 2008), and climate data (Easterling et al., 2000).

In all of the above cases, it is important to accurately account for the boundaries of the constraint set to avoid biases from boundary effects incorrectly interacting with smoothness assumptions inherent in typical probability models. Doing so, however, raises computational challenges due to the need to evaluate integrals over complicated subsets of a Euclidean space. This problem is exacerbated when the data exhibits rich multimodal and correlation structure, a common situation that requires mixture (and sometimes nonparametric mixture) models.

Edge effects can sometimes be avoided by changing the parametric family used for the mixture components. For example, in Kottas and Sansó (2007) and Matechou et al. (2017), the authors used nonparametric mixtures of beta or gamma distributions to represent the observed data. However, such models cannot easily model correlation structure in multi-dimensional data, something that is natural to Gaussian mixture models. Another approach is to transform the data to be unconstrained, and then model the transformed data with a mixture of Gaussians. This can suffer from edge-effects that are not easy to characterize. Importantly, both approaches are also not applicable to more complex constraint sets.

We propose to treat observations lying on the constrained space as the outcome of a rejection-sampling algorithm (Robert and Casella, 2005), with a proposal distribution q defined on the simpler unconstrained space and with observations falling outside the constraint set discarded. Following Rao et al. (2016); Beskos et al. (2006), we carry out inference over the distribution q by imputing the rejected samples. Implicit in q is all information about the original distribution of interest. Working directly with the unconstrained q and imputing the rejected proposals allows us to use standard Bayesian modeling and computational techniques (such as nonparametric Bayesian models like Dirichlet process mixture models (Lo, 1984; Escobar and West, 1995), and associated Gibbs samplers based on the Chinese restaurant process (Neal, 2000) and the stick-breaking process (Ishwaran and James, 2001)).

In an application example in Rao et al. (2016), the authors briefly considered a setting that we call *truncated mixtures of Gaussians*. We study this in more detail, and also consider another natural approach: *mixtures of truncated Gaussians*. For both models, we describe exact Markov chain Monte Carlo (MCMC) schemes to impute the rejected proposals, allowing inference over cluster assignments, cluster parameters, and cluster weights. In our experiments, we observe that naively implementing this can result in poor MCMC mixing. To speed up computations, improve mixing, and reduce MCMC estimation variance, we also propose and study modifications of the original models and the associated MCMC sampling algorithms. The main results from Chapters 3-6 can be found in the manuscript Sudyanti and Rao (2018). The resulting algorithm significantly ourperforms the original algorithms, especially in higher-dimensional settings.

1.2 Clustering mutations and estimating contaminations in tumor samples

The second part of this dissertation examines a different type of problem involving mixture models on constrained spaces; specifically, DNA sequencing data of tumor samples. Somatic mutations are DNA alterations that are unique to the tumor tissue which drives its growth and proliferation. Correctly identifying these variants is essential to drug discoveries and targeted treatments of cancer patients. However, bulk DNA sequencing of tumor samples is often contaminated by normal samples, leading to difficulties and inaccuracies in the detection of somatic mutations. While DNA sequencing of a single cell can be used, the technology for this technique is expensive, prone to high technical variations, and lacks the sequencing depths found in bulk cells DNA sequencing technologies (Yadav and De, 2014).

The current standard approaches of estimating contamination in a DNA sequencing of tumor samples use external databases to improve the accuracy of the estimation and make them dependable to the comprehensiveness of outside resources. Several methods proposed the use of a matched normal and tumor sample to control for inherited mutations (Bergmann et al., 2016; Su et al., 2012; Larson and Fridley, 2013). However, these methods do not concurrently provide detection of somatic mutations and assume uniform contamination levels along the genome.

For this problem, we developed two novel hierarchical Bayesian models to estimate contamination levels along the genome and accurately detect somatic mutations in patients' matched normal and tumor samples. We need mixture models to flexibly cluster contamination levels, allowing for shared information along the genome. Due to the label-switching problem commonly seen in mixture models, we placed an informative prior over the contamination rate to prevent this problem. In this way, we constrain the parameter space to allow for meaningful inference over contamination level as well as mutations.

The first model is the simpler model with a Dirichlet prior on both the contamination parameter and the joint genotypes. We extend this model to a framework of Dirichlet processes, where clustering structure of the contaminations can be shared across chromosomes. Both our models allow for nucleotide-specific contamination, as well as genome-wide contamination, making them more flexible than existing methods. We employ a Markov chain Monte Carlo algorithm to sample from the joint posterior distribution of the contamination rate and the binomial probabilities. We evaluate our models on synthetic and real data.

1.3 Dissertation organization

The structure of this dissertation is summarized as follows:

- Chapter 2 provides a review of mixture models, both in the finite and infinite case, as well as MCMC methods to sample from the posterior distribution of the parameters in the model. It describes the main Gibbs sampling method which we extend and implement in this dissertation.

- Chapter 3 starts with a review of the rejection sampling algorithm, followed by a description of how this algorithm is used on complex spatial domains. We then introduce our two models: *truncated mixture of Gaussians* and *mixture of truncated Gaussians.*

- Chapter 4 starts with a description of the existing data augmentation approach proposed by Rao et al. (2016), followed by the associated MCMC algorithms of each model within the framework of data augmentation. This chapter also describes and justifies our extension to existing MCMC algorithms which we call the *thresholded sampler*. This new sampler provides improvement in terms of mixing and efficiency.

- Chapter 5 provides evaluations of our approach to various synthetic datasets. A detailed comparison of the two models is presented.

- Chapter 6 presents the result of the application of our models and algorithms to real-world data sets.

- Chapter 7 provides sensitivity analysis of our methods to different hyperparameter settings, as well as training set size. A further modification to the algorithm is also explored here.

- Chapter 8 describes our approaches to the tumor contamination problem. The models and algorithms are presented here, as well as identifiability issues.

- Chapter 9 presents performance results of the models and algorithms of Chapter 9 to various data, both synthetic as well as real.

- Chapter 10 ends this dissertation with a summary and potential future research directions.

2. OVERVIEW OF MIXTURE MODELS

2.1 Introduction

Mixture models are a class of latent variable models widely used to represent data with several different sub-populations, allowing flexibility in modeling complex densities. There are two main applications of mixture models: density estimation and clustering. Example applications can be found in domains such as bioinformatics (Ji et al., 2005; McNicholas and Murphy, 2010; Si Quang et al., 2008; Taslim et al., 2011), astronomy (DeMars and Jah, 2013; Hao et al., 2009), social sciences (Newman and Leicht, 2007; Bauer, 2007; Gormley and Murphy, 2008), neuroscience (Nord et al., 2017; Górriz et al., 2009), text mining (Blei and Lafferty, 2009; Steyvers and Griffiths, 2007), pattern recognition (Permuter et al., 2006; Li et al., 2013; Lagrange et al., 2017), finance (Rombouts and Stentoft, 2015; Kalkbrener and Packham, 2015), and many others.

Many of the contributions in this dissertation use mixture models as their conceptual and theoretical foundation. The models and algorithms developed in later sections, however, are motivated by real-world data that traditional mixture models are unable to accurately describe. In particular, we focus on improving density estimation of data lying within often complex boundaries. Examples include biological data with known truncation values as well as spatial data with geographical constraints (Figure 3.1 shows crime in Chicago). In this chapter, we review the foundational literature of finite and infinite mixture models as well as associated inference procedures in the Bayesian context. This will serve as an overview and reference of key concepts used in later chapters. Readers who are interested in more in-depth explanations are directed to McLachlan et al. (2019) for finite mixture models and to Rasmussen (2000) and Neal (2000) for infinite mixture models.

2.2 Finite Mixture Models

We begin by reviewing mixture models with a finite number of components. Let $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ be a random sample of size n, where X_i is a p-dimensional random vector. In a mixture model, X_i is assumed to come from one of K components, with $c_i \in \{1, \ldots, K\}$ representing the discrete latent cluster for that observation, and where each component belongs to some parametric family of probability distributions with its own unknown vector of parameters given by θ_k. Let $\boldsymbol{\theta}$ be the collection of all component parameters. For a given cluster k, the likelihood of an observation X_i is $p(X_i|c_i = k, \boldsymbol{\theta}) = p_k(X_i|\theta_k)$, where p_k is the k^{th} component's distribution. We write π_k for the mixing weight of the k^{th} component; this is a probability vector satisfying $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$. Then the joint probability of observation X_i and cluster c_i is given by

$$p(X_i, c_i = k) = \pi_k p_k(X_i|\theta_k), \tag{2.1}$$

and the marginal probability of observation X_i is given by

$$p(X_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(X_i|\theta_k). \tag{2.2}$$

For Gaussian mixture models, p_k is the Gaussian density, and θ_k would be the mean and covariance of the k^{th} Gaussian distribution; we write $\theta_k = \{\mu_k, \Sigma_k\}$ and $p(X_i|\theta_k) = \mathcal{N}(X_i|\mu_k, \Sigma_k)$. In a Bayesian setting, a prior is placed over the components' weights $\boldsymbol{\pi}$, as well as over the parameter θ_k. For the prior over $\boldsymbol{\pi}$, it is common to use the conjugate Dirichlet distribution. For the prior over the component parameters θ_k, we place a Normal-Inverse-Wishart (NIW) distribution. Overall, a data point X_i is drawn by first choosing $c_i \in \{1, \ldots, K\}$ from a multinomial distribution with weights given by $\boldsymbol{\pi}$ and then sampling from the coresponding Gaussian component

$\mathcal{N}(\mu_{c_i}, \Sigma_{c_i})$. This process is summarized by Equation (2.3) and is illustrated as a graphical model in Figure 2.1.

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \\
\theta_k &\sim \text{NIW}(\theta_0) \\
c_i|\boldsymbol{\pi} &\sim \text{Multinomial}(\boldsymbol{\pi}) \\
X_i|c_i, \boldsymbol{\theta} &\sim \mathcal{N}(X_i|\mu_{c_i}, \Sigma_{c_i})
\end{aligned}
\tag{2.3}
$$

In this model, $\theta_0 = \{\mu_0, \lambda, \Phi, \nu\}$, the hyperparameters for the Normal-Inverse-Wishart prior. Notationally, we write $\boldsymbol{X}$ as the collection of observed data, $\boldsymbol{c}$ as the collection of all class indicator variables, and $\boldsymbol{\theta} = \{\mu_k, \Sigma_k\}_{k=1}^{K}$ as the collection of all K component parameters. The complete joint distribution of the above Gaussian mixture model is as follows:

$$
\begin{aligned}
P(\boldsymbol{X}, \boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\pi}|\alpha, \theta_0) &= \left(\prod_{i=1}^{N} P(X_i|c_i, \theta_{c_i}) P(c_i|\boldsymbol{\pi})\right) \left(\prod_{j=1}^{K} P(\theta_j|\theta_0)\right) P(\boldsymbol{\pi}|\alpha) \qquad (2.4) \\
&= \prod_{i=1}^{N} \mathcal{N}(X_i|\mu_{c_i}, \Sigma_{c_i}) \text{Mult}(c_i|\boldsymbol{\pi}) \prod_{j=1}^{K} \text{NIW}(\{\mu_j, \Sigma_j\}|\theta_0) \text{Dir}(\boldsymbol{\pi}|\alpha)
\end{aligned}
$$

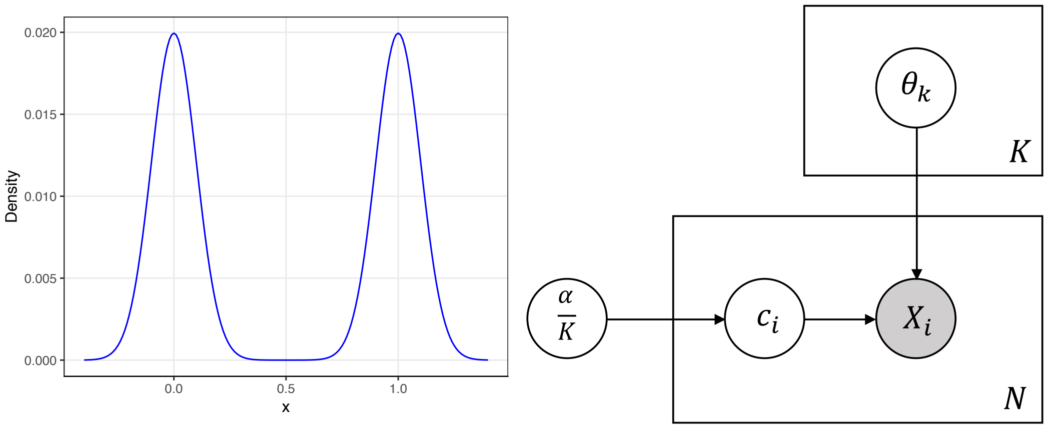


Figure 2.1.: The density of a Gaussian mixture model in 1D with 2 components (left) and the corresponding directed graphical model for N data points with $K = 2$ from this density (right).

2.3 Inference for Finite Mixture Models

The most commonly used approach to estimate the parameters $(\boldsymbol{\pi}, \boldsymbol{\theta})$ in a finite mixture model is through maximum likelihood estimation via the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). This algorithm employs an iterative algorithm to find the solution to the maximum likelihood problem. It alternates between two update steps: the expectation step uses the current parameter values to evaluate the posterior probabilities of the latent class and the maximization step estimates new values for the parameters. However, like other ML-based algorithms, the EM provides a point estimate of the parameters and relies on asymptotic properties to describe the uncertainties. If we are interested in learning the full distribution of the estimated parameters and maintaining uncertainty given prior beliefs and observed data, then we can take the Bayesian approach to perform inference on the parameters of the model. Now, the counterpart to EM algorithms are Markov chain Monte Carlo (MCMC) sampling algorithms (Gamerman and Lopes, 2006; Gilks et al., 1995; Brooks et al., 2011). Specifically, we use an MCMC technique called the Gibbs sampler (Casella and George, 1992; Smith and Roberts, 1993), which generates samples from the posterior probability distribution of the parameters resulting from the priors and observations. Each step of the Gibbs sampler draws from the conditional distribution of a variable (or block of variables), with the remaining variable fixed to its current value. We briefly describe Gibbs sampling for finite mixture models here as it is more closely related to the contributions of this work.

Recall the probabilistic model of a finite Gaussian mixture model with conjugate priors from Equation (2.3). In order to use the Gibbs sampling procedure to perform inference, we must first determine the posterior distribution of all the unknown variables of the model. Applying Bayes's rule and conditioning on the observed data, we see that the posterior is proportional to the joint distribution given by Equation (2.4). The conditional distribution of the component indicators $\boldsymbol{c}$, the component parameters $\boldsymbol{\theta}$, and the weight vector $\boldsymbol{\pi}$ can then be derived from this. Let $\boldsymbol{\pi}^{t-1}$, $\boldsymbol{\theta}^{t-1}$, and $\boldsymbol{c}^{t-1}$ be the value of the parameters for the current iteration of the Gibbs sampler. Details of the update steps for the next iteration, t, are given as follows:

To update $\boldsymbol{c}$: draw new cluster assignments for every observation i by computing the probabilities of choosing a cluster k for all $k = 1, \ldots, K$. The conditional probabilities for c_i are given by

$$P(c_i^t = k | \boldsymbol{c}^{\neg \boldsymbol{i}}, \boldsymbol{X}, \boldsymbol{\pi}^{t-1}, \boldsymbol{\theta}^{t-1}, \alpha, \theta_0) \propto \pi_k^{t-1} \mathcal{N}(X_i | \theta_k^{t-1}), \tag{2.5}$$

where $\boldsymbol{c}^{\neg \boldsymbol{i}}$ is the collection of cluster indicators without observation i.

To update $\boldsymbol{\pi}$: compute the number of observations assigned to each cluster and use these to update the parameters of the distribution over the mixing proportion of the model. Write $n_k^t = \sum_{i=1}^{N} \mathbb{1}\{c_i^t = k\}$ for the number of observations assigned to component k at iteration t. Then,

$$P(\boldsymbol{\pi}^t | \boldsymbol{X}, \boldsymbol{c}^t, \boldsymbol{\theta}^{t-1}, \alpha, \theta_0) \propto \mathrm{Dir}(n_1^t + \alpha/K, \ldots, n_K^t + \alpha/K). \tag{2.6}$$

To update $\boldsymbol{\theta}$: estimate the new mean for component k by sampling from (2.7) for all $k = 1, \ldots, K$. Equation (2.7) suggests that the conditional posterior probability of cluster k, θ_k, only depends on observations that are assigned to k.

$$P(\theta_k^t | \boldsymbol{X}, \boldsymbol{c}^t, \alpha, \theta_0) \propto \mathrm{NIW}(\theta_k | \theta_0) \prod_{\{i:\, c_i = k\}} \mathcal{N}(X_i | \theta_k) \tag{2.7}$$

Since NIW is the conjugate of the multivariate Gaussian distribution, the conditional distribution of the parameter update of θ_k still belongs to NIW.

Each iteration of the Gibbs sampler alternates between sampling $\boldsymbol{\theta}$, $\boldsymbol{c}$, and $\boldsymbol{\pi}$ from their conditional distribution given in equations (2.7), (2.5), and (2.6), respectively. The process is repeated until the required number of samples have been generated. Upon convergence of this MCMC procedure, the distribution of the draws produced by the Gibbs sampler for the individual model parameters will approximate the posterior distribution of the associated variable.

2.4 Dirichlet process mixture models (DPMM)

The finite mixture models described previously require setting the number of components K. When K is unknown, practitioners usually search over a range of values, say between 5 to 10, before implementing some model selection criteria (AIC, BIC) to determine the final K. The "elbow" finding technique (Thorndike, 1953) is another popular method designed to help practitioners to determine K, where an evaluation metric (i.e. sum of squared errors) is chosen, and its value is computed across a wide range of k. One can then plot the value of the evaluation metric against its corresponding k. The appropriate K is the k such that the value of the metric at k sits at the "elbow" of the graph, where the marginal gain of an additional component is deemed insignificant. This approach can be computationally expensive as parameter estimations are done on a variety of different k. Furthermore, it adds another predefined variable to the problem, which is the range of k.

The above techniques can also be inappropriate because oftentimes the number of components in a mixture model grows as new data are observed. A Bayesian nonparametric approach to this problem is to set K equal to infinity, which allows K to flexibly grow and adjust with the size of the data. In our example of the finite Gaussian mixture model given by Equation (2.3), if we take the limit of K to infinity, the Dirichlet prior over $\boldsymbol{\pi}$ becomes the Dirichlet Process prior (Teh, 2010; Ferguson, 1973). Placing a Dirichlet Process prior over the mixing component enables the mixture model to have infinitely many clusters, hence the name infinite mixture models or nonparametric mixture models. Applications to DPMM includes: density modeling using mixture models, clustering (Dahl, 2006), classification (da Silva, 2007), haplotype inference in bioinformatics (Xing et al., 2007), topic modeling of documents in information retrieval (Blei and Lafferty, 2009), recommendation systems (Gong et al., 2015), and many others. We focus on a particular infinite mixture model called the Dirichlet Process mixtures of Gaussians (Lo, 1984; Escobar and West, 1995; Rasmussen, 2000).

Written as DP(α, G_0), the Dirichlet process is a stochastic process that describes a distribution over probability measures on Θ. It is parameterized by a base distribution G_0 (which is some probability distribution) and a real-valued concentration parameter α. For DP mixtures of Gaussians, the base measure G_0 is typically a Normal-Inverse-Wishart distribution. A sample from a DP is an infinite component discrete probability distribution (Figure 2.2) given over the parameter $\theta = (\mu, \Sigma)$. Equation (2.8) provides the generative process of the Dirichlet Process mixtures of Gaussians. The graphical model of this process is given in Figure 2.2.

$$
\begin{aligned}
G|\alpha, G_0 &\sim DP(\alpha, G_0) \\
\theta_i &\sim G \\
X_i|\theta_i = \{\mu_i, \Sigma_i\} &\sim \mathcal{N}(X_i|\mu_i, \Sigma_i)
\end{aligned}
\tag{2.8}
$$

In the next section, we review in more detail the properties of the Dirichlet Process prior, provide descriptions of the different representations, and present the generative process of the infinite mixture models associated with the different representations.

2.4.1 Dirichlet processes

Consider the Dirichlet Process, DP(α, G_0). Random draws from a Dirichlet Process are discrete probability measures consisting of locations and their weights (Figure 2.2). The concentration parameter α determines the sparsity of the location of the draws, with smaller values of α yielding few locations with large weights, and the remaining majority of locations with small weights. For a probability distribution over a random measure G to be considered as a DP, it must satisfy the following condition. Consider any finite partition $A_1, \ldots, A_K$ of Θ; G is distributed according to DP(α, G_0) if

$$
(G(A_1), G(A_2), \ldots, G(A_K)) \sim \mathrm{Dir}(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_K)). \tag{2.9}
$$

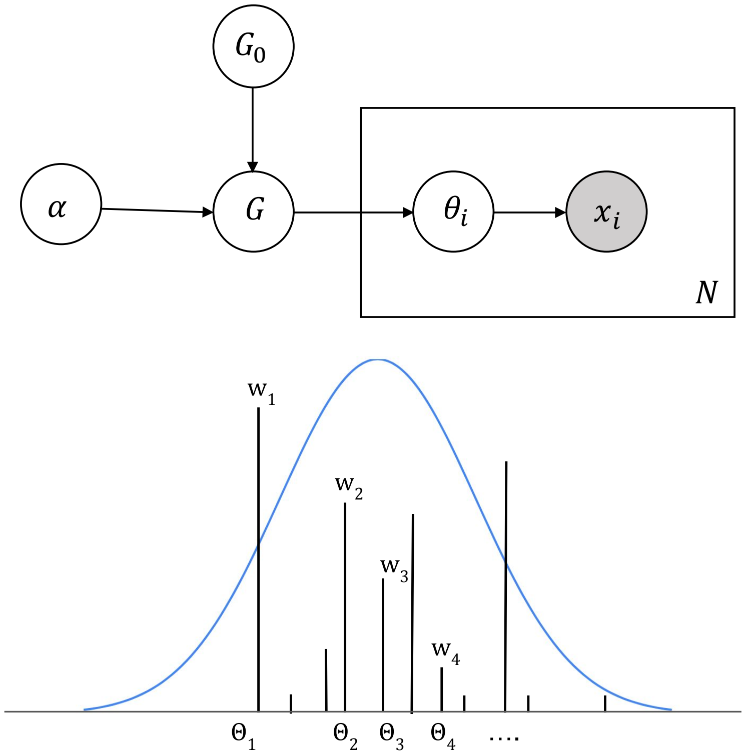


Figure 2.2.: Graphical representation of Dirichlet Process mixtures of Gaussian (top). Samples from a DP with base measure G_0 (blue curve) from a standard Normal distribution.

This condition suggests that the marginal distribution of a DP follows a Dirichlet distribution, with parameters involving the product of α and a projection of the base measure into the separate segments. Several properties follow this condition:

1. $E[G(A)] = G_0(A)$ for all A,
2. $\text{Var}[G(A)] = \frac{G_0(A)(1-G_0(A))}{1+\alpha}$ for all A,
3. As $\alpha \to \infty, G(A) \to G_0(A)$ for all A.

The second property indicates that the variance is inversely related to the concentration or strength parameter α. Greater α yields denser locations, leading to a lower variance in the mapping of G onto any A. On the other hand, if $\alpha \to 0$, G consists of a single atom with a weight of 1 and location drawn from G_0. It is important to note that the third property does not imply that G converges in distribution to G_0 as $\alpha \to \infty$. This is due to the fact that draws from a DP form a discrete distribution, even if G_0 is a continuous function (Teh, 2010).

Let $\theta_1, \ldots, \theta_n \in \Theta$ be draws from $G \sim DP(\alpha, G_0)$ and $A_1, \ldots, A_K$ be a finite partition of Θ. We can determine the number of realizations that belong in each partition A_k, denoting this by n_k; that is, $n_k = \sum_i^N I_{A_k}(\theta_i)$, where $I_{A_k}(.)$ is an indicator function for partition A_k. By the property of the marginal distribution of the DP in Equation (2.9) and the conjugacy between a Dirichlet distribution and a multinomial distribution, we have:

$$(G(A_1), G(A_2), \ldots, G(A_K))|(\theta_1, \ldots, \theta_n) \sim \text{Dir}(\alpha G_0(A_1) + n_1, \ldots, \alpha G_0(A_K) + n_K). \tag{2.10}$$

It follows that the posterior must also be a DP with parameters that are a function of the number of realizations in each partition. After some reparameterization, the posterior of the DP becomes:

$$(G(A_1), G(A_2), \ldots, G(A_K))|(\theta_1, \ldots, \theta_n) \sim \text{DP}\left(\alpha + n, \frac{\alpha}{\alpha + n} G_0 + \frac{n}{\alpha + n} \sum_i^n I(\theta_i)\right) \tag{2.11}$$

To sample a random probability distribution G that satisfies the properties described above, we can follow the stick-breaking scheme introduced in Sethuraman (1994). This samples G through the use of a Beta distribution, to provide the weight of each point drawn from the base distribution G_0. Observations X can then be drawn from G. It is also possible to sample observations from G directly without having to simulate G. This scheme is described in other constructions of the Dirichlet Process, specifically the Chinese Restaurant Process (Blackwell et al., 1973). Both of these techniques are used within this dissertation.

Stick-breaking construction

The stick-breaking construction uses the discreteness property of the DP to represent the random measure G as the sum of independent sequences of atomic masses as follows:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \qquad \pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_i). \tag{2.12}$$

The sequence of independent random variables $(\beta_k)_{k=1}^{\infty}$ and $(\phi_k)_{k=1}^{\infty}$ are distributed as:

$$\beta_k | \alpha_0, G_0 \sim \text{Beta}(1, \alpha_0) \qquad \phi_k | \alpha_0, G_0 \sim G_0. \tag{2.13}$$

It can be shown that $\sum_{k=1}^{\infty} \pi_k = 1$ almost surely. Notationally, the distribution over the countable sequence of weights $\boldsymbol{\pi}$ is usually referred to as $\boldsymbol{\pi} \sim \text{GEM}(\alpha_0)$ after Griffiths, Engen, and McCloskey (Pitman, 2002). We can then rewrite the DP mixture of Gaussian model of Equation (2.3) as:

$$\begin{aligned} \boldsymbol{\pi} | \alpha_0 &\sim \text{GEM}(1, \alpha_0) \\ c_i = k | \boldsymbol{\pi} &\sim \pi_k \\ \theta_k | \theta_0 &\sim G_0 \\ X_i | c_i = k, \boldsymbol{\theta} &\sim \mathcal{N}(X_i | \mu_k, \Sigma_k). \end{aligned} \tag{2.14}$$

Note that the observations are conditionally independent given the class indicator $\boldsymbol{c}$ and the component parameters $\boldsymbol{\theta}$. Due to the decreasing nature of the sequence of weights in this representation, it is possible to create a truncated approximation of the DP by specifying the number of components, K. Our models in the first part of this dissertation truncate the number of components at $K = 50$.

The Chinese Restaurant Process

Another way of simulating observation X from a DP is through the Chinese Restaurant Process (CRP) (Pitman, 1995). Consider a restaurant with an infinite number of tables with one dish per table. In this representation, each customer is the observation, X_i, generated by the mixture model. The tables are the components or clusters, and the dish on the table is the value of the parameter shared by observations at that table. The first customer enters and picks a table randomly. This is the cluster with parameter θ_1 that observation 1 belongs to ($c_1 = 1$). The customer takes a value from a Gaussian with parameter θ_1, $X_1 \sim \mathcal{N}(X_1|\theta_1)$. The second customer who comes in has two choices: join the table occupied by the first customer (with probability of $\frac{1}{1+\alpha}$), or pick a new table (with probability $\frac{\alpha}{1+\alpha}$), where α is a concentration parameter of the process. Having picked a table (cluster), the second customer takes a value of X_2 from a Gaussian with the parameter at the chosen table (cluster).

In the same fashion, the $(n+1)^{st}$ customer will pick an empty table with probability $\frac{\alpha}{n+\alpha}$ and join an occupied table c with probability $\frac{n_c}{n+\alpha}$, where n_c is the number of customers (or equivalently, observations) that are in table c, and n is the total number of customers at the restaurant. When a customer picks an occupied table, they share the dish (component parameter) that is already available at that table. On the other hand, if a customer chooses a new table, then a new dish (parameter) must be selected for that table. A new dish for an empty table is drawn from the base distribution G_0. The $(n+1)^{st}$ observation's cluster parameter is distributed as:

$$\theta_{n+1}|\theta_{1:n} \sim \frac{1}{\alpha+n}\left(\alpha G_0 + \sum_{c=1}^{K^*} n_c \delta_{\theta_c^*}\right), \tag{2.15}$$

with its value following $X_{n+1} \sim \mathcal{N}(X_{n+1}|\theta_{n+1})$. The CRP is invariant under permutations; this is formally known as exchangeability. Additionally, the seating arrangements across tables in the Chinese Restaurant Process forms a partition of customers. Any permutations of these partitions with an identical number of customers in each table will have the same probability. The expected number of occupied tables for n customers is given by $O(\alpha \log n)$. The next section discusses how this representation leads to the inference techniques of DPMM.

2.5 MCMC methods for DPMM

Markov chain Monte Carlo sampling is a family of algorithms to simulate draws from some complex distribution (Gamerman and Lopes, 2006; Gilks et al., 1995; Brooks et al., 2011). Given data assumed to be generated according to a proposed Bayesian model, samples from the posterior distribution of the parameters are of interest. As an example, in the finite mixture model setting given by Equation (2.3), samples from the posterior distribution of $\boldsymbol{\theta}$, $\boldsymbol{c}$, and $\boldsymbol{\pi}$ given the observed data $\boldsymbol{X}$ can be used to summarize the entire model. For more complex models, it may be infeasible or computationally inefficient to sample directly from the posterior density. In such cases, MCMC methods are used to simulate these samples.

Samples from an MCMC procedure are a sequence of random variables satisfying the Markov property, whereby each random variable depends only on the last preceeding value in the sequence. Additionally, these sequences must be constructed such that the chain converges to the posterior distribution. Through this process, the chain is able to improve its approximation of the target density, and for a long enough sequence, the equilibrium distribution of the chain is approximately the targeted posterior distribution.

This section looks at a specific type of MCMC method called Gibbs sampling (Casella and George (1992), Smith and Roberts (1993)). Thus, the update step at iteration t for θ_j is conditional on components that have already been updated at t, as well as on other components that have yet to be updated at iteration t, still having values from iteration $t-1$.

The main reference on Markov chain methods to estimate the posterior distribution of a DPMM is given by Neal (2000). This section reviews the general Gibbs sampling algorithm for DP mixture models for both conjugate priors and non-conjugate priors. These algorithms assume different representations of the DP, such as the CRP and the Stick-breaking prior. The Gibbs sampling procedure with the Stick-breaking prior is implemented in the development of the first part of this dissertation, while the non-conjugate Gibbs sampler is utilized for inference on a CRP-based DPMM of the tumor contamination problem.

We first discuss the Gibbs sampler for the CRP with a conjugate prior. This will provide a basic foundation to a more complicated setting such as the use of a non-conjugate prior, which we implement in this work.

Gibbs sampling based on the Chinese Restaurant Process

The exchangeability property of the CRP can be used to treat every observation as the last of n observations. Treating observation i as the last observation yields the following conditional distribution for its class assignment:

$$P(c_i = k|\boldsymbol{c^{\neg i}}, \boldsymbol{X}, \boldsymbol{\theta}) \propto P(X_i|c_i = k, \boldsymbol{c^{\neg i}}, \boldsymbol{\theta})P(c_i = k|\boldsymbol{c^{\neg i}}), \tag{2.16}$$

where $\boldsymbol{c^{\neg i}}$ is the collection of cluster indicators for every observations outside of observation i. Equation (2.15) provides the expression for the second term under the CRP. Let G_0 be NIW(θ_0). Then, combining with the likelihood of observation X_i we get:

$$P(c_i = k|\boldsymbol{c_{\neg i}}, X_i, \boldsymbol{\theta}) = \begin{cases} b\frac{n_{-i,k}}{n+\alpha-1}\mathcal{N}(X_i|\theta_k), & \text{for seen } k \\ b\frac{\alpha}{n+\alpha-1}\int \mathcal{N}(X_i|\theta_k)\text{NIW}(\theta_k|\theta_0)d\theta, & \text{for unseen } k, \end{cases} \tag{2.17}$$

where n is the total number of observations, $n_{-i,k}$ is the number of observations in component k outside of i, and b is the appropriate normalizing constant such that the equation is a proper density distribution. When an unseen k is sampled for a new c_i, θ_{c_i} is drawn from the posterior distribution of θ_{c_i}. In this example,

$$\theta_{c_i}|X_i \sim \mathcal{N}(X_i|\theta_{c_i})\text{NIW}(\theta_{c_i}|\theta_0). \tag{2.18}$$

Given all component memberships, we can now sample from the posterior distribution of $\boldsymbol{\theta}$. The component parameter for cluster k, θ_k, depends only on observations in k. Conjugacy leads to a simple closed form solution to the conditional distribution:

$$\theta_k|\boldsymbol{X_k}, \boldsymbol{c_k} \sim \text{NIW}(\theta_k|\theta_0) \prod_{\{i\in\boldsymbol{C}_k\}} \mathcal{N}(X_i|\theta_k), \tag{2.19}$$

where $\boldsymbol{C}_k$ is the collection of the index of observations in cluster k, and $\boldsymbol{X}_k$ is the collection of observations in cluster k. By conjugacy, Equation (2.19) is still NIW. The full algorithm is given in Algorithm 1 (Algorithm 2 in Neal (2000)).

Algorithm 1: Gibbs sampling with CRP prior and latent state

```
Input: Given $\boldsymbol{\theta}$ and $\boldsymbol{c}$ of the current state of the Markov chain
for $i = 1, \dots, n$ do
    If $n_{-i,c_i} = 0$, remove $\theta_{c_i}$ from the state;
    Draw a new $c_i$ from Equation (2.17);
    If the new $n_{-i,c_i} = 0$, draw $\theta_{c_i}$ from Equation (2.18).
end
for $k \in \{c_1, \dots, c_n\}$ do
    Draw a new value for $\theta_k$ from Equation (2.19)
end
Repeat
```

Gibbs sampling based on the truncated Stick-breaking representation

The CRP-based Gibbs sampler relies on the full conditional distribution $P(c_i|\boldsymbol{c}_{\neg \boldsymbol{i}}, \boldsymbol{X})$, which results in a sampler that updates one coordinate at a time. This can result in slow mixing and is inefficient as the total number of observations grows larger. In particular, the sampler can get stuck at certain values of c_i, and may take many iterations to break the pattern. These issues can be avoided by using a truncated stick-breaking prior which reduces the model to a finite-dimensional problem. The truncation of the prior into finite dimensionality allows blocks of parameters to be updated at the same time, giving rise to its name, the blocked Gibbs sampler (Ishwaran and James, 2001).

This Gibbs sampler iteratively samples from the conditional distributions of the variables $\boldsymbol{\theta}, \boldsymbol{c}$ and $\boldsymbol{\pi}$. Recall that GEM is the prior over the weights described in

Section 2.4.1. To derive the conditional posterior distributions for $\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\pi}$, we first determine its joint probability distribution, which is given by:

$$P(\boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{c}|\alpha_0, \theta_0) = \mathrm{GEM}(\boldsymbol{\pi}|\alpha_0) \prod_{k=1}^{K} \mathrm{NIW}(\theta_k|\theta_0) \prod_{i=1}^{N} \pi_{c_i} \mathcal{N}(X_i|\theta_k). \tag{2.20}$$

Here K is the total number of components allowed by the stick-breaking prior. To update the parameters of the model, we iteratively sample from its conditional distribution given as follows:

1. The conditional distribution for the mixing proportion $\boldsymbol{\pi}$:

   $$\pi_1 = V_1^* \quad \text{and} \quad \pi_k = (1-V_1^*)(1-V_2^*)\dots(1-V_{k-1}^*)V_k^*, \quad \text{for } k = 2, \dots, K-1, \tag{2.21}$$

   where

   $$V_k^* \sim \mathrm{Beta}\left(1 + M_k, \alpha_0 + \sum_{l=k+1}^{K} M_l\right)$$

   for $k = 1, \dots, K-1$, and M_k is the total number of c_is that equal to k.

2. The conditional distributions for $\boldsymbol{c}$:

   $$P(c_i = k|\boldsymbol{c}^{\neg \boldsymbol{i}}, \boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\theta}, \alpha_0, \theta_0) \propto \pi_{c_i} \mathcal{N}(X_i|\theta_k). \tag{2.22}$$

   Conditioned on $\boldsymbol{\pi}$, we can update all c_is independently. Recall that in the CRP, this update step is done sequentially.

3. The conditional distribution of the component parameter $\boldsymbol{\theta}$:

   $$P(\theta_k|\boldsymbol{c}, \boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\theta}_{\neg \boldsymbol{i}}, \alpha_0, \theta_0) \propto \mathrm{NIW}(\theta_k|\theta_0) \prod_{i \in C_k} \mathcal{N}(X_i|\theta_k), \tag{2.23}$$

   where C_k are the index of all observations in the k^{th} cluster.

Gibbs sampling involving non-conjugate priors

The algorithms in Section 2.5 can be challenging to implement when G_0 is not a conjugate prior for $\boldsymbol{\theta}$, as Equation (2.17) would often involve intractable integrals.

One way to get around this problem is by including an auxiliary parameter temporarily during the update process. When updating c_i, m auxiliary components are introduced with parameters drawn from G_0. If observation i is not associated with these auxiliary components at the end of the update step, the m parameter values are discarded, and new ones are to be drawn at the update step of the next observation.

The use of auxiliary variables is most useful for cases where it is more feasible to update with respect to the joint distribution rather than the marginal. Given two random variables x and y, our goal is to sample from $p(x)$ which involves evaluating intractable integrals. We can instead sample from another distribution $p(x, y)$ in which the marginal with respect to x is $p(x)$. Values of y are drawn from $p(y|x)$, and then parameter updates are performed on $p(x, y)$, before samples of y are discarded. As long as $p(x)$ is the marginal of x under $p(x, y)$, this process will converge to $p(x)$.

Let k^- be the total number of non-empty components and $h = k^- + m$. By using the auxiliary variable trick described above, the conditional cluster assignment of Equation 2.5 becomes

$$P(c_i = k|\boldsymbol{c_{-i}}, X_i, \theta_1, \ldots, \theta_h) = \begin{cases} b\frac{n_{-i,k}}{n+\alpha-1}\mathcal{N}(X_i|\theta_k), & \text{for } 1 \leq k \leq k^- \\ b\frac{\alpha/m}{n+\alpha-1}\mathcal{N}(X_i|\theta_k), & \text{for } k^- \leq k \leq h. \end{cases} \tag{2.24}$$

Note that now the conditional distribution do not involve any integration. The complete update process is given by Algorithm 2.24. The MCMC sampler of our proposed model for estimating tumor contamination involves a non-conjugate prior over the parameters of the components. The algorithm will be discussed with more detail in Chapter 8.

2.6 Summary

In this chapter, we reviewed mixture models, the main type of model used throughout this work. We started by discussing the finite mixture model, presented a concrete example of the model, and provided its MCMC algorithm in the Bayesian setting. Then we moved to the more complex setting of infinite mixture models. We reviewed the Dirichlet Process prior, described its properties and its two representations: the stick-breaking construction and the Chinese Restaurant Process. We then discussed how the Dirichlet Process is used in the mixture model setting to become what is commmonly known as the Dirichlet Process Mixture Models (DPMM). Lastly, we reviewed the MCMC techniques often utilized to perform posterior inference for DPMM. These techniques provide the key tools for the majority of the work in this dissertation.

Algorithm 2: Gibbs sampling for non-conjugate priors

```
Input: Given $\boldsymbol{c} = (c_1, \dots, c_n)$ and $\boldsymbol{\theta} = (\theta_k : k \in \{c_1, \dots, c_n\})$ of the current
       state of the Markov chain
for $i = 1, \dots, n$ do
    Set $k^-$ as the number of distinct values in $\boldsymbol{c}$ and $h = k^- + m$
    if $c_i = c_j$ for some $j \neq i$ then
        Draw values of $\theta_k$ from $G_0$ for which $k^- < k \leq h$
    if $c_i \neq c_j$ for all $j \neq i$ then
        Label $c_i = k^- + 1$;
        Draw values of $\theta_k$ from $G_0$ for which $k^- + 1 < k \leq h$
    Draw a new value for $c_i \in \{1, \dots, h\}$ from Equation (2.24)
end
for $k \in \{c_1, \dots, c_n\}$ do
    Draw a new value from $\theta_k | X_i$
end
Repeat
```

3. FLEXIBLE MIXTURE MODELS ON CONSTRAINED SPACES

3.1 Introduction

This chapter introduces the first problem this dissertation addresses, nonparametric mixture modeling on constrained spaces. We review the rejection sampling algorithm and describe how this algorithm is used to model observations lying within a constrained space. We introduce two mixture models, *truncated mixtures of Gaussians* (TMoG) and *mixtures of truncated Gaussians* (MoTG), and we discuss and illustrate their data generating processes. This chapter provides the modeling framework within which our algorithmic contributions and extensions are applied.

3.2 Rejection sampling

We first describe the rejection sampling (Gilks and Wild, 1992), or the accept-reject, algorithm that is used as the main building block of the algorithms presented in this dissertation. Suppose we are interested in sampling from a distribution $p(x)$ on some space of x. Assume that directly sampling from p is infeasible due to the complexity of the distribution. Suppose there exists another distribution q which we already know how to sample from that satisfies $\frac{p(x)}{q(x)} \leq M$ for a known constant M. This ensures that the support of p is a subset of the support of q, otherwise, there would be regions of p that are never sampled by q. The density p will be referred to as the “target density”, and q will be the “proposal density.” We can then draw proposals x from q and accept or reject x with a probability of $\frac{p(x)}{Mq(x)}$. Under this process, all accepted draws will be distributed as p. The rejection sampling algorithm can be summarized as follows:

1. Sample x from q.

2. With probability $\frac{p(x)}{Mq(x)}$, accept x as a draw from p. Otherwise, return to 1.

Observe that if $p(x) \propto \mathbb{1}_{\mathbb{S}} q(x)$, where $\mathbb{1}_{\mathbb{S}}(.)$ is an indicator function for $\mathbb{S}$, then proposals from q inside $\mathbb{S}$ are always accepted and are always rejected from outside $\mathbb{S}$.

This process can be repeated until the desired number of draws from p is attained. The constant M controls the efficiency of the algorithm and a large value can cause the rejections of many samples, leading to inefficiencies in the process. One important property of the rejection sampling algorithm is that the number of draws needed from the candidate density q before acceptance is a geometric random variable with success probability $\frac{1}{M}$.

3.3 Rejection sampling on complex spatial domains

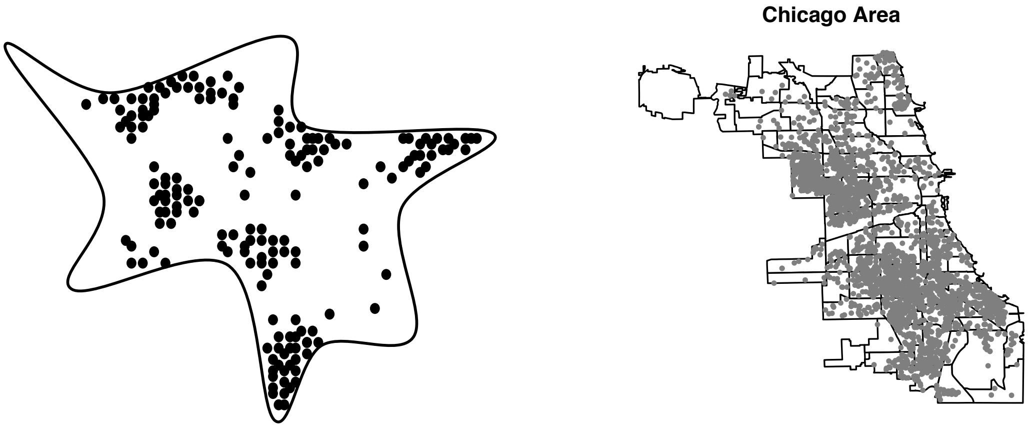


Figure 3.1.: Data lying on constrained subsets of Euclidean space. The left pane shows synthetic observations lying in the space $\mathbb{S}$ specified by the solid black line. The right pane shows the location of homicide events in the city of Chicago. Here, the city limit defines the space $\mathbb{S}$.

Given the above definition of rejection sampling, we can translate the idea to model observations on a more complex spatial space. Consider observations $X = \{x_1, \ldots, x_n\}$ lying on a subset $\mathbb{S}$ of a Euclidean space $\mathbb{X}$. The set $\mathbb{S}$ might be the set of

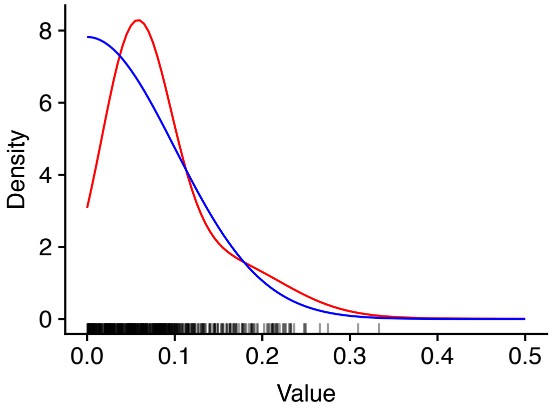


Figure 3.2.: An illustration of edge effects for observations distributed as $\mathcal{N}(0, 0.1)$ truncated to $[0, 1]$. Shown in red is the posterior predictive density of the truncated density when estimated with a vanila mixture of Gaussians with no consideration of $\mathbb{S} = [0, 1]$. Shown in blue is the real density of a truncated $\mathcal{N}(0, 0.1)$ in $\mathbb{S}$.

positive reals, the unit sphere, or something more interesting like the city of Chicago. Examples of such observations could be crime events in Chicago or coyote sightings in Rhode Island where the observations are a 2-dimensional vector consisting of latitudes and longitudes representing the location of the events. We show a simulated and a real example in Figure 3.1.

We model the observations as i.i.d. draws from a probability distribution $p(x)$ whose support equals $\mathbb{S}$. Our goal is to estimate $p(x)$ from the observations X, and to do so, we will take a Bayesian approach, placing a prior over $p(x)$ and studying the resulting posterior distribution. Unfortunately, the requirement that $p(x)$ be restricted to $\mathbb{S}$ raises challenges for both model specification as well as computation. For many interesting settings, the probability density $p(x)$ will involve an integral over $\mathbb{S}$, and will therefore be intractable for all but simple choices of $\mathbb{S}$. Posterior inference is consequently a doubly-intractable problem (Murray et al., 2006).

Note that it is important to account for edge effects in any modeling approach. Typically, probability density inside the constraint will be smooth. However, the absence of observations outside the constraint set will result in the probability density at the boundaries being underestimated (Figure 3.2). For a simple constraint set like the unit square, edge effects can be avoided by changing the parametric family used for the mixture components, (e.g. in Kottas and Sansó (2007) and Matechou et al. (2017), the authors used nonparametric mixtures of beta or gamma distributions). However, such models cannot easily model correlation structure in multi-dimensional data, something that is natural to Gaussian mixture models. Another approach is to transform the data to be unconstrained, and then model the transformed data with a mixture of Gaussians. This can suffer from edge-effects that are not easy to characterize. Importantly, both approaches are also not applicable to more complex constraint sets, like the city of Chicago (Figure 3.1).

Our approach is to regard the distribution $p(x)$ as a restriction and renormalization on $\mathbb{S}$ of some other distribution $q(x)$ on the ambient space $\mathbb{X}$. Thus, $p(x) \propto \mathbb{1}_{\mathbb{S}}(x)q(x)$, where $\mathbb{1}_{\mathbb{S}}(\cdot)$ is the indicator function for $\mathbb{S}$, and $q(x)$ is a standard distribution, chosen such that MCMC posterior sampling techniques already exist. Defining $p(x)$ this way allows sharp drops in the probability density from inside to outside $\mathbb{S}$ and avoids undesirable smoothing effects across the boundaries. We restrict ourselves to subsets having positive probability under $q(x)$ and, in practice, to subsets of $\mathbb{X}$ having positive Lebesgue measure.

Observe that the probability $p(x)$ has an intractable normalizing constant given by:

$$z = \int_{\mathbb{S}} q(x)dx.$$

Despite this, it is easy to simulate from $p(x)$ by rejection sampling with an acceptance probability of 1 for inside $\mathbb{S}$ and 0 outside. Thus, the rejection sampling algorithm from Section 3.2 now becomes:

1. Propose x from q

2. Accept x if $x \in \mathbb{S}$, otherwise, go to 1.

At a high level, our strategy is to specify a flexible, possibly nonparametric prior over the distribution $q(x)$ and thus implicitly over $p(x)$. Placing a prior over the unconstrained distribution $q(x)$, allows us to avail of standard Bayesian modeling tools. We treat the observations X as the outcome of a rejection sampling algorithm, where we propose from $q(x)$ and discard samples falling outside of $\mathbb{S}$.

3.4 Mixture modeling on constrained spaces

One characteristic of data given in the above examples is that X is often not evenly distributed throughout the domain $\mathbb{S}$. For instance, some areas of Rhode island can experience more coyote sightings because it is less densely populated or are near their food sources and like any other city, there are regions of Chicago where crime rates are higher. A common approach to model such data is through the use of mixture models, as described in Chapter 2. Mixture models allow flexibility in estimating more complex densities, especially those with multimodality. Unfortunately, specifying a mixture model on a complex domain like in Figure 3.1 is not easy. We thus take the rejection sampling approach explained previously. More concretely, $p(x)$ will be a mixture model truncated within $\mathbb{S}$, and $q(x)$ will be a proposal distribution on the whole space. Proposals within $\mathbb{S}$ are accepted with probability of 1, and those that are outside are rejected. We parametrize both the constrained density of interest, as well as the proposal density, by θ, writing them respectively as $q(x|\theta)$ and $p(x|\theta) \propto \mathbb{1}_{\mathbb{S}}(x)q(x|\theta)$. The proposal density $q(x|\theta)$ is a mixture model, and the parameter θ represents the mixing proportions π, as well as the component parameters β, where, in the case of Gaussians, $\beta = (\mu, \Sigma)$.

Following Chapter 2, in a Bayesian setting, it is typical to place a Dirichlet prior over π, and in nonparametric setting, a DP prior over π. When working with a mixture of Gaussians, we would place a Normal-Inverse-Wishart (NIW) prior over the (μ, Σ) pairs. Let θ_0 be the hyperparameter for the prior over the component parameters and α_0 be the hyperparameter for the distribution over weights. We write the latter as

Dir, though it could be either the Dirichlet or a stick-breaking prior. Then, for the case of Gaussian likelihoods, the prior over parameters can be written as follows:

$$\pi|\alpha_0 \sim \mathrm{Dir}(\alpha_0), \quad (\mu_k, \Sigma_k)|\theta_0 \sim \mathrm{NIW}(\theta_0), \quad k = 1, 2, \cdots . \tag{3.1}$$

3.4.1 Truncated Mixture of Gaussians (TMoG)

In the first approach we consider, our proposal distribution $q(x|\theta)$ is a simple unconstrained mixture model. Assume K components, each with its own mean $\mu_k \in \mathbb{R}^d$ and covariance matrix $\Sigma_k \in \mathbb{R}^{d\times d}$. Then the proposal has the form

$$q(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k). \tag{3.2}$$

The parameter θ consists of the mixing distribution π and the K component parameters $\{(\mu_1, \Sigma_1), \dots, (\mu_K, \Sigma_K)\}$. In nonparametric settings with a Dirichlet process prior, K is infinity, and the proposal distribution becomes a Dirichlet process mixture of Gaussians. We assume our constraint set $\mathbb{S}$ is some subset of $\mathbb{R}^d$ with nonzero Lebesgue measure, so that the observations follow a density equal to $q(x|\theta)$, truncated to $\mathbb{S}$ and renormalized. We call this distribution a truncated mixture of Gaussians (TMoG):

$$p(x|\theta) = \frac{\mathbb{1}_{\mathbb{S}}(x) \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\int_{\mathbb{S}} \mathrm{d}x \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}. \tag{3.3}$$

To simulate observations from this model, we make proposals from equation (3.2) until one lies in $\mathbb{S}$. We note that since proposals are made independently from the mixture distribution of Equation (3.2), rejected proposals preceding a particular observation need not belong to the same cluster. Figure 3.3 describes the generative process of this model in more detail. The full model with the stick-breaking prior is given by the following:

$$\begin{aligned} \pi|\alpha_0 &\sim \mathrm{GEM}(1, \alpha_0) \\ (\mu_k, \Sigma_k)|\theta_0 &\sim \mathrm{NIW}(\theta_0), \quad k = 1, 2, \dots \\ x_i, c_i = k|\boldsymbol{\pi}, \{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k})\}_{k=1}^{\infty} &\sim \mathrm{TMoG}(\{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k})\}_{k=1}^{\infty}, \boldsymbol{\pi}) \end{aligned} \tag{3.4}$$

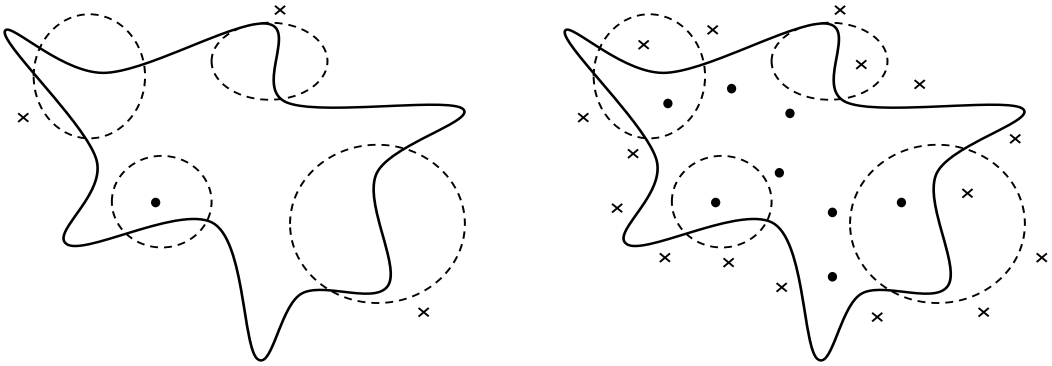


Figure 3.3.: The generative process of TMoG starts by selecting a cluster and sampling a single draw from it. If this falls outside $\mathbb{S}$ (the solid line), again select a cluster, and sample another draw from that cluster. The process is repeated until a cluster component produces an accepted draw (left). When the desired number of accepted draws is reached, the set of accepted draws is distributed as TMoG (right). Dashed lines represent cluster likelihoods; dots and crosses are accepted and rejected proposals, respectively.

3.4.2 Mixtures of Truncated Gaussians (MoTG)

In the previous section, we modeled an unknown density on a constrained space with a truncated mixture model (in particular, a truncated mixture of Gaussians). In this section, we take a second approach, modeling the density as a *mixture of truncated distributions* (for concreteness, a mixture of truncated Gaussians). For a subset $\mathbb{S}$ of a d-dimensional Euclidean space, write $\mathcal{N}_{\mathbb{S}}(x|\mu,\Sigma)$ for a Gaussian with mean μ and covariance Σ restricted to that subset:

$$\mathcal{N}_{\mathbb{S}}(x|\mu,\Sigma) = \frac{\mathbb{1}_{\mathbb{S}}(x)\mathcal{N}(x|\mu,\Sigma)}{\int_{\mathbb{S}}\mathcal{N}(x|\mu,\Sigma)dx}. \tag{3.5}$$

A mixture of K truncated Gaussians with parameters $\{(\mu_1,\Sigma_1),\ldots,(\mu_K,\Sigma_K)\}$ and with mixing proportion π has probability density given by:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}_{\mathbb{S}}(x|\mu_k,\Sigma_k). \tag{3.6}$$

Unlike the truncated mixture of Gaussians which involves a single intractable normalization constant (Equation (3.2)), the equations above show that the mixture of truncated Gaussian involves K (albeit simpler) intractable normalization constants. As before, we place Dirichlet/stick-breaking priors on π and a conjugate Normal-Inverse-Wishart prior on the components parameters (μ_k, Σ_k).

The generative process then follows:

$$\pi|\alpha_0 \sim \text{Dir}(\alpha_0), \quad (\mu_k, \Sigma_k)|\theta_0 \sim \text{NIW}(\theta_0), \quad k = 1, \dots, K,$$
$$X_i|c_i, \{(\mu_k, \Sigma_k)\}_{k=1}^{K} \sim \mathcal{N}_{\mathbb{S}}(x_i|\mu_{c_i}, \Sigma_{c_i}), \quad c_i|\pi \sim \text{Multinomial}(\pi), \quad i = 1, \dots, N.$$

For mixture models with the stick-breaking construction, we change the distribution of the weight vector to GEM(1, α_0).

Having chosen the cluster c_i of observation i from the distribution π, the challenge now is to sample from the corresponding truncated normal distribution $\mathcal{N}_{\mathbb{S}}(x|\mu_{c_i}, \Sigma_{c_i})$. To do this, we again use rejection sampling, now proposing from the unconstrained Gaussian distribution $\mathcal{N}(x|\mu_{c_i}, \Sigma_{c_i})$ until acceptance. Figure 3.4 outlines this process in more detail.

3.5 Comparison between TMoG and MoTG

We have described two models for representing data on a constrained space. We showed how both models have different data-generating mechanisms, but both can be used to represent the same observed data within a bounded region. In this section, we will explore in more detail the relationship of the two models. First we will look at a setting where the value of the component parameters are the same across the two models, in that $\theta_i^{\text{TMoG}} = \theta_i^{\text{MoTG}}$. We show there exist component weights that make the two densities equal. Then, we show by means of a simulation and a toy example that if both the weights and the component parameters are the same, the resulting probability density will be different.

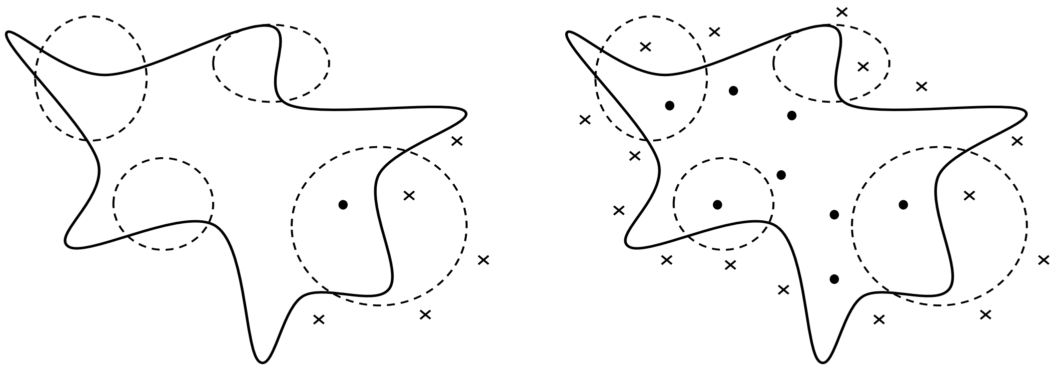


Figure 3.4.: The generative process of MoTG starts by picking a cluster and sampling from the associated unconstrained distribution until an acceptance (left). This is repeated until the desired number of acceptances are made (right). The set of all accepted draws are distributed as MoTG on $\mathbb{S}$. The solid line is the constraint set $\mathbb{S}$, the dashed lines represent the cluster components; dots and crosses are accepted and rejected proposals, respectively.

Set $\theta_k^{\text{MoTG}} = \theta_k^{\text{TMoG}}$. Denote $\boldsymbol{w}$ as the weights of the components in a MoTG and $\tilde{\boldsymbol{w}}$ as the weights of the components in a TMoG. Given x, an observation in $\mathbb{S}$, the probability of x under the two models are given by

$$p(x|\boldsymbol{\theta}, \boldsymbol{w})^{\text{MoTG}} = \frac{w_1 \mathcal{N}(x|\theta_1)}{\int_{\mathbb{S}} \mathcal{N}(x|\theta_1)dx} + \cdots + \frac{w_K \mathcal{N}(x|\theta_K)}{\int_{\mathbb{S}} \mathcal{N}(x|\theta_k)dx} \tag{3.7}$$

and,

$$p(x|\boldsymbol{\theta}, \tilde{\boldsymbol{w}})^{\text{TMoG}} = \frac{\tilde{w_1} \mathcal{N}(x|\theta_1)}{\int_{\mathbb{S}} \sum_{k=1}^{K} \tilde{w}_1 \mathcal{N}(x|\theta_k)dx} + \cdots + \frac{\tilde{w_K} \mathcal{N}(x|\theta_K)}{\int_{\mathbb{S}} \sum_{k=1}^{K} \tilde{w}_1 \mathcal{N}(x|\theta_k)dx}. \tag{3.8}$$

We can find a set of weights such that the value of Equation (3.8) equals that of Equation (3.7) for observation x. Under this assumption, the relationship between the weights is given by

$$w_k = \frac{\tilde{w}_k \int_{\mathbb{S}} \mathcal{N}(x|\theta_k)}{\int_{\mathbb{S}} \sum_{k=1}^{K} \tilde{w}_1 \mathcal{N}(x|\theta_k)dx}, \qquad k = 1, \cdots, K. \tag{3.9}$$

Additionally, we visually compare the probability density of both models through simulation. Setting $K = 50$, we first sample component weights from a Dirichlet

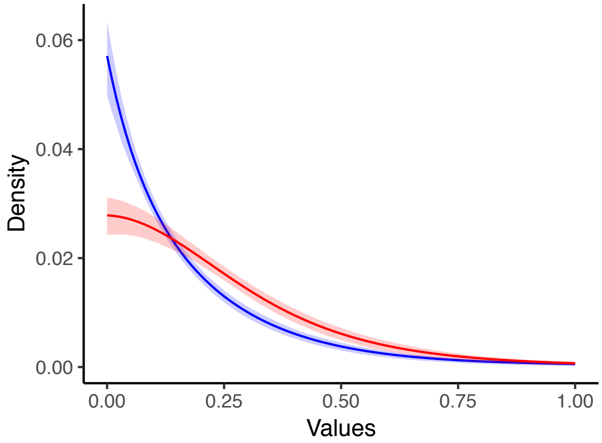


Figure 3.5.: Probability densities of MoTG (blue) and TMoG (red) for points in $[0,1]$ through random simulation from the prior. The shaded area represents the area between the first and third quartiles of the distribution.

distribution with parameter 1 and component parameters from a Normal-Inverse-Gamma with parameters $\mu_0 = 0, \lambda_0 = 2, \alpha_0 = 2$, and $\beta_0 = 0.05$. We computed the values of Equation (3.7) and Equation (3.8) for $x \in [0,1]$. By repeating this process 1000 times, we then find the mean, first quartile, and third quartile of the distribution. Figure 3.5 visualizes our result, which indicates that the two models have different probability densities.

Lastly, we will show that in a scenario where both the component parameters and the weights are equal across the two models, the probability densities of the two functions will differ as well. Suppose we have one-dimensional continuous observations that lie in $\mathcal{D} = [0,1]$. For illustration, suppose we model these observations with a mixture of two Gaussian distributions, $\mathcal{N}(0, 0.1)$ and $\mathcal{N}(0.2, 0.1)$. Assume that the weights given to the first and second Gaussians are $w_1 = \frac{1}{4}$ and $w_2 = \frac{3}{4}$, respectively. The domain D here is simple enough that the analytical solution is easy to compute with standard numerical integration techniques. Within this setup, we would like to compute the density of a point in the domain; let this point be $x = 0.3$. Write $\boldsymbol{w}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ as the collection of the weights, mean, and standard deviation, respectively. The analytical solution is given as follows:

For TMoG, we have:

$$p(x|\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{\sum_{i=1}^{2} w_i N(x|\mu_i, \sigma_i)}{\int_{\mathcal{D}} \sum_{i=1}^{2} w_i N(x|\mu_i, \sigma_i) dx} = 2.128197.$$

Similarly, for MoTG, we get:

$$P(x|\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{2} \frac{w_i N(x|\mu_i, \sigma_i)}{\int_{\mathcal{D}} N(x|\mu_i, \sigma_i) dx} = 1.39831.$$

The above results suggest that if the component weights, mean, and standard deviation are the same across the two models, their analytical solutions are different. In our simulation studies in Chapter 5, we show that these differences are greater in more complex constraints, compared to that of simple contraints, such as the unit square seen in this example.

3.6 Summary

In this chapter, we briefly reviewed the rejection sampling method and described how this mechanism can be translated to modeling observations on a constrained space. We proposed two models, truncated mixtures of Gaussians (TMoG) and mixtures of truncated Gaussians (MoTG), and illustrated their data generating processes. We concluded this chapter by comparing the two models via an analytical toy example and showed that they differ. In the next chapter, we discuss the MCMC algorithms, specifically, the Gibbs sampling methods associated with each of these models.

4. MCMC FOR MIXTURE MODELS ON CONSTRAINED SPACES

4.1 Introduction

This chapter provides the main computational algorithms for the two models we introduced in Chapter 3. We review a data augmentation scheme to impute rejected samplers given observations from a rejection sampling method. This was proposed in Beskos et al. (2006) and Rao et al. (2016). We show how this mechanism can be incorporated in the original Gibbs sampling scheme for infinite mixture models described in Chapter 2. We described the exact MCMC schemes for the two models that target the exact posterior over the distribution q and, through this, the posterior distribution over p. We present the joint distribution as well as the conditional distributions for each update step of the Gibbs sampler. In addition, we introduce and justify a modification to the original sampler that aims to speed-up computation, reduce variance, and improve mixing.

4.2 Data augmentation for rejection sampling

We first describe a general data augmentation scheme for MCMC inference on models with truncation described in Chapter 3. Recall that observations X were accepted proposals from the proposal distribution q and a prior over q. Given X, we wish to sample from the posterior distribution over $q(x)$, implicit in which is all information contained in the posterior over $p(x)$. While $q(x)$ is a simpler object than $p(x)$, its conditional distribution given the observations X is still not easy to sample from. In order to do this, we recognize that if we augment the observations X with the rejected proposals from $q(x)$, then conditional inference over $q(x)$ is straightforward. In particular, the rejected proposals (call these Y) together with the observations

X form i.i.d. samples from the unconstrained model q. This allows the use of standard MCMC methods for posterior inference over q, and imputing Y eliminates any intractable integrals arising from the constraint set $\mathbb{S}$.

The question now is how to impute the rejected proposals Y. In Rao et al. (2016), it was shown that the set of rejected samples preceding each observation are exchangeable across different observations. Consequently, in order to reconstruct the rejected samples for any observation x, one merely has to sample a new observation from the rejection sampler on $\mathbb{S}$ and associate all rejected samples to x. Concretely, this involves simulating from the proposal distribution $q(x)$ until an acceptance and, after discarding the accepted sample, assigning the remainder to observation x. This idea was first proposed by Beskos et al. (2006) in the specific setting of parameter inference for stochastic differential equations. Repeating this for each observation in the dataset X allows all discarded samples to be imputed.

Recall the model described in Section 3.4, where we are interested in the posterior of the parameter θ given observations X, $p(\theta|X)$. To sample from this distribution, we simulate from the distribution $p(\theta, Y|X)$, which has $p(\theta|X)$ as its marginal distribution. We sample on this augmented space by repeating two Gibbs steps:

1. Simulate the rejected proposals Y given the parameter θ

2. Update the parameter θ given the rejected samples Y, targeting the density $q(\theta|X, Y)$.

Algorithm 3 provides the general data augmentation scheme. A proof of its correctness can be found in Rao et al. (2016). The next sections will describe the exact Gibbs sampling methods based on this data augmentation scheme for both TMoG and MoTG.

Algorithm 3: An iteration of MCMC for posterior inference over $p(\theta|X)$ (Rao et al., 2016)

```
  Data: The observations $X = \{x_1, \dots, x_n\}$, and the current parameter values $\theta$
  Result: New parameter value $\tilde{\theta}$
1 for each observation $x_i$ do
2     while an accepted sample $\hat{x}$ has not been drawn do
3         draw $y$ independently from $q(\cdot|\theta)$;
4     end
5     Discard $\hat{x}$ and treat the preceding rejected proposals as $Y_i$;
6 end
7 Gather all the rejected samples, calling them $Y$: $Y = \bigcup_{i=1}^{n} Y_i$;
8 Update $\theta$ from $q(\theta|X, Y) \propto q(X, Y|\theta)p(\theta)$ using any MCMC kernel, calling the
    new value $\tilde{\theta}$;
9 Discard the rejected samples $Y$
```

4.3 Gibbs sampling for TMoG

Recall the full TMoG models given by Equation (3.4). Given observations X from this process, MCMC inference over the parameters involves first imputing the rejected proposals conditioned on the parameters and then updating the parameters given these imputed variables. Having updated the parameters, we discard the rejected samples and repeat the process. We describe the steps of the overall Gibbs sampler below.

Imputing the rejected proposals Y: To impute the rejected proposals given X and $\theta = (\pi, \mu, \Sigma)$, we follow steps 1 to 6 of Algorithm 3, proposing from the mixture of normals (Equation (3.2)) to generate a pseudo-dataset of the same size as X, keeping all the rejected proposals generated along the way. Call these Y, and write R for the total number of elements in Y. Each element y_i

lies outside the constraint set $\mathbb{S}$ and is associated with a mixture component c_i^* from which it was drawn. Write $C^* = \{c_1^*, \ldots, , c_R^*\}$ for the set of cluster assignments of the imputed proposals. Y and C^*, together with X, will be used to update the parameters θ, as well as the cluster assignments of the observations (write these as $C = \{c_1, \ldots, c_n\}$). The joint probability is

$$p(X, Y, \pi, \mu, \Sigma, C, C^* | \alpha_0, \theta_0) = \text{Dir}(\pi|\alpha_0) \prod_{k=1}^{K} \text{NIW}(\mu_k, \Sigma_k|\theta_0) \prod_{i=1}^{N} \pi_{c_i} \mathcal{N}(x_i|\mu_{c_i}, \Sigma_{c_i}) \times \prod_{r=1}^{R} \pi_{c_r^*} \mathcal{N}(y_r|\mu_{c_r^*}, \Sigma_{c_r^*}). \tag{4.1}$$

Updating C, π, and the (μ_k, Σ_k)'s is now straightforward as described next.

Updating the mixing proportions π: Write n_k and m_k for the total number of observations and rejected samples, respectively, in cluster k. These are easily calculated from C and C^*. For a Dirichlet distribution prior over π, the Gibbs conditional over π takes the simple form

$$p(\pi|X, Y, \theta, C, C^*, \alpha_0, \theta_0) = \text{Dir}(n_1 + m_1 + \alpha_0, \ldots, n_K + m_K + \alpha_0). \tag{4.2}$$

For a nonparametric stick-breaking prior over π, the conditional update for π is a simple adaptation of standard methodology (such as in Ishwaran and James (2001)) and is similar to that given in Equation (2.21). Specifically, to update the mixing proportion when a stick-breaking prior with truncation is used, we follow

$$\pi_1 = V_1^* \quad \text{and} \quad \pi_k = (1 - V_1^*)(1 - V_2^*) \ldots (1 - V_{k-1}^*) V_k^*, \quad \text{for } k = 2, \ldots, K-1 \tag{4.3}$$

where

$$V_k^* \sim \text{Beta}\left(1 + n_k + m_k, \quad \alpha_0 + \sum_{l=k+1}^{K} (n_l + m_l)\right) \tag{4.4}$$

for $k = 1, \ldots, K-1$. The key point implied by Equation (4.2) and (4.3) for any prior over π is to include *both* observations and rejected samples in the cluster counts.

Updating the cluster assignments c_i: Use $\neg i$ to represent quantities calculated after excluding observation i. The conditional distribution for c_i, the cluster assignment of observation i, is then given by

$$p(c_i = k | C^{\neg i}, C^*, X, Y, \theta, \alpha_0, \theta_0) \propto \pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k). \tag{4.5}$$

Thus, conditioned on π, all c_is can be updated independently. Alternately, one can also marginalize out π and update c_is sequentially. Now, if the c_i's follow a Pólya urn/Chinese restaurant process update (Neal, 2000), the conditional distribution is then similar to that of Equation (2.17). Again, we must consider *both* the number of observations as well as the number of rejected proposals at any cluster, as given by the following:

$$P(c_i = k | C^{\neg i}, C^*, X, Y, \theta, \alpha_0, \theta_0) \propto \begin{cases} (n_k^{\neg i} + m_k^{\neg i}) \mathcal{N}(x_i | \theta_k), & \text{for seen } k \\ \alpha_0 \int \mathcal{N}(x_i | \theta_k) \text{NIW}(\theta_k | \theta_0) d\theta_k, & \text{for unseen } k \end{cases} \tag{4.6}$$

where $\theta_k = (\mu_k, \Sigma_k)$.

Updating the cluster parameters (μ_k, Σ_k): The conditional distribution for the cluster parameters depends both on observations and rejected samples assigned to that cluster. Writing C_k and C_k^* for the indices of observations and rejected samples assigned to cluster k, respectively, and $\theta^{\neg k}$ for all parameters except (μ_k, Σ_k), we have

$$\begin{aligned} p(\mu_k, \Sigma_k | \theta^{\neg k}, X, Y, C, \alpha_0, \theta_0) &\propto p(\mu_k, \Sigma_k | \theta_0) \prod_{i \in C_k} p(x_i | \mu_k, \Sigma_k) \prod_{r \in C_k^*} p(y_r | \mu_k, \Sigma_k) \\ &\propto \text{NIW}(\mu_k, \Sigma_k | \theta_0) \prod_{i \in C_k} \mathcal{N}(x_i | \mu_k, \Sigma_k) \prod_{r \in C_k^*} \mathcal{N}(y_r | \mu_k, \Sigma_k). \end{aligned} \tag{4.7}$$

The parameters for all components can be updated independently and with a conjugate Normal-Inverse-Wishart prior, sampling from this distribution is easy. We expand the hyperparameters of the NIW to $\theta_0 = (\mu_0, \lambda_0, \Phi_0, \nu_0)$.

Then, the posterior distribution of the parameters of component k is also a Normal-Inverse-Wishart distribution with parameters given by the following:

$$p(\mu_k, \Sigma_k | \theta^{\neg k}, X, Y, C, \alpha_0, \theta_0) = \mathrm{NIW}(\mu_k, \Sigma_k | \mu_n, \lambda_n, \Phi_n, \nu_n) \tag{4.8}$$

$$\mu_n = \frac{\lambda_0 \mu_0}{\lambda_0 + (n_k + m_k)} + \frac{(n_k + m_k) + \bar{z}}{\lambda_0 + (n_k + m_k)} \tag{4.9}$$

$$\lambda_n = \lambda_0 + (n_k + m_k) \tag{4.10}$$

$$\nu_n = \nu_0 + (n_k + m_k) \tag{4.11}$$

$$\Phi_n = \Phi_0 + S + \frac{\lambda_0 + (n_k + m_k)}{\lambda_0 + (n_k + m_k)} (\bar{z} - \mu_0)(\bar{z} - \mu_0)^T, \tag{4.12}$$

where

$$\bar{z} = \frac{1}{n_k + m_k} \left(\sum_{i \in C_k} x_i + \sum_{r \in C_k^*} y_r \right) \tag{4.13}$$

$$S = \sum_{i \in \{C_k \cup C_k^*\}} (z_i - \bar{z})(z_i - \bar{z})^T, \qquad z \in X \cup Y. \tag{4.14}$$

This completes all update steps by the Gibbs sampler for TMoG.

4.4 Gibbs sampling for MoTG

Recall from Section 3.4.2 that unlike the truncated mixtures of Gaussians model, in MoTG all rejected samples associated with an observation come from the same component as that observation. This results in subtle differences in the associated MCMC sampler, where now rejected proposals of each observation must be imputed in a cluster-specific manner. Having imputed these, we must update cluster assignments and parameters, again in a manner slightly different from the TMoG case. As before, at the end of these updates, we discard the imputed samples and repeat. We describe the full Gibbs sampling procedure below.

Imputing the rejected proposals Y: Unlike TMoG, where each observation is an accepted proposal from a mixture of Gaussians, now each observation is an accepted proposal from the single Gaussian component to which it was initially assigned. Accordingly, to impute the rejected proposals for observation x_i belonging to cluster c_i, we repeatedly propose from component c_i until acceptance and keep all the rejected proposals. We now have to keep track of which rejected proposals belong to which observation, and we write Y_i for the set of rejected samples preceding observation x_i, R_i for the cardinality of Y_i, and C for the collection of all c_i. The joint probability density is then

$$\begin{aligned} p(X, Y, \pi, \theta, C|\alpha_0, \theta_0) =& \mathrm{Dir}(\pi|\alpha_0) \prod_{k=1}^{K} \mathrm{NIW}(\mu_k, \Sigma_k|\theta_0) \prod_{i=1}^{N} p(c_i|\pi) \\ & \left(\mathcal{N}(x_i|\mu_{c_i}, \Sigma_{c_i}) \prod_{r=1}^{R_i} \mathcal{N}(y_{ir}|\mu_{c_i}, \Sigma_{c_i}) \right). \end{aligned} \tag{4.15}$$

We use this to update the latent variables as described below.

Updating the mixing proportion π: Write n_k for the number of observations assigned to component k. The conditional distribution for π follows a Dirichlet distribution:

$$p(\pi|X, Y, \theta, C, \alpha_0, \theta_0) \propto \mathrm{Dir}(n_1 + \alpha_0, \ldots, n_K + \alpha_0). \tag{4.16}$$

Unlike TMoG, this distribution does not involve the rejected samples at all. This is because for each observation, a cluster is chosen from π only once, with all rejected proposals assigned to that same component.

For any other prior over π (e.g. a Dirichlet process), the conditional update rule remains unchanged from standard methodology, involving only cluster assignment counts of observations. Thus, when the truncated stick-breaking prior is used over π, we update the mixing proportion as follows:

$$\pi_1 = V_1^* \quad \text{and} \quad \pi_k = (1 - V_1^*)(1 - V_2^*) \ldots (1 - V_{k-1}^*)V_k^*, \quad \text{for } k = 2, \ldots, K - 1, \tag{4.17}$$

where

$$V_k^* \sim \mathrm{Beta}\left(1 + n_k, \quad \alpha_0 + \sum_{l=k+1}^{K} n_l\right)$$

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

Updating the cluster assignment c_i: Unlike TMoG, updating c_i now involves the rejected proposals asssociated with observation i and has conditional distribution

$$p(c_i = k|C^{\neg i}, X, Y, \theta, \alpha_0, \theta_0) \propto \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \prod_{r=1}^{R_i} \mathcal{N}(y_{ir}|\mu_k, \Sigma_k). \tag{4.18}$$

This is a consequence of the fact that the cluster assignment is made only once, after which proposals are made from that cluster until acceptance. This also accounts for the fact that without the rejected proposals, this distribution would be proportional to $\pi_k \mathcal{N}_{\mathbb{S}}(x_i|\mu_k, \Sigma_k)$ and would involve the intractable normalization constant of component k. Imputing the rejected proposals avoids having to calculate this quantity, though now the associated rejected proposals are transferred along with that observation to a new cluster.

If π were marginalized out (e.g. under a Chinese Restaurant Process), the cluster update rule becomes similar to that of the update equations found in Chapter 2 as follows:

$$p(c_i = k|C^{\neg i}, C^{\neg i*}, X, Y, \theta, \alpha_0, \theta_0) \propto n_k^{\neg i} \mathcal{N}(x_i|\mu_k, \Sigma_k) \prod_{r=1}^{R_i} \mathcal{N}(y_{ir}|\mu_k, \Sigma_k) \tag{4.19}$$

for seen k and

$$\propto \alpha_0 \int \left(\mathcal{N}(x_i|\mu_k, \Sigma_k) \prod_{r=1}^{R_i} \mathcal{N}(y_{ir}|\mu_k, \Sigma_k) \right) \mathrm{NIW}(\theta_k|\theta_0) d\theta_k \tag{4.20}$$

for unseen k. Here $C^{\neg i*}$ refers to all rejected proposals except those associated with i.

Updating the cluster parameters (μ_k, Σ_k): For the same reason as above, updating cluster parameters also requires conditioning on the imputed samples. The conditional distribution is identical to that for TMoG:

$$p(\mu_k, \Sigma_k|\theta^{\neg k}, X, Y, C, \alpha_0, \theta_0) \propto \mathrm{NIW}(\mu_k, \Sigma_k|\theta_0) \prod_{i \in C_k} \left(\mathcal{N}(x_i|\mu_k, \Sigma_k) \prod_{y \in Y_i} \mathcal{N}(y|\mu_k, \Sigma_k) \right).$$

With a Normal-Inverse-Wishart prior for the Gaussian mixture model, sampling from this distribution is easy. As with TMoG, the posterior distribution is also a NIW with a slight difference in notation, where now $m_k = \sum_{i \in C_k} R_i$:

$$p(\mu_k, \Sigma_k | \theta^{\neg k}, X, Y, C, \alpha_0, \theta_0) = \mathrm{NIW}(\mu_k, \Sigma_k | \mu_n, \lambda_n, \Phi_n, \nu_n) \tag{4.21}$$

$$\mu_n = \frac{\lambda_0 \mu_0}{\lambda_0 + (n_k + m_k)} + \frac{(n_k + m_k) + \bar{z}}{\lambda_0 + (n_k + m_k)} \tag{4.22}$$

$$\lambda_n = \lambda_0 + (n_k + m_k) \tag{4.23}$$

$$\nu_n = \nu_0 + (n_k + m_k) \tag{4.24}$$

$$\Phi_n = \Phi_0 + S + \frac{\lambda_0 + (n_k + m_k)}{\lambda_0 + (n_k + m_k)} (\bar{z} - \mu_0)(\bar{z} - \mu_0)^T, \tag{4.25}$$

and similarly,

$$\bar{z} = \frac{1}{n_k + m_k} \left(\sum_{i \in C_k} \left(x_i + \sum_{y \in Y_i} y \right) \right) \tag{4.26}$$

$$S = \sum_{i \in C_k} (z_i - \bar{z})(z_i - \bar{z})^T, \quad z_i \in \{x_i \cup Y_i\}. \tag{4.27}$$

This completes the entire Gibbs sampling step for MoTG.

4.5 Reducing MCMC variance

In our experiments, we see that despite their exactness, these MCMC algorithms come at the cost of a high variance, especially in higher-dimensional settings. At a high level, this can be attributed to the nonparametric prior over q being too flexible and placing too much probability mass on proposal distributions q that themselves place significant probability mass outside of $\mathbb{S}$. The observations X govern how q assigns probability within $\mathbb{S}$, characterizing the distribution p that we care about. Right outside the boundary of $\mathbb{S}$, the observations X indirectly constrain q because of prior smoothness assumptions. As we move away from $\mathbb{S}$, the influence of the observations starts to wane, and the posterior over q reverts back to the prior. In the part of $\mathbb{X}$-space away from $\mathbb{S}$, the MCMC sampler explores what is effectively the prior distribution over q by simulating rejected proposals Y from q and then simulating q from its posterior distribution given these rejections. The resulting coupling between Y and q can cause the MCMC chain to mix very poorly, and the estimate of the distribution over q can have large variance.

To understand the mechanics of this more clearly for mixture models, observe that as the MCMC algorithm explores parameter space, a cluster will occasionally be produced under which the subset $\mathbb{S}$ has low probability. Producing an accepted proposal from this component can require a large number of rejected proposals. As a consequence, the MCMC algorithm will experience a slowdown, taking a long time to move through an MCMC iteration, with the increased number of rejected proposals also increasing coupling between MCMC iterations. The large number of rejected samples will start to swamp out the observations X, causing the cluster parameters to drift away from the observations and $\mathbb{S}$, further exacerbating the issue. The net effect is longer computation times and larger variances in the MCMC estimates. Since MoTG proposes from a particular Gaussian component until acceptance (unlike TMoG, which picks a new component for each proposal), we expect that this effect is worse for MoTG than TMoG.

An obvious remedy is then to choose the hyperparameters of the nonparametric prior to concentrate on qs with high probability around $\mathbb{S}$. Setting parameters in this manner is not easy, however, especially with complex, asymmetric constraint sets such as those shown in Figure 3.1.

We instead take a more direct approach. At a highlevel, rather than setting the hyperparameters of the DP prior over q, we aim to allow the practitioner to use fairly default settings. Instead, their judgement will be to set a parameter $\rho \in (0, 1)$, corresponding to the belief that q is restricted to distributions satisfying $q(\mathbb{S}) = \rho$. Small values of ρ imply that q assigns relatively large probability outside $\mathbb{S}$, resulting in a large number of rejected samples. This is useful when the constraint set $\mathbb{S}$ is complex and one expects significant probability mass at the edges. Values close to one imply beliefs that the proposal distribution need not be too complex outside. Note that since we use a nonparametric prior over q, one does not have to be too careful about ρ. In fact, we expect that for any setting of ρ, the posterior is asymptotically consistent provided a reasonable prior is placed over the variance of the mixture components (see e.g. Canale and De Blasi (2017)). ρ however can have a significant impact on mixing and finite-sample performance. From our experiments, we recommend $\rho = 0.5$ as a reasonable default.

Write $\mathcal{Q}_\rho$ for the space of probability measures that assign probability ρ to $\mathbb{S}$. Working with these as priors over q is closely related to the approach outlined in Kessler et al. (2015), where the authors are interested in nonparametric Bayesian priors given specifications of certain marginal distributions; in our setting, we specify that the marginal probability of $\mathbb{S}$ equals ρ. Following that work, posterior simulation would have to be approximate, involving kernel density estimates of marginal probabilities under the original prior. Unlike that work which considers general marginal specifications, we have a simple and specific requirement: any q simulated from the nonparametric prior (whether TMoG or MoTG) must satisfy $q(\mathbb{S}) = \rho$. We instead propose a much simpler approximation strategy, one that becomes more accurate as the number of observations increase. As before, the Gibbs sampler involves three steps:

Imputing the rejected proposals Y: The fact that $q(\mathbb{S}) = \rho$ means that the number of rejected samples is geometrically distributed with success probability ρ. This, and the fact that they are exchangeable (Rao et al., 2016), allows a simple simulation approach: first sample the number of rejected samples from the geometric distribution with parameter ρ, and then use rejection sampling to sample their locations outside $\mathbb{S}$.

Updating cluster parameters β and weights w: This step presents a challenge, since we need to update these while satisfying the constraint $q(\mathbb{S}) = \rho$. However, we observe that for reasonably large datasets, the likelihood allows us to drop this constraint in the prior without introducing too much error. In particular, with n observations, our imputation scheme introduces $\frac{1-\rho}{\rho}n$ rejected samples. For large n, standard consistency results for DP mixture models tell us that the posterior will concentrate around qs that satisfy $q(\mathbb{S}) = \rho$. Since we are really interested in $p(x) \propto \mathbb{1}_{\mathbb{S}}(x)q(x)$, any deviation of $q(\mathbb{S})$ from ρ is renormalized out, and introduces minimal error into our final analyses. Our experiments confirm this.

Updating cluster assignments C: Having instantiated the proposal distribution q through its parameters $\theta = (\beta, w)$, updating the cluster parameters is identical to corresponding steps in the original TMoG and MoTG samplers.

4.6 Additional simplifications

In the original MCMC samplers, to impute rejected proposals for an observation, proposals are made until one is accepted. In the previous section, to restrict ourselves to proposal distributions satisfying $q(\mathbb{S}) = \rho$, we simulate a geometrically distributed number of rejections, recognizing the added fact that now $q(\mathbb{S}) = \rho$. For a dataset with n observations, the average number of rejected proposals is $\frac{1-\rho}{\rho}n$. For a reasonably large n, the law of large numbers allows us to approximate the number of rejected samples with its average, avoiding the need for any stochastic simulation and simplifying the algorithm.

Secondly, in many settings, the practitioner might be conservative choosing ρ, so that the number of rejections from the geometric distribution *exceeds* that produced by the original algorithm. To avoid this undersirable inefficiency, we suggest a further simplifications, changing the first step of the Gibbs sampler to:

Imputing the rejected proposals Y: simulate rejected proposals from q outside $\mathbb{S}$ until $n\frac{1-\rho}{\rho}$ or n acceptances are produced, whichever occurs first.

Effectively, we are changing our marginal specification, restricting our nonparametric prior to q's satisfying $q(\mathbb{S}) \geq \rho$ (instead of the strict equality from before). We repeat

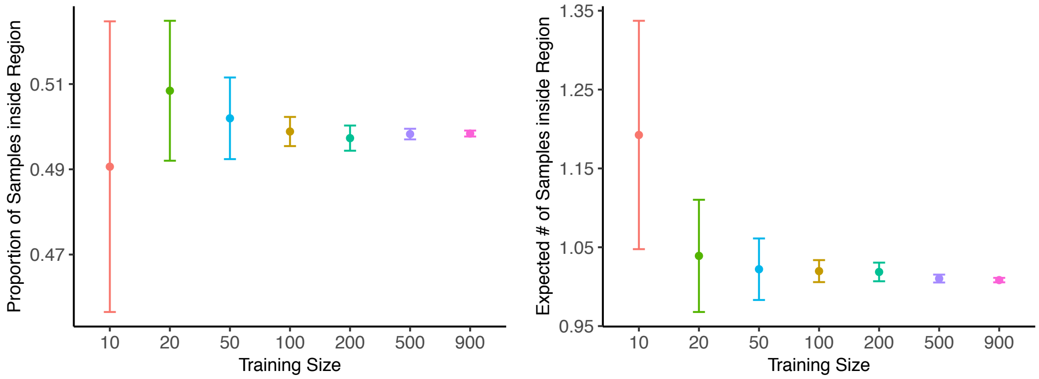


Figure 4.1.: Monte Carlo estimates of the proportion of samples inside $\mathbb{S}$ (left) and the expected number of samples inside $\mathbb{S}$ (right) for a threshold of 1 on TMoG model (MoTG shows a similar result), corresponding to $\rho = 0.5$. Shown in the figure are the mean and ±1 standard deviation (as the error bar), of the values for every training set size.

that changing the marginal specification in this way does not affect the asymptotic properties of our nonparametric model. For finite number of observations, a small value of $\frac{1-\rho}{\rho}$ simplifies computation and improves MCMC mixing. Making this too small can introduce edge effects as not enough observations are produced outside $\mathbb{S}$, hurting the ability to create large density values at the boundaries.

In our experiments, we consider different settings of this threshold, setting $\frac{1-\rho}{\rho}$ to 0.5, 1, 5, and 50. We refer to the final simplified sampler as a *thresholded sampler*. Chapter 5 studies the trade-offs involved for different threshold settings and the performance of the algorithm in terms of time and the predictive likelihood for various simulation studies. Our experiments suggest a threshold of 1, corresponding to $\rho = 0.5$, is most useful for typical settings.

When updating the cluster parameters in the Gibbs sampling step, our sampler does not enforce the restriction we placed on the prior of q. For this reason, our thresholded sampler is an approximate MCMC sampler. We argue that the standard consistency result guarantees a small approximation error with large sample sizes.

We show this empirically in Figure 4.1 by plotting the Monte Carlo estimates of $q(\mathbb{S})$ as the number of observations is increased. We used the MCMC samples from our experiments in Section 5.5 to draw observations from the posterior distribution and calculated the proportion and expected number of draws that are in S. We see that the mean proportion converges to 0.5 and, consequentially, the expected number of samples converges to 1, as training set size increases. This is consistent with our chosen threshold of 1 which corresponds to $\rho = 0.5$. This verifies that our sampler, although an approximation, has small error as long as the data set size is greater than 100.

4.7 Summary

We outlined a data augmentation scheme based on the rejection sampling algorithm and showed how this mechanism can be incorporated in a Gibbs sampler. We present the conditional distributions for the update steps in the Gibbs sampling methods with data augmentation that are associated with our two proposed models, TMoG and MoTG. Additionally, we showed how each sampler can be used in both finite and infinite mixture models. Both of the methods we outlined described the exact sampler that allows for infinite draws from the candidate distribution q. While this can estimate p with high accuracy, we show in the experiments that it is computationally ineffecient.

We discussed the drawback of the exact MCMC sampler and proposed modifications to improve the algorithm. We started by introducing ρ, corresponding to the weight given to the region $\mathbb{S}$ by the proposal q, so that $q(\mathbb{S}) = \rho$. We explained that the number of rejections in the rejection sampling scheme is geometrically distributed with acceptance probability of ρ. Our final approach, the *thresholded sampler*, simulates from a model requiring $q(\mathbb{S}) \geq \rho$. This translates to an acceptance probability that is greater than or equal to ρ, with mean less than $\frac{1-\rho}{\rho}$. Our *thresholded sampler* allows practitioners to choose the mean number of rejections per observations, which we call *thresholds*. If the threshold is 5, which corresponds to $\rho = \frac{1}{6}$, and n is 100, then we would expect at most $n\frac{1-\rho}{\rho} = 500$ number of rejections produced by the sampler. In Chapters 5 and 6, we show that through experimental results on a variety of synthetic and real datasets, our approach improves mixing and increases the efficiency of the original algorithm.

5. EXPERIMENTS ON SYNTHETIC DATA

5.1 Introduction

We applied the models introduced in Chapter 3 and their associated MCMC algorithms with the improvements described in 4 to synthetic datasets. This simulation study includes both simple and complex constraints, in one-dimensional as well as two-dimensional settings. We studied the modeling and computational trade-offs between the truncated mixtures of Gaussians (TMoG) and the mixtures of truncated Gaussians (MoTG) models as well as between different settings of ρ (or equivalently, different thresholds for the number of rejections). Each of these experiment provides a good assessment of the models performance on a variety of different setting.

We repeated each experiment 100 times, plotting the median and 25% and 75% quantiles for both test log-likelihood and compute time. Our MCMC samplers used a total of 5000 iterations with the first 2000 discarded as burn-in. In all experiments, we used a stick-breaking prior truncated to 50 components with concentration parameter set to 1.

For most studies, we generated datasets with 500 observations and evaluated performance by using 80% of the data as training and the remaining 20% as held-out test data. We evaluated the test-likelihood using importance sampling, verifying these numbers with numerical integration when possible.

We considered 6 different settings of ρ corresponding to threshold values of 0, 0.5, 1, 5, 50, and infinity. Here, 0 means no data augmentation, so that the data are modeled with an unconstrained mixture model that is not cognizant of truncation boundaries. We will see that this can result in inappropriately low probability at the boundaries of the constraint set as the mixture model tries to explain the absence of any observations outside $\mathbb{S}$. A threshold of infinity means no thresholding, corresponding to our exact MCMC algorithms. The remaining settings correspond to our proposed *thresholded sampler*, which limit the maximum number of rejections per observation, corresponding to progressively smaller bounds on $q(\mathbb{S})$.

5.2 Truncated Gaussian distributions on the unit interval

We start with a simple one-dimensional setting where the constraint set $\mathbb{S}$ is the unit interval. This does not really require our methodology since all normalization constants can easily be calculated numerically, or since one might choose a different modeling approach like a mixture of Beta distributions. Nevertheless, this provides a simple test case to exactly evaluate the effect of different modeling and algorithmic choices. In higher dimensions (such as the unit square or hypercube), the limitations of standard methods become more evident and our methodology will be necessary to capture correlation structure as well as high probabilities at the edges. We consider two datasets, one drawn from a truncated Gaussian centered at the midpoint of the interval (in particular, $\mathcal{N}(0.5, 0.1)$), and one at the edge (in particular, $\mathcal{N}(0, 0.1)$). We model these datasets using TMoG and MoTG, placing on the components a Normal-Inverse-Gamma prior given by

$$\mu|(\sigma^2, \mu_0, \lambda_0) \sim \mathcal{N}(\mu_0, \sigma^2\lambda_0), \qquad \sigma^2|(\alpha_0, \beta_0) \sim \text{Inv-Gamma}(\alpha_0, \beta_0). \tag{5.1}$$

We set the mean parameter μ_0 to the true mean (0 or 0.5) and set parameters λ_0, α_0 and β_0 to $2, 2$, and 0.05 respectively.

We present the results for TMoG on the first dataset in Figure 5.1 (the results of MoTG are almost identical). The left panel plots the posterior predictive density given the observations for all threshold settings, and we can see no noticeable differences here. The right-panel quantifies this, plotting the log-likelihood of the test dataset (on the y-axis) against the time taken to run 100 iterations. Again, we see no differences in the predictive log-likelihood or run-times.

The lack of sensitivity to thresholding stems from the simplicity of the problem where the true data generation mechanism involves a Gaussian density, most of whose

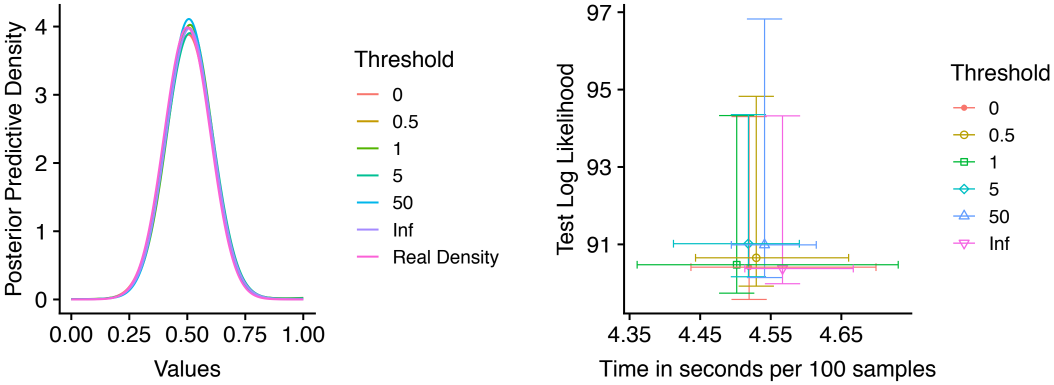


Figure 5.1.: Posterior predictive density (left) and speed-accuracy performance (right) of MoTG on data from Gaussian $\mathcal{N}(0.5, 0.1)$ with varying thresholds on the number of rejected proposals. The performance plot shows the 25% and 75% quartiles along x- and y-axes.

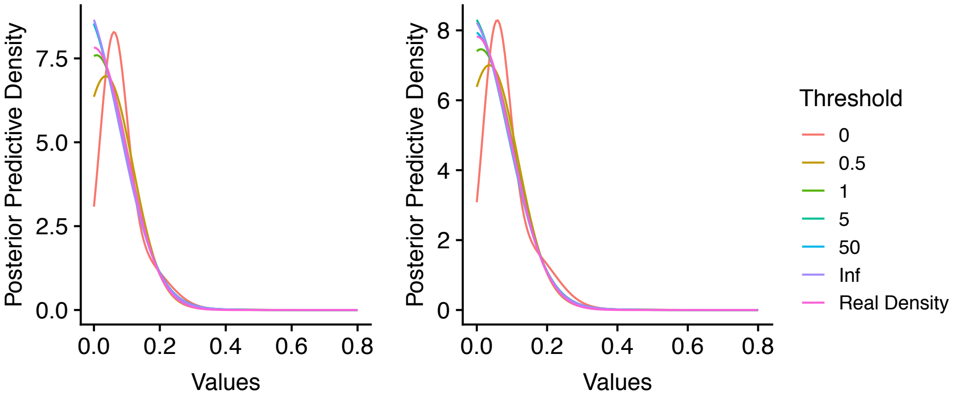


Figure 5.2.: Posterior predictive density for MoTG (left) and TMoG (right) for different threshold settings. The settings $0, 0.5$ and 1 clearly understimate the density at the origin.

mass is restricted to within the constraint set. Our algorithm recovers this fact so that the inferred proposal distribution q closely approximates the generating Gaussian, resulting in the number of rejected samples rarely hitting even the smallest threshold level.

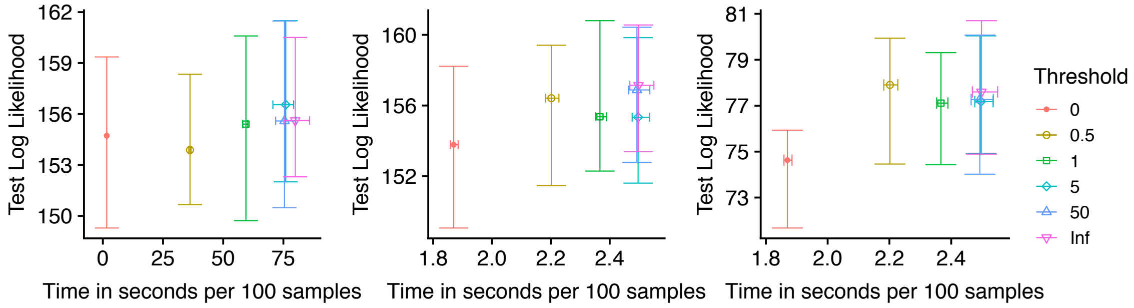


Figure 5.3.: Speed-accuracy performance for MoTG (left) and TMoG (center) for $\mathcal{N}(0, 0.1)$. The rightmost panel shows TMoG with test data biased towards the edges.

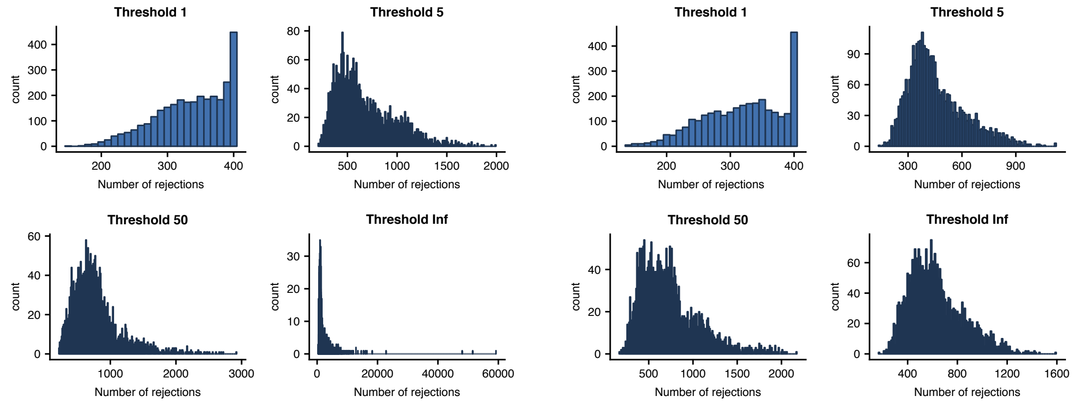


Figure 5.4.: Histogram of the number of rejections at every iteration for MoTG (left) and TMoG (right) for the truncated $\mathcal{N}(0, 0.1)$ Gaussian on the unit interval.

The second dataset presents a more challenging problem with the mode located at the boundary. Now, thresholding does play a role, with the fit becoming increasingly accurate as rejected proposals are incorporated. Figure 5.2 shows the mean predictive density for MoTG (to the left) and TMoG (to the right). With low thresholds (and especially when equal to 0) the posterior predictive experiences a clear shift away from the left boundary; despite the fact that we use a flexible mixture of Gaussians to model this distribution. Essentially, the model struggles to reconcile the abrupt change in the number of observations across the boundary and compromises by smoothing across the boundary, resulting in a moderate (rather than high) density at the edge. While it might be possible to try to get around this using a prior allowing very peaked components, it will not account for the smoothness of the density inside the interval. This emphasizes the importance of accounting for boundary effects in modeling constrained data.

Figure 5.3 presents the speed-accuracy trade-off for different threshold settings, each plotting the log-likelihood of the test dataset on the y-axis against the run-time for 100 iterations on the x-axis. Results for MoTG are to the left and TMoG to the middle. As expected, we see an increase in run-time as the threshold increases, though there are no significant differences among threshold larger than 1 (indicating that these thresholds are never reached). Interestingly, the qualitative degradation resulting from setting the expected number of rejections to 0 does not manifest itself quantitatively, with no difference in test performance for different settings of the threshold. This is partly because not enough test observations lie close to the edge to significantly affect the log-likelihood. We will see such effects later with sharper densities and higher-dimensional settings. If however, we bias the test dataset to favor observations near the edges, we see a clear drop in performance when the threshold is set to 0. Here, we selected the test set from observations in the interval $(0, .05)$.

In terms of efficiency, we find that TMoG runs significantly faster than MoTG. This is a result of the generative process of MoTG where having picked a cluster, one must repeatedly sample from it until an acceptance. When the chosen cluster assigns low probability to $\mathbb{S}$, we get a large number of rejections before acceptance. This can also initiate a runaway event where the rejected proposals (that lie outside $\mathbb{S}$) draws the cluster away from $\mathbb{S}$ while increasing its mixture selection probability, resulting in even more rejections. Figure 5.4 demonstrates this by plotting histograms of the number of rejected proposals for different threshold settings. When this threshold is infinite, the number of rejected samples is more than an order of magnitude larger for MoTG.

With regards to the diagnostic of the sampler, we computed the effective sample size (ESS) of MCMC results for both samplers by storing the parameter value of the highest weighted component. For MoTG, the effective sample size is highest for threshold 0.5 at 660, followed by threshold 1 at 132. Similarly, the ESS for TMoG is highest for threshold 0.5 at 410, followed by threshold 1. For both models, all other thresholds have ESS of less than 50; further emphasizing the need for a *thresholded sampler*.

5.3 Beta Distribution

The previous section suggests that the number of auxiliary rejected samples need not be much larger that the observed dataset. In fact, we recommend fixing $\rho = 0.5$ (corresponding to a threshold of 1) for typical settings. Our next experiment, while still relatively simple, considers the situation when the true density lies outside the model class. Now we used TMoG and MoTG to model observations from a $\text{Beta}(0.1, 0.1)$ on the interval $[0, 1]$. This parameter setting results in sharp modes at each end of the interval. We still use the Normal-Inverse-Gamma prior as in Equation (5.1) with parameters: $\mu =$ sample average of the data, $\lambda = 2$, $\alpha = 2$, and $\beta = 0.01$.

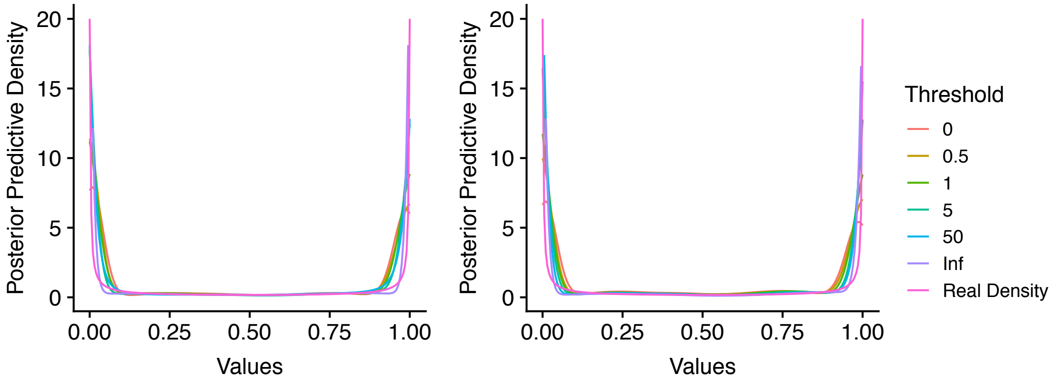


Figure 5.5.: Posterior mean density of MoTG (left) and TMoG (right) for data from a $\text{Beta}(0.1, 0.1)$.

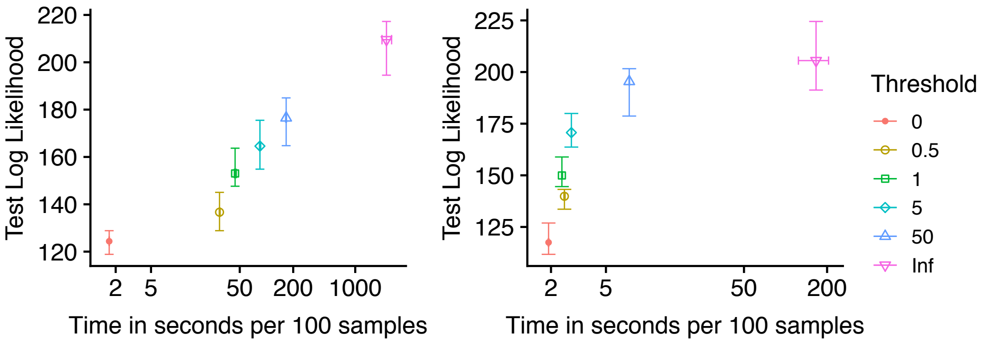


Figure 5.6.: Speed-accuracy plot for various thresholds for MoTG (left) and TMoG (right) for data from Beta(0.1, 0.1)

Figure 5.5 plots the estimated densities for both MoTG and TMoG. Similar to the previous experiment, we see that if the truncation is too strong, the model fails to adequately capture the peaks at the boundaries of the interval. Looking at the quantitative results in Figure 5.6, we see a clear monotonic improvement in test-likelihood for both MoTG and TMoG as more and more rejected samples are included. Now, capturing the sharp peaks at the edges of the boundary requires more rejected samples outside the interval. From this figure, it is clear that thresholding below 5 rejections per observation results in poor performance where the case with no thresholding performs best. Of course in practice, one does not know the true density. In such situations, one can use the number of observations near the edges as a guideline for setting the threshold. As an example, if a large proportion of the observations lies near the edges, then one might need a larger threshold. As before, this accuracy comes at a computational cost with larger thresholds having longer run-times. Again, each TMoG setting is significantly more efficient than its MoTG counterpart.

More over, the trace plot of the number of rejections in Figure 5.7 shows that the mixing in TMoG is better than its counterpart across all thresholding. In the exact case where no thresholding is enforced, both models shows poor mixing. As we limit the number of rejections through our proposed thresholded sampler, the MCMC

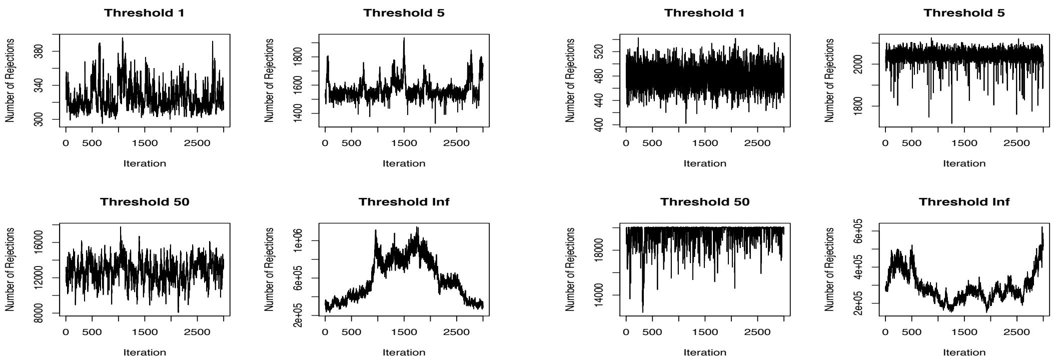


Figure 5.7.: Trace plot of the number of rejections for MoTG (left) and TMoG (right) on varying degree of thresholding.

mixing improved significantly. These results proved that the new sampler is able to reduce variance and increase efficiency.

In terms of the ESS of each sampler, we see a similar trend to that of the first experiment. Again, we examine the parameter estimates for the mean of the model on the highest weighted component. We then compute the ESS for the collection of means across all iterations. For TMoG, the ESS is highest at threshold 0.5 with a value of 1654 and decreases monotonically as the threshold is increased, reaching to only 3 where no thresholding is enforced. For MoTG, the ESS averages at 1935 across all thresholds. These results suggests that while the number of rejections are high for MoTG, the model were still able to produce independent samples.

Further, we examine whether the samplers for both models have converged by computing the shrink factor, $\hat{R}$, proposed by Gelman *et. al.* Gelman et al. (2013). To do this, we computed the between- and within-sequence variances of 5 individual runs with different initializations for all thresholds. We find that $\hat{R}$ is equal to 1 across all experiments, suggesting that the chains have converges.

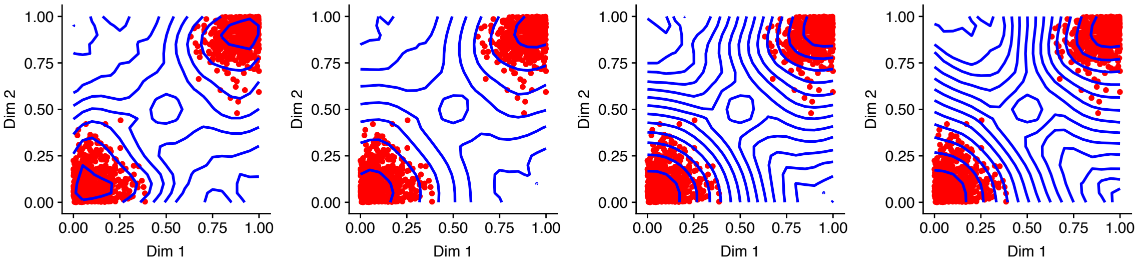


Figure 5.8.: Contour plot of log posterior mean density for data drawn from a mixture of two bivariate Gaussians. Thresholds are 0, 1, 5, and Infinity (left to right)

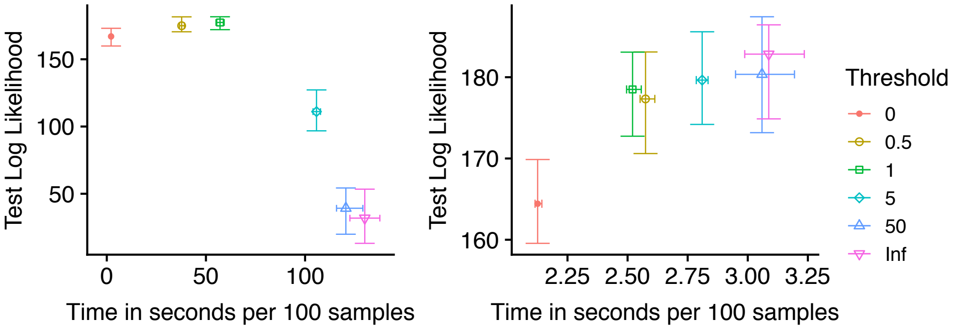


Figure 5.9.: Speed-accuracy plot for MoTG (left) and TMoG (right) for data from the mixture of two bivariate Gaussians

5.4 Mixture of Bivariate Gaussians in the unit square

Our next synthetic experiment considers data on the two-dimensional plane. Our constraint set $\mathbb{S}$ is the unit square $[0,1]^2$ with data generated from a mixture of two Gaussians, one centered at $(0,0)$ and the other diagonally across at $(1,1)$. We generate 1000 observations from this model, using 400 as training and 100 as test. For both TMoG and MoTG, we place Normal-Inverse-Wishart priors on the components given by

$$\mu|(\mu_0,\lambda,\Sigma) \sim N(\mu_0,\Sigma/\lambda), \quad \Sigma|(\Phi,\nu) \sim \text{Inv-Wish}(\Phi,\nu). \tag{5.2}$$

The parameters are set as follows: $\mu_0 = (0.5, 0.5)$, $\lambda = 1$, $\Phi = 0.01 I_2$, and $\nu = 4$, where I_2 is the 2-dimensional identity matrix. We show contour plots of the estimated densities for TMoG for different threshold settings in Figure 5.8. Visually, it is clear from the figures that there is an increasing accuracy in modeling the mode of the density at the corner of the constrained space and is consistent with the results seen on the previous experiments.

Figure 5.9, plotting test log-likelihood against compute time for different threshold settings, tells a more complicated story. As before, we see that TMoG is significantly faster than MToG, now by almost two orders of magnitude. Additionally, increasing the threshold parameter results in an increase in the run time. For TMoG, performance with a threshold of 0 is significantly worse than other settings and larger thresholds do results in significant improvement in performance.

Interestingly for MoTG, large settings of the threshold parameter result in a *decrease* in test-likelihood. Figure 5.10 shows that MoTG allows for observations to have a high number of rejections when the average number of rejections across all iterations is almost similar: 2894 for MoTG and 3899 for TMoG. Furthermore, Figure 5.11 shows that the variance of the highest weighted component for MoTG is exponentially higher than that of TMoG. These results illustrate the importance of controlling variance through the introduction of asymptotic bias in the MCMC algorithm. The effect is more noticeable in two-dimensions than in one because the probability of rejection increases with dimension. A threshold of 1 to 5 represents a reasonable compromise between too much bias and variance. This effect is much less noticeable for TMoG.

5.5 Mixture of Bivariate Gaussians in complex constraint

Lastly, we consider a significantly more complex 2-dimensional domain $\mathbb{S}$ shown in Figure 5.13. We randomly draw 500 observations from 3 bivariate Gaussians centered on the edges of an island. Because our algorithms are set to improve density estimation at the boundary of the domain, this simulation study investigates relative performance in this area. For this particular experiment, we tested the performance

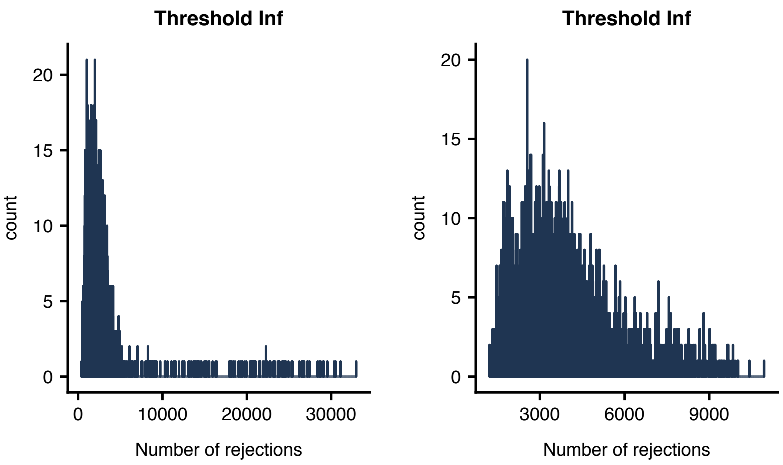


Figure 5.10.: Number of rejections on exact sampler for MoTG (left) and TMoG (right).

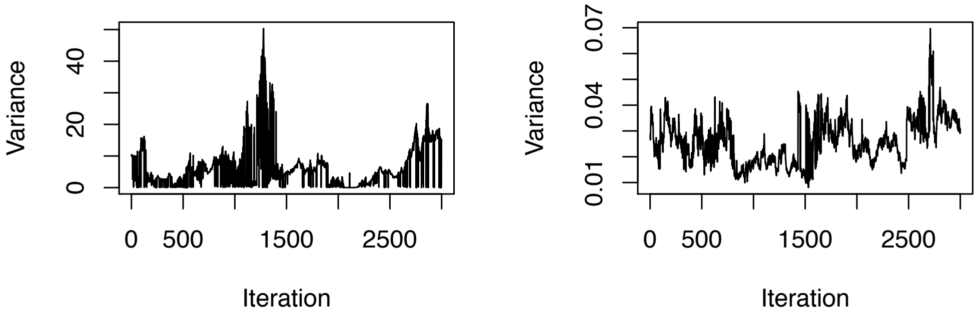


Figure 5.11.: The variance of the component with the highest weight for MoTG (left) and TMoG (right).

of the algorithms only on observations near the edges of the boundary. We scaled the data by 1.05, find observations that lie outside the boundary, use 100 of these observations as the test set, and randomly select 400 of the remainder as the training

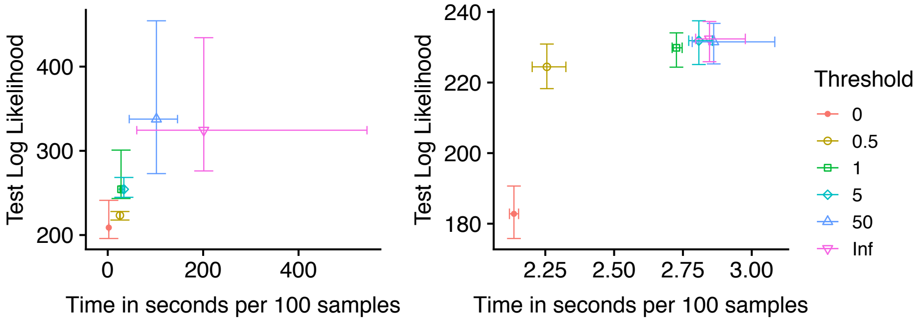


Figure 5.12.: Speed-accuracy plots for MoTG (left) and TMoG (right).

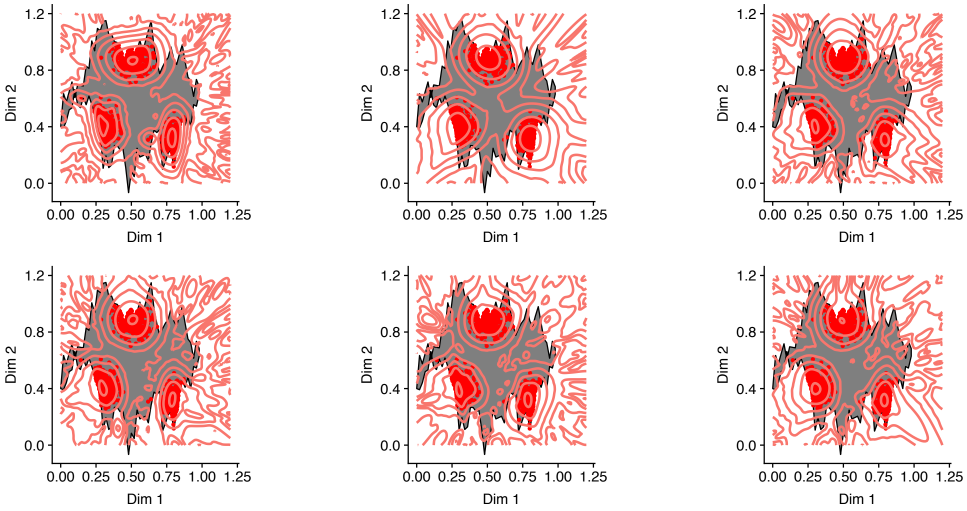


Figure 5.13.: Contour plot of log posterior mean density for TMoG for different ρ. Thresholds are 0, 0.5, 1, 5, 50, and Infinity (left to right, top to bottom)

set. We then run 5000 MCMC interations with a burn-in of 2000. For both TMoG and MoTG, we place Normal-Inverse-Wishart priors on the components as follows

$$\mu|(\mu_0, \lambda, \Sigma) \sim N(\mu_0, \Sigma/\lambda), \quad \Sigma|(\Phi, \nu) \sim \text{Inv-Wish}(\Phi, \nu). \tag{5.3}$$

The parameters of the NIW distribution are: $\mu_0 = (0.5, 0.5)$, $\lambda = 0.1$, $\Phi = 0.001I_2$, and $\nu = 4$, where I_2 is the 2-dimensional identity matrix. Figure 5.13 shows the estimated density for TMoG for different ρ, and again we see a steep drop in density outside the constraint set, which is likely to lower probability density at the boundaries. Figure 5.12 plots performance of TMoG and MoTG for varying ρ and again, TMoG is much faster. Further, TMoG's predictive performance plateaus at thresholding equal to 1 ($\rho = 0.5$), while MoTG requires higher thresholds. That a threshold of 1 does a good job can be guessed from the fact that unlike the beta distribution, observations are relatively well-dispersed near the boundaries. In the case where the test set were randomly sample through out the island, the performance of MoTG shows similarity to that of the bivariate case in a unit square in that, it decreases significantly as threshold increases. Further investigation towards the number of rejections and the variance of the components shows similar trends to that of Figure 5.10 and Figure 5.11.

Like before, we use the parameter estimates of the highest weighted component to compute the ESS of the samplers. The ESS for MoTG from lowest threshold to no threshold are: 433, 621, 279, 210, 22, and 433. Similarly for TMoG, the ESS are: 261, 137, 52, 52, 49, and 12. In general, MoTG consistently provides more independent samples than TMoG. In both models, a threshold of 0.5 results in the highest number of ESS. However, this specific threshold value results in insignificant performance in MoTG when compare to that of no thresholding. In these cases, a threshold of 1 corresponding to a $\rho = \frac{1}{2}$ should be a good compromise for both models.

In this example also, we see that the value of the test log likelihood estimate differ between the two models, even when using the same test observations. These differences are almost unnoticeable in our previous experiments and are most pronounce here due to the more complex bounds. This is consistent to our analytical findings in Section 3.5.

5.6 Summary

In this chapter, we evaluated our two models with the thresholded sampler on synthetic datasets. We contrasted the predictive performance as well as the efficiency of both samplers. In the first experiment which involves a simple one-dimensional

Gaussian, we show that using vanilla mixture models with no truncation can lead to inaccuracy in the density estimation at the edge of the support. Our second experiment with the beta distribution, revealed the need for higher thresholds to accurately predict the mode in this density. Moving on to two-dimensional data in our third experiment, we show that MoTG performs very poorly at higher thresholds due to the run away effect which leads to high variance. In our last experiment, we show that with bias test cases, both samplers perform well and produce significant differences in prediction when compare to the mixture models with no truncation.

Across all our experiments, we find that TMoG is consistently faster than its counterpart and more robust to the dimensionality of the data. We find that our thresholded sampler is effective in reducing computation time while maintaining the same relative degree of performance to that of the exact sampler. Additionally, this sampler prevents the run-away effect that is evident in higher threshold that in turns improve MCMC mixing. In the next chapter, we applied both samplers to real-world data.

6. APPLICATIONS TO REAL DATA

6.1 Introduction

In this chapter, we consider two real datasets, one of flow-cytometry data and one of homicide data in the city of Chicago. Our experiments with synthetic data suggest that MoTG is significantly more expensive computationally than TMoG, both in terms of the number of auxiliary variables produced as well as the mixing of the resulting MCMC algorithm. This is even more clear with the real datasets; in fact, for the crime data, we focus only on results from TMoG in this section.

6.2 Flow Cytometry data

We first consider a dataset of acute graft-versus-host disease (GvHD) flow-cytometry measurements from patients with bone-marrow transplants (Brinkman et al., 2007). GvHD can arise after receiving stem cells, when transplanted donor T-cells ("the graft") attack the patient's healthy organs ("the host"). The dataset from Brinkman et al. (2007) contains 6809 "control" and 9083 "positive" observations from 31 patients. We focus on the control observations, which are from patients who did not develop either acute or chronic GvHD after the transplant. Each observation involves measurements of 4 activation markers: CD4, CD8b, CD3, and CD8, with each measurement varying between 0 and 1024. The data collection process is such that observations outside this range are discarded, resulting in a sharp drop in intensity outside this set (see Figure 6.1 for projections of the raw data onto two-dimensional planes). We scale the data to lie in the four-dimensional unit hypercube, which forms our constraint set $\mathbb{S}$.

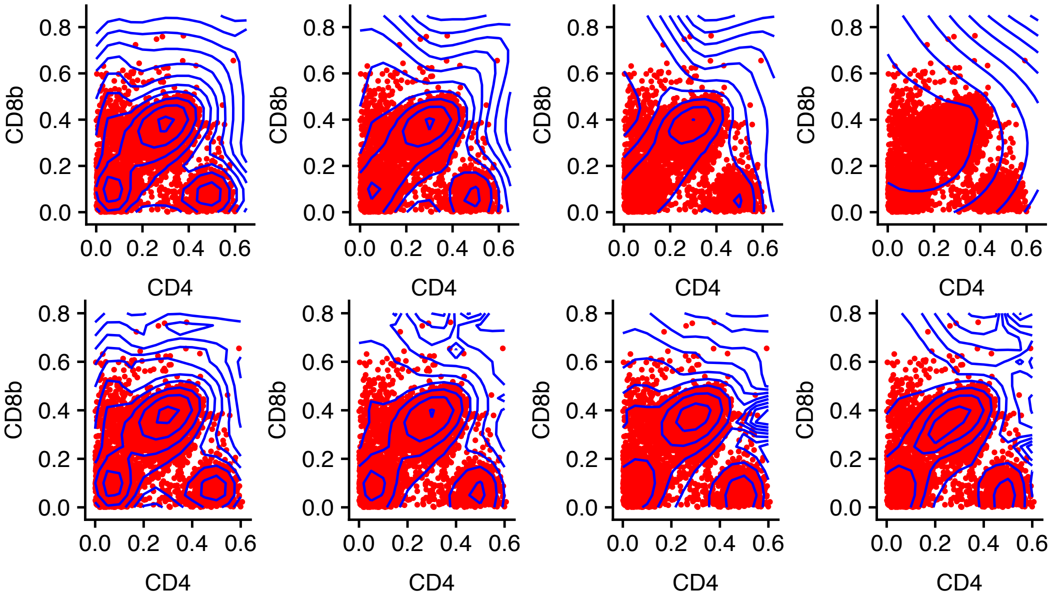


Figure 6.1.: Contour plots of the posterior mean distribution for MoTG (top) and TMoG (bottom) applied to flow-cytometry data. Subfigures are thresholds of 0, 1, 5, and 50.

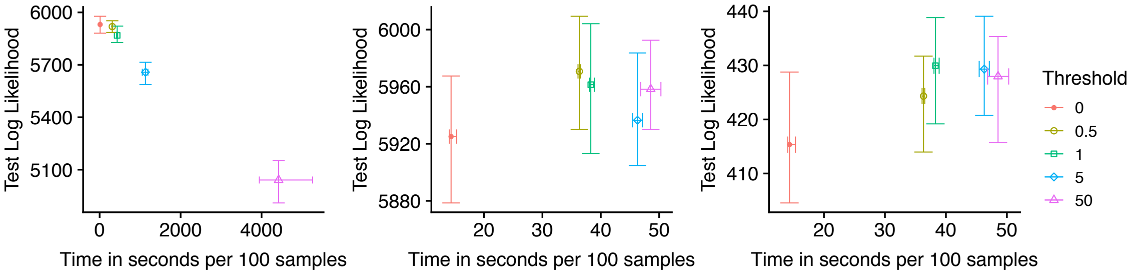


Figure 6.2.: Speed-accuracy performance of MoTG (left), TMoG (center), and TMoG with biased test set for flow-cytometry data

This dataset was briefly considered in Rao et al. (2016), where the authors demonstrated the feasibility of MCMC inference for TMoG. Here we analyze it more systematically, evaluating the performance of our two models, as well as different threshold settings. As before, in our mixture models we use the Normal-Inverse-Wishart as the prior, with parameters: $\mu_0 = 0.5$, $\lambda_0 = 0.01$, $\Phi = 0.001 I_4$, and $\nu = 5$. I_4 is the four-dimensional identity matrix.

Figure 6.1 shows contour plots of the predictive density of the models for different threshold settings, while Figure 6.2 shows how thresholding trades-off between predictive performance and compute time. For MoTG, we observe now that test-likelihood is no longer monotonic with the threshold setting, in fact, performance *worsens* as the threshold increases above 5. This is largely because now we are working with 4-dimensional observations, and without controlling the prior over $q(\mathbb{S})$, it is easy to explore elements that produce very large numbers of rejected samples. This can severely affect MCMC mixing.

In fact, we do not include the case where the threshold is set to infinity, since this occasionally produced a very large number of rejections that brought our simulations to a near halt. Figure 6.2 shows a steep drop in test log-likelihood for large thresholds, and in the contour plots of Figure 6.1, this manifests itself in the loss of multimodal structure.

This reiterates the point that even though a large threshold reduces bias, it can significantly increase variance, and that MoTG is particularly sensitive to this. As we described earlier, a significant driver of this large variance is poor mixing due to the large number of auxiliary variables. Figure 6.3 shows MCMC traceplots of the number of rejected samples over MCMC iterations for MoTG, and we see that other than for the low thresholds, the MCMC chain mixes very poorly. By thresholding the number of rejections, we are implicitly focusing on simpler model structure outside $\mathbb{S}$, resulting in improved MCMC performance.

The TMoG model runs much faster, and produces much better fit than their MoTG counterpart; this is clear qualitatively from Figure 6.1 and quantitatively from Figure 6.2. We do not see any significant changes in predictive performance with increasing threshold (although there is improvement in median performance). More importantly we do not see performance decay as the threshold increases, though without any thresholding, we did observe occasional MCMC iterations with a large

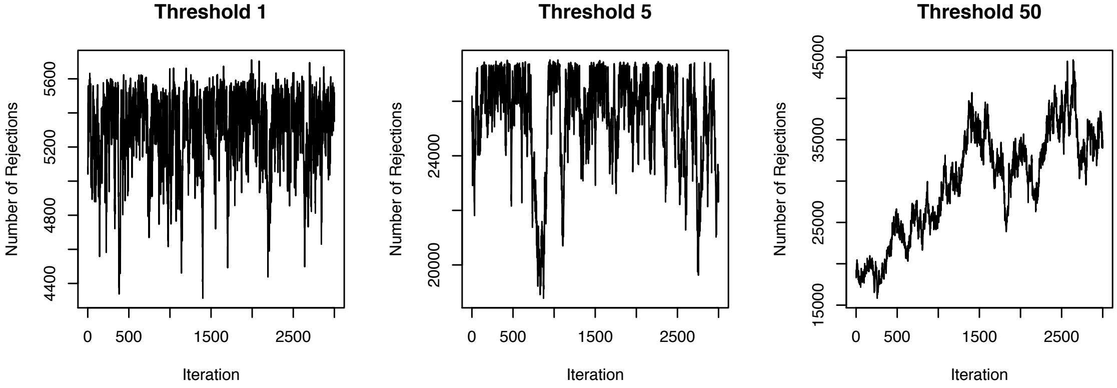


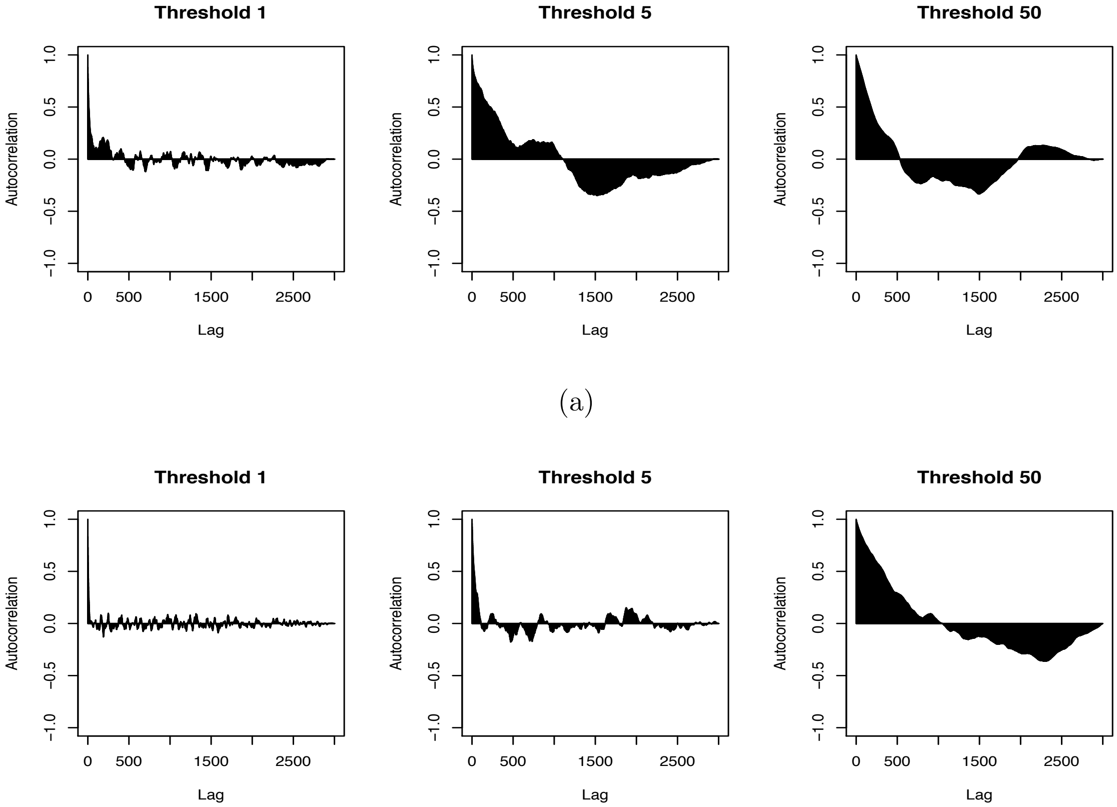
Figure 6.3.: MCMC trace plot of number of rejections for MoTG for flow-cytometry data

number of rejections. As before, this is because a large fraction of test samples lie away from the edges. A quick analysis of the dataset reveals that less that 5% of the observations lie within 0.01 of an edge, suggesting (as the figure verifies) that a threshold of 1 will be adequate. The rightmost subplot in Figure 6.2 focuses on performance at the boundaries, constructing the test set from observations within 0.01 of an edge. Now we see a slight improvement with threshold value. Figure 6.4 shows that autocorrelation of the number of rejections across iteration increases with threshold, and given the same threshold, that of MoTG is much worse. The autocorrelations of TMoG for threshold 1 does not extend beyond 10 to 15 iterations.

6.3 Chicago crime data

In our final experiment, we consider a dataset of criminal activity recorded in the city of Chicago. We gathered data from the city of Chicago data-portal[1], and plot it in Figure 3.1. Each recording in this dataset includes details such as case number, time, type of crime, as well as the longitude and latitude of the crime location. We restrict ourselves to modeling locations of homicide crimes occuring between the years 2012

[1] `https://data.cityofchicago.org/Public-Safety`



(a)

(b)

Figure 6.4.: ACF plot of the number of rejections for MoTG (top) and TMoG (bottom) on application to flow cytometry data

to 2017, resulting in 3220 observations. Thus, each observation is a measurement in a two-dimensional space where the longitude and latitude are the x- and y-coordinates respectively. The range of longitude value is from -87.8465 to -87.5316, while latitude range from 41.6479 to 42.0225. We rescaled these values to between -1 and 1.

Within the 2-dimensional Euclidean space, our constraint set $\mathbb{S}$ is the interior of the city of Chicago. To characterize this complex, non-convex set, we gathered data for the boundaries of the 77 neighborhood limits of Chicago[2], and approximated each

[2] `https://www.cityofchicago.org/city/en/depts/doit/provdrs/gis.html`

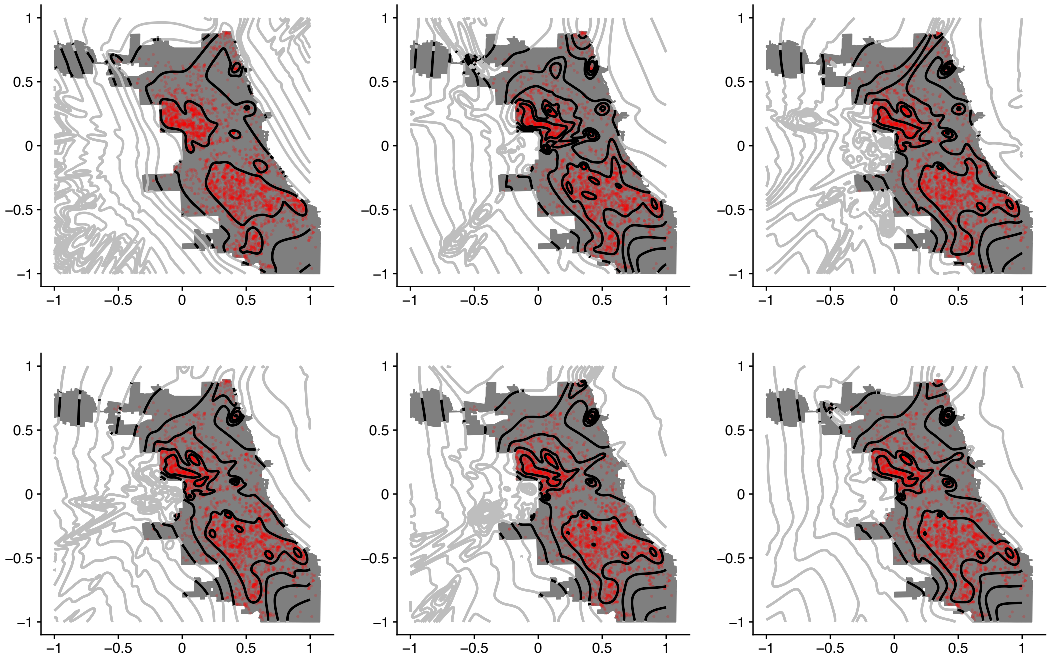


Figure 6.5.: Contour plots of the log posterior mean density for TMoG on the Chicago crime data with thresholds 0, 1, 5, 50, and Infinity (left to right, and then top to bottom).

neighborhood with a polygon using the `R` spatial polygon package, `SP`[3]. Combined together, these polygons formed the entire city limits. The package `SP` also allows to check whether a point lies inside a polygon. We used this function in our rejection sampling algorithm, to decide whether or not a proposal on $\mathbb{R}^2$ lies within Chicago.

We present results from modeling this data using TMoG in Figure 6.5. We do not include results from MoTG since, as indicated by our previous experiments, this takes much longer to run and produces results that 1) are much worse and 2) are sensitive to ρ. For our prior over cluster parameters, we used a two-dimensional Normal-Inverse-Wishart distribution with parameters $\mu_0 = (0, 0)$, $\lambda_0 = 0.1$, $\Phi = 0.001 I_2$, and $\nu = 4$. Figure 6.5 visualizes the posterior distribution through samples from the MCMC algorithm for different settings of the threshold parameter. Each subplot

[3] `https://cran.r-project.org/web/packages/sp/`

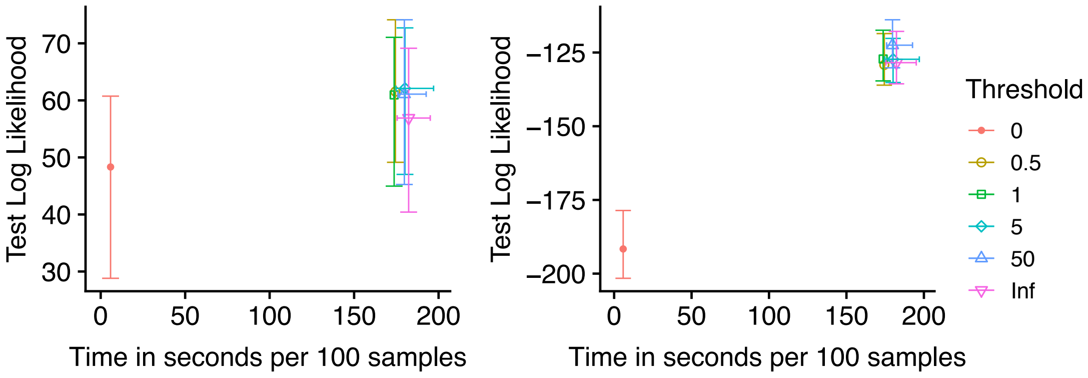


Figure 6.6.: Speed-accuracy plot of TMoG on Chicago test data. The left figure gives the performance on test data with random observations, the right figure gives the performance on test data with observations near the boundary.

shows the log of the posterior mean density for TMoG given the crime data, with the grey contour lines showing the estimated proposal distribution q, and the black lines highlighting them within the Chicago limits. The latter gives (up to a constant) the density of interest, whose properties can easily be estimated from the MCMC samples.

For all threshold settings, we observe two modes, one to the south of Chicago, and one to the west. When the threshold is set to 0, the range of density values near the boundary is smaller, being flattened to account for the absence of observations just outside the border. This misses many details near the boundary, for instance, there is a sharp cluster of observations right on the north-east boundary of Chicago which is lost for the threshold settings of 0. For larger threshold settings, the rejected proposals produce a significant cluster most of whose mass lies outside the city limits, but which overlaps with the city to allow a bump in probability at the corner. Edge smoothing-effects due to the constraint also result in coarser estimates at the mode to the west of the city.

An interesting phenomenon is the mild structure in the density contours away from the city boundaries. These are transients, resulting from the data-augmentation

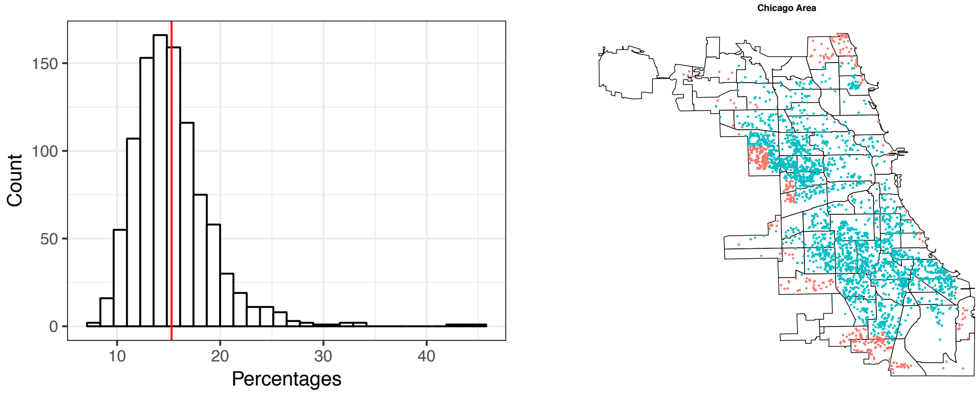


Figure 6.7.: Distribution of percentages near the boundary (left) from posterior predictive checks on 1000 MCMC samples for threshold 1. The vertical line at 15.28 gives the percentage of the real data with an estimated p-value of 0.456. Crime instances along the edges (right) of Chicago, that are prone to underestimation.

interacting with the thresholding. These do not affect inferences over the subset of interest, and are not relevant to the main estimation problem. Nevertheless, these represent an inefficiency in the thresholding procedure, and a waste of computational resources. A future research direction is to favor thresholding away from the subset $\mathbb{S}$ of interest.

Figure 6.6 quantifies the effect of threshold settings, plotting predictive performance of TMoG on held-out test data versus threshold. The left plot gives results for test data randomly sampled from the original dataset, and we see a slight, but not significant performance hit with no augmentation. The right subplot presents results when the test dataset is drawn from observations in Chicago neighborhoods touching the boundary. Now we see a significant loss of predictive performance without data-augmentation, providing quantitative justification of the importance of our data-augmentation scheme to accurately model probability structure near the edges of the constraint set.

As far as run-time is concerned, we see that a threshold of 0 is much faster that other settings, and that the average run-time does not increase significantly for larger thresholds. In fact, the relative inefficiency of these settings has little to do with our data-augmentation scheme. For instance, our recommended setting of TMoG, with threshold set to 1, produces on average around 2346 rejected proposals, which is less than a 50% increase in the original dataset. Instead, the increased run-time reflects our relatively crude approach to deciding if a proposal lies in the Chicago city limit. Our implementation, using the `R` package `SP`, needs to check that a proposal does not lie in any of the 72 Chicago neighborhoods before we can reject it. This accounts for the bulk of the run-time for non-zero thresholds (the vanilla mixture model without data-augmentation is unaware of the borders of the city and therefore does not include such checks). With a more careful implementation of the Chicago border, we can reduce this overhead.

To further validate our model, we used the MCMC posterior samples to run a posterior predictive check. For each MCMC sample, we generated the same amount of data (3220 observations) inside Chicago. We then scaled the data by a factor of 1.2 and computed the percentage that lie outside of Chicago. We compare the distribution of this quantity to the realized value in the observed dataset. Figure 6.7 plots both these quantities, with the observed value lying right in the middle of the predictive distribution. The resulting p-value (corresponding to the proportion of simulated values to the right of the observed quantity) was 0.46, suggesting a good fit.

In terms of the quality of the samples provided by the sampler, we computed the effective sample size for the estimated mean and variance of the highest component of the mixtures. We find that threshold 0.5 provide the highest ESS value at 50 for the mean and 32 for the variance of the first dimension. These values suggest an equilibrium of the simulated sequences has been achieved.

6.4 Summary

We applied our thresholded sampler to two real world datasets: flowcytometry measurements and homicides in Chicago. We show results that are consistent with our synthetic data in that for higher dimensions and higher thresholds, MoTG fails exhibits a high variance and fails to estimate the appropriate posterior density. On the other hand, TMoG consistently showed better results across all experiments and applications. The thresholded sampler of TMoG improves time efficiency as well as mixing.

7. SENSITIVITY ANALYSIS AND FURTHER MODIFICATIONS

7.1 Introduction

In Chapters 5 and 6, we tested our approach in various experimental scenarios and evaluated its performance in terms of predictive performance and time efficiency. In this chapter we assess our models' sensitivity to different hyperparameter settings of the stick-breaking prior. In addition, we explore a modification of our thresholded sampler which involves restriction on the boundaries of the rejected samples, and targets proposals that lie close to the target space $\mathbb{S}$. We show that this improves mixing of the Markov chain Monte Carlo algorithms while maintaining the accuracy and efficiency of the original algortihms.

7.2 Sensitivity to the DP prior

We conducted several more experiments to measure the effect of the choice of the prior on the performance of our samplers, specifically, the Dirichlet Process prior. We focus on one experiment, the bivariate Gaussian on a small island (Section 5.5). In Chapter 4, we discussed how ρ specifies the weight or probability given over the target region $\mathbb{S}$ when sampling from the proposal distribution $q(x)$. It is important to study the effect of the number of components provided by this proposal, which in turn affects that of p, to the predictive performance of the algorithm.

In an infinite mixture model, the number of components is controlled by the hyperparameter of the stick-breaking prior, α_0. Small α_0 values correspond to fewer components, allowing for densely populated clusters which can lead to smaller number of rejections. Larger α_0 produces greater number of non-empty clusters, causing the pool of rejections to come from a more diverse set of component parameters.

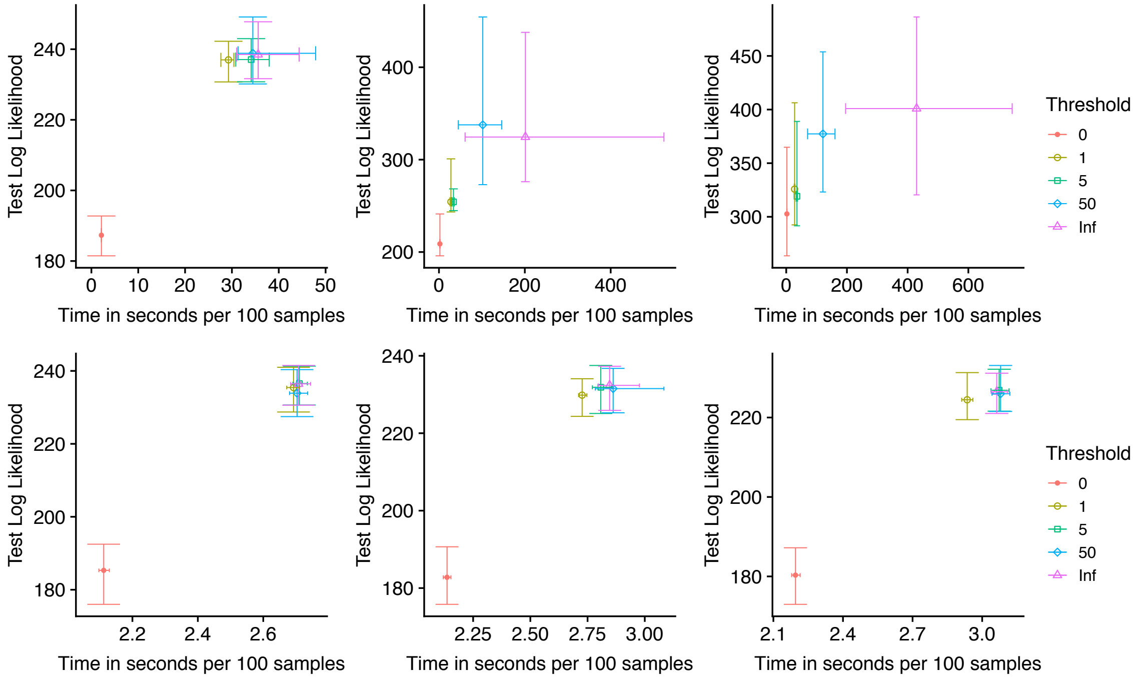


Figure 7.1.: Speed-accuracy plot for MoTG (top) and TMoG (bottom) for varying levels of α_0, from left to right, $\alpha_0 = 0.1, 1$, and 10.

Recall that in all our experiments, we set a symmetric prior with $\alpha_0 = 1$. In this section, we study the performance of our Gibbs sampler with α_0 values at $0.1, 1$, and 10, in the same experimental set up as mentioned in Chapter 5. Across all our experiments in this section, we omit threshold 0.5 as this produces result that closely resembles that of threshold 1.

Figure 7.1 compares the results for TMoG and MoTG in 3 separate sub-figures. We see that in terms of time, the results are consistent with our hypothesis in that smaller α_0 requires less computational time as fewer clusters are formed. The variation in the efficiency across α_0 is higher in MoTG, where there is almost a 10-fold change in accuracy and time moving from 0.1 to 1. In particular, the change in the hyperparameter of the DP highly effects the exact case of MoTG, as seen by its significantly longer run time (top right of Figure 7.1). This suggests that the generative

Table 7.1.: Estimated p-values from model checking of application to Crime data with varying degree of α_0.

Threshold α_0	0	1	5	50	Inf
0.1	0.017	0.011	0.073	0.118	0.011
1	0.433	0.456	0.612	0.621	0.537
10	0.091	0.036	0.063	0.093	0.169

process for MoTG causes it to be more sensitive to the prior. Unlike its counterpart, the TMoG model appears to be more robust to the choice of α_0.

In addition to looking at the effect of the concentration parameter of the DP, we also examine its effect on the model by running posterior predictive checks (Gelman et al., 2013). Here, we examine the MCMC samples of the application to the crime data from Section 6.3. Similar to the above experiment, we vary the sparsity of the components by setting α_0 to $0, 0.1$ and 10. for each posterior sample of the DPMM parameters from the Gibbs sampler, we generated 3220 observations and calculated the proportion that fall within the boundary of Chicago (see Figure 6.7). We do this with 3000 MCMC samples and computed the p-value of this distribution, presenting the result in Table 7.1. With $\alpha_0 = 0.1$, the table suggests that the model is inadequate for the data; except for thresholds 5 and 50, their p-values are mostly significant under the 0.05 cut-off. This result reinforces the need to use enough components that in turn will allow higher number of rejected draws to accurately model the data. Again, the choice of thresholding helps the model to fit the data better. On the other hand, higher number of clusters does not translate to a better model, as seen from results with $\alpha_0 = 10$. In general, with $\alpha_0 = 1$, the model performs well under the model checking diagnosis.

7.3 Constrained rejections

Rejected samples that are located too far from $\mathbb{S}$ slow computation and lead to the run-away effect. This is especially true in MoTG. Ideally, our samplers only draws proposals from regions close enough to the ambient space of $\mathbb{S}$ since only these are needed to estimate the density within $\mathbb{S}$. A natural approach is to restrict the area at which the sampler will be drawing proposals from. If the proposals are beyond a prespecified distance away from $\mathbb{S}$, we can ignore them in the sampling algorithm. We do this by imposing an additional boundary, $\mathbb{T}$, around $\mathbb{S}$.

Algorithm 4: An iteration of MCMC for posterior inference over $p(\theta|X)$

```
Data: The observations $X = \{x_1, \ldots, x_n\}$, and the current parameter values $\theta$
Result: New parameter value $\tilde{\theta}$
for each observation $x_i$ do
    while an accepted sample $\hat{x}$ has not been drawn do
        draw $y$ independently from $q(\cdot|\theta)$;
        check that $y$ is in $\mathbb{T}$;
    end
    Discard $\hat{x}$ and treat the preceding rejected proposals as $Y_i$;
end
Gather all the rejected samples, calling them $Y$: $Y = \bigcup_{i=1}^{n} Y_i$;
Update $\theta$ from $q(\theta|X, Y) \propto q(X, Y|\theta)p(\theta)$ using any MCMC kernel, calling the
  new value $\tilde{\theta}$;
Discard the rejected samples $Y$
```

When the sampler draws a proposal that lies between $\mathbb{S}$ and $\mathbb{T}$, we consider this draw as a candidate for our rejection sampling scheme. Alternatively, if a proposal lies outside of both $\mathbb{S}$ and $\mathbb{T}$, we ignore it altogether, and proceed to draw another proposal from the prior, q. Figure 7.2 illustrates this idea in more detail. The choice $\mathbb{T}$ should ideally be simple, such as a unit square or a circle around the perimeter of

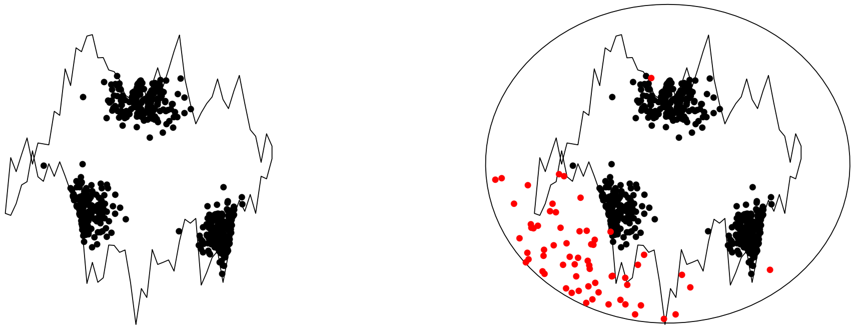


Figure 7.2.: Bivariate Gaussian data lying on a subset in a Euclidean space. The left pane shows synthetic observations, shown in black, lying in the space $\mathbb{S}$ specified by the solid black line. The right pane shows data with boundary $\mathbb{T}$ encompassing $\mathbb{S}$ as the constrained for rejected proposals in red.

$\mathbb{S}$, as to prevent additional overhead in computation time. This method requires only a slight modification to Algorithm 3. We summarize this in Algorithm 4. Theoretically, this modification truncates the proposal $q(x)$ to within $\mathbb{T}$. Further theoretical and experimental studies need to be conducted to fully understand the effect of this truncation to the posterior of $p(x)$. Results of experiments to simulated data are described in the next subsection.

7.3.1 Experiments

We evalaluated and tested the performance of this new modification on the synthetic data used in Section 5.5. Recall that the observations comes from a bivariate Gaussian distribution lying on an island. Here $\mathbb{S}$ is the island, and our choice of $\mathbb{T}$ is a circle surrounding $\mathbb{S}$ (see Figure 7.2). The radius of the circle is half of the longest span on $\mathbb{S}$ multiply by a tuning parameter v. Here, v ranges from 1 to ∞ where ∞ is

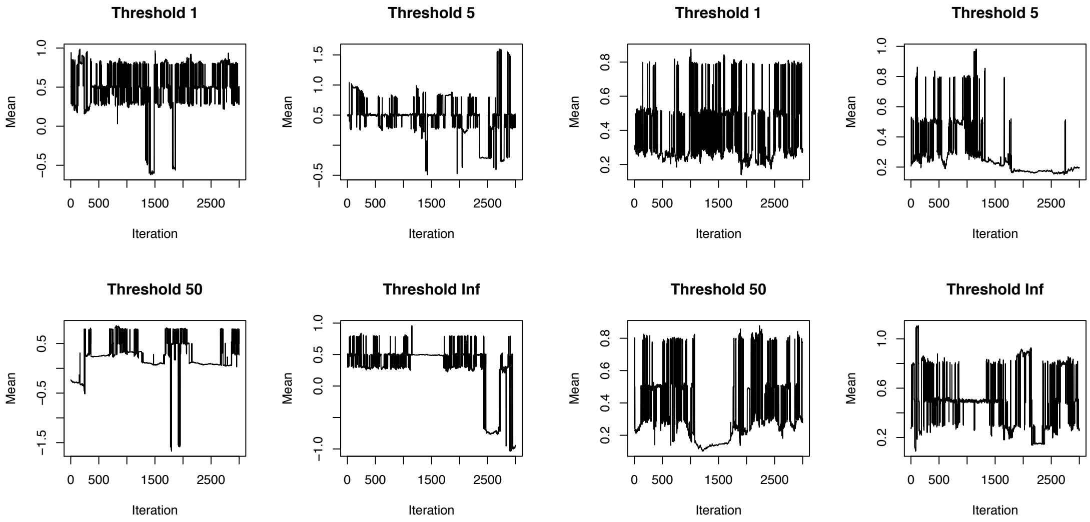


Figure 7.3.: Trace plot for the parameters of the most likely clusters without constrained rejections (left) and with constrained rejections (right) on TMoG.

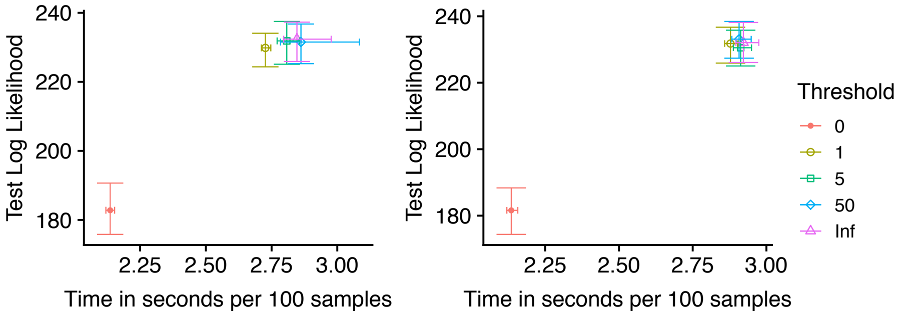


Figure 7.4.: Performance plot for method without constrained rejections (left) and with constrained rejections (right) on TMoG.

equivalent to our algorithm with no thresholding. We placed the same prior to that used in Section 5.5 and picked v as 1.1.

Figure 7.3 shows the traceplot of the component with the highest weight at every iteration. This figure suggests improvement in the sampler when $\mathbb{T}$ is imposed on the algorithm. Improvement can also be seen from the effective sample size of the Markov chain sampler. At threshold of 1, the original method produce an effective

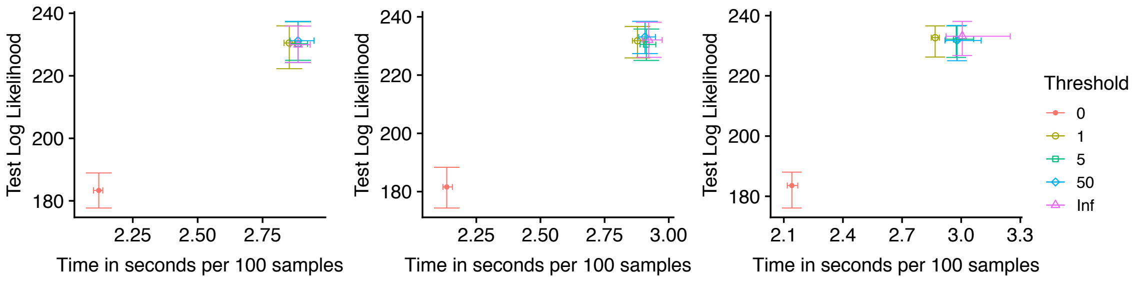


Figure 7.5.: Effect of varying size of v on speed and accuracy, with values of v from left to right: $1.05, 1.1$, and 1.5

sample size of 16. This number is increased to 139 with a bounded rejection space. The modification comes at no cost to the efficiency and accuracy of the algorithm. Figure 7.4 shows that the two methods produces comparable performance on 100 cross-validation data. The performance is also robust with respect to the varying size of v (see Figure 7.5). Although the result is promising, a deeper theoretical and empirical understanding is needed to fully comprehend how the restriction in the form of $\mathbb{T}$ affect the prior on $\mathbb{S}$.

7.4 Effect of sample size

In Chapter 4 we mentioned the effect of training size on the probability of the prior in $\mathbb{S}$. We showed that as the sample size increases, the proportion of the posterior samples in $\mathbb{S}$ converges to the true ρ. In this section, we explore the effect of training size with varying degree of thresholding, on the predictive performance of the samplers. Understanding its relationship can help practitioners decide on the appropriate thresholding based on the size of the data set. Again, we use the same synthetic data given in Figure 7.2, with identical hyperparameters and experimental setting to that of the experiments in Chapter 5. We vary the training set size from 10, 50, 100, 200, 300, and 500 by random sampling and ran the sampler with 100 cross-validations each. Figure 7.6 presents the result of the experiment for TMoG.

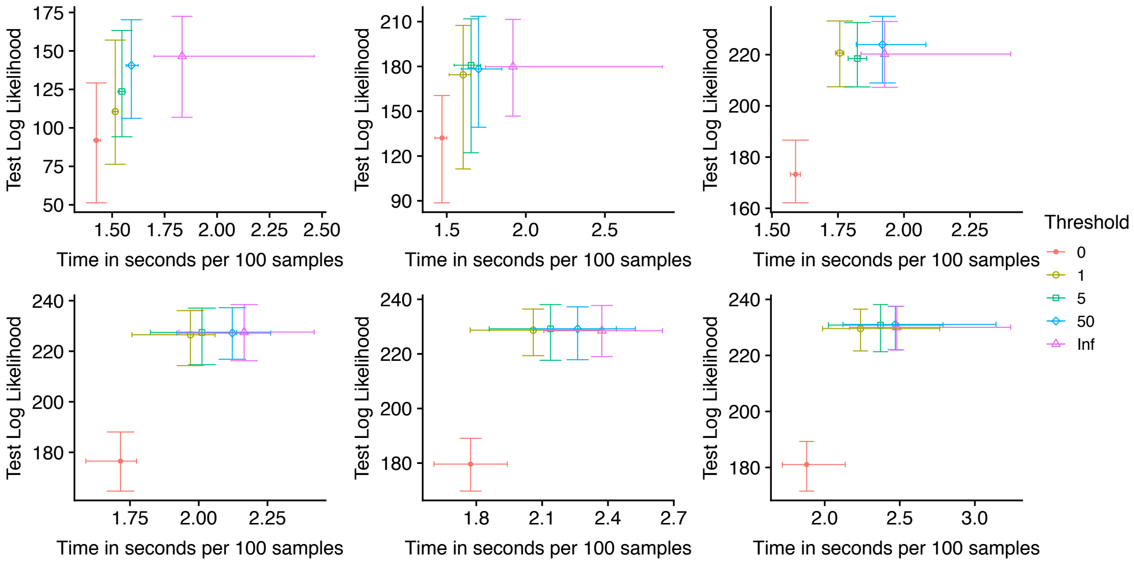


Figure 7.6.: Performance plot for method with varying sample sizes on TMoG model. From left to right and top to bottom, the training size is: $10, 50, 100, 200, 300$ and 500

We see that for small sample sizes (i.e. 10 and 50), the predictive likelihood does not show any significant change across the different threshold. When the training set is 100, there is a clear difference between the thresholded sampler to that of the model with no data augmentation scheme (threshold 0). However, we see that the test likelihood plateaus when the training set size is greater than 200. Along with our result in Chapter 4 (Figure 4.1), for this type of data, we would need at least 200 observations with threshold equals to 1 ($\rho = 0.5$) for good predictability.

7.5 Summary

In this chapter, we explored the effect of the concentration parameter of the DP on the performance of the model. We showed that the sampler is robust to the different settings in terms of accuracy and, model checking reveals that an $\alpha_0 = 1$ suffices for most experiments. Additionally, we introduced a slight modification to our thresholded sampler where we restrict the proposals to be within a specified distance from $\mathbb{S}$. Our experiments suggest that this can further improve mixing and has no effect to the predictability and efficiency of the model. Lastly, we explored the effect of sample size on the predictive performance of the model with results that are consistent to our results from Chapter 4.

8. CLUSTERING MUTATIONS AND ESTIMATING CONTAMINATION RATES VIA MIXTURE MODELS

8.1 Introduction

In this chapter and the next, we focus on the second contribution of this thesis which involves the use of nonparametric mixture models to better understand the characteristics of DNA structure in tumor cells. Cells that rapidly duplicate are susceptible to changes in the genetic information that they carry; often, these cells will form clusters of mutated cells commonly known as tumors. Malignant tumor cells, or cancer cells, can spread and invade nearby tissues in the body, and so understanding the DNA alteration that occurs in tumor cells is important to learn the genetic markers that are the drivers of tumor growth.

Scientists are particularly interested in DNA alterations that are somatic. These are mutations that are unique to the tumor environment, in constrast to germline mutations that are hereditary. Somatic mutations at a single base location of the genome are also referred to as single nucleotide variants (SNVs). We use both terms interchangeably in this chapter. To better distinguish between somatic mutations and germline mutations, studies have proposed the use of DNA sequencing of cells from matched normal and tumor tissues of a patient (Bergmann et al., 2016; Su et al., 2012; Roth et al., 2012). By doing this, scientists hope to reduce false discoveries and increase the accuracy of detecting somatic mutations.

Figure 8.1 illustrates the process of using DNA sequencing information from a matched normal and tumor sample. Reads from the sequencing machines are aligned to a reference genome (a database assembled by scientist that catalogues the DNA sequence representative of a species) and matches are computed at every location along the genome. The results are count observations at each locus of the DNA,

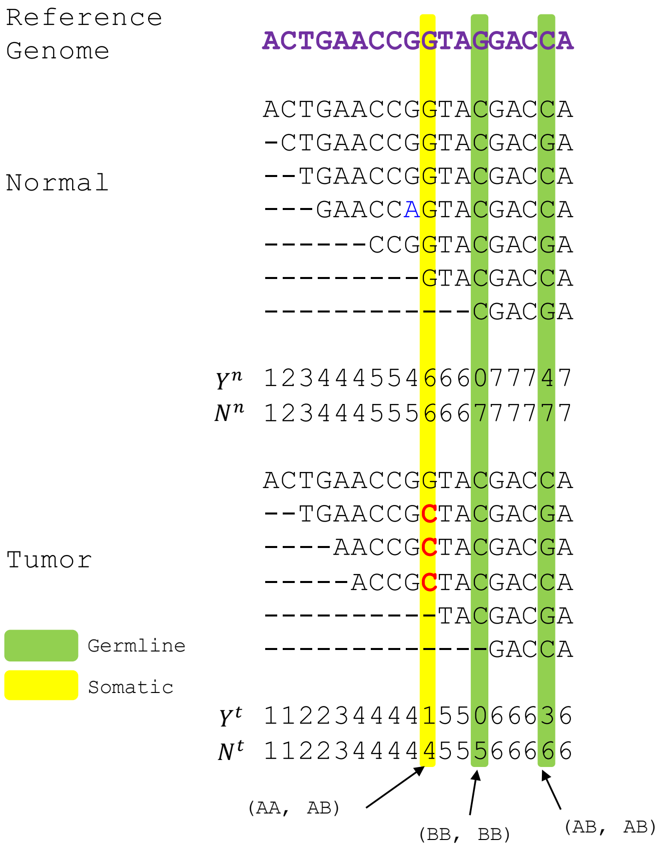


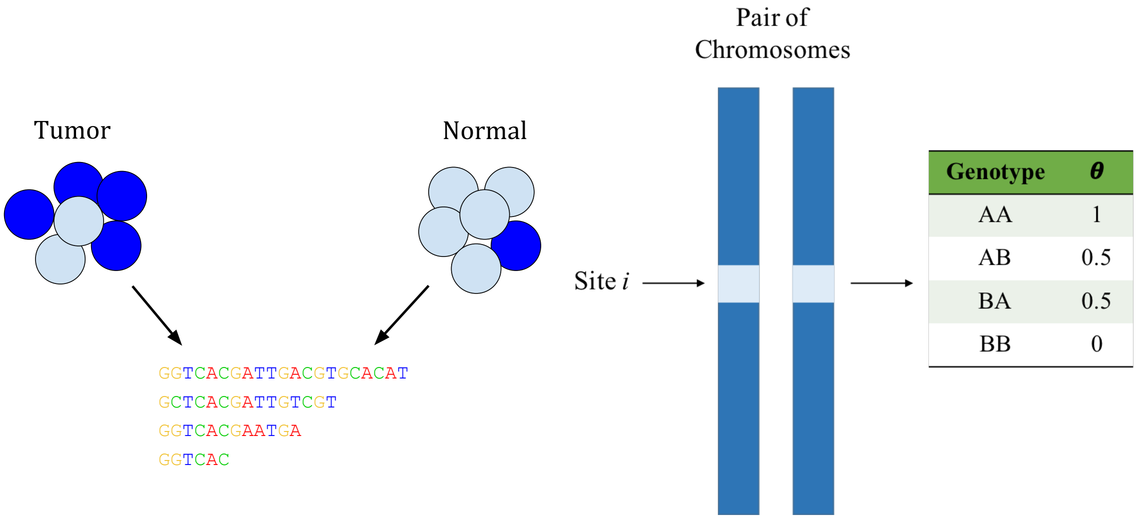
Figure 8.1: Example of read alignments from a tumor and normal sample to the reference genome (purple). The number of matches, Y, and the depth of the reads, N, are counted at each locus i. These informations are then used to determine whether a site is germline (green) or somatic (yellow).

consisting of the number of match reads and the total number of reads at that locus. Based on these findings, two alleles, A for a match and B for a mismatch, for that specific site in the DNA are determined. Given these, one can interpret the genetic constitution (genotype) of a base location using Table 8.1. Through this process, scientists are better able to distinguish and detect somatic mutations that exist in the tumor environment.

DNA sequencing of tumor samples are known to produce sample contamination which is a condition where tumor samples are contaminated with normal cells (or conversely, normal samples with tumor cells). Contaminations can alter the read match counts of a tumor sample with respect to the reference genome (Figure 8.1) which lead to difficulties and inaccuracies in downstream analysis, such as in validating predictive models of somatic mutations, and in providing biological interpretation of expressed genes (Zhao and Simon, 2010). While DNA sequencing of a single-cell

Table 8.1.: Biological interpretation of joint-genotypes (source: Roth et al. (2012)).

θ^n/θ^t	AA	AB	BB
AA	Wild-type	Somatic	Somatic
AB	LOH	Germline	LOH
BB	Error	Error	Germline



Genotype	θ
AA	1
AB	0.5
BA	0.5
BB	0

Figure 8.2.: An illustration of contaminated tumor and normal sample, along with their contaminated read sequences (left). A pair of chromosome at site i, its possible genotypes, and its associated probabilities of matching the reference genome.

can be used, the technology for this technique is expensive, prone to high technical variations, and lacks the sequencing depths found in bulk cells DNA sequencing technologies (Yadav and De, 2014).

Yadav and De (2014) presented a detailed study to compare existing methods across different technologies to estimate the contamination level of a sample. The current standard approaches are VerifyBamID (Jun et al., 2012) and ContEst (Cibulskis et al., 2011). Both of these methods use external databases to improve the accuracy of the estimation, making them dependent on the comprehensiveness of outside resources. Methods that involve the use of sequencing informations alone by means of a matched normal and tumor samples were also introduced (Bergmann et al., 2016; Su et al., 2012; Larson and Fridley, 2013). However, these methods do not concurrently provide detection of somatic SNVs. Further, current approaches assume contamination rates are uniform across all loci.

We propose two hierarchical Bayesian models to flexibly estimate contamination rate in a matched normal and tumor DNA sequencing data, while concurrently detecting somatic mutation at a base location. Our models allow for contamination levels to vary across locations, permitting information to be shared along the genome. This is particularly useful for when DNA sequencing data are noisy (Tomlinson and Oesper, 2018), which is commonly found in multi-sample bulk sequencing data. We do this via the use of a mixture model with a Dirichlet prior (Section 2.2). Next, we extend this framework to use a Dirichlet Process prior (2.4) that allows for contaminations to be shared by multiple locations along the genome and does not restrict the number of common contamination values. These models are described in more details in the next section.

8.2 Methodology

Consider a human genome with only two possible alleles at each base position in the genome, A and B, corresponding to a match and a mismatch to the reference genome (see Figure 8.2). Then, the set of possible diploid genotypes are $\mathcal{G} = \{BB, AB, AA\}$. For genotype AB, we should observe approximately 50% of the reads to match the reference genome (Kim et al., 2013). Similarly, we expect close to 100% of the reads to match the reference genome for genotype AA, and 0% matches for genotype BB. Thus, the match probabilities of observing the aforementioned combinations are {0, 0.5, 1}, respectively. Denote these probabilities by $\boldsymbol{\theta^n} = \{\theta_1^n, \theta_2^n, \ldots, \theta_m^n\}$ for sample with normal cells and $\boldsymbol{\theta^t} = \{\theta_1^t, \theta_2^t, \ldots, \theta_m^t\}$ for tumor samples, where m indicates the total number of locations (i.e. base positions) of the genome, and θ_i^n, $\theta_i^t \in \{0, 0.5, 1\}$.

Write $\zeta_i \in [0,1]$ for the contamination level at a specific location i in the chromosome. Now, the probability of observing a match is given by

$$\rho_i = \zeta_i \theta_i^n + (1 - \zeta_i)\theta_i^t \tag{8.1}$$

In a paired normal and tumor human sample, we have 2 measurements at each location i, and following the standard convention in Roth et al. (2012), the set of possible joint-genotypes are the Cartesian product of the set $\mathcal{G}$, which is $\mathcal{G} \times \mathcal{G} = \{(g_n, g_t) \in \mathcal{G}\}$. Accordingly, the joint-genotype probabilities at location m is also (θ_m^n, θ_m^t) where θ_i^n, $\theta_i^t \in \{0, 0.5, 1\}$. These joint-genotypes are of interest as they provide information as to the type of mutation that occured at a specific base position of the DNA, and can be interpreted as given by Table 8.1. Here, wild-type describes a condition of perfect matches in both normal and tumor samples: (AA, AA), somatic indicates a potential tumor variant: (AA, AB) or (AA, BB), germline is a variant in both samples of normal (AB, AB) and tumor (BB, BB), LOH is the loss of heterozygosity: (AB, AA) or (AB, BB). An error occur when a homozygous variant mutates back to match the reference genome; an unlikely event, and suggests a potential error in the base calling or alignment process.

Now, at each locus i, we have 2 contamination levels ζ_i^n for the normal sample and ζ_i^t for the tumor sample, as well as 2 match probabilities ρ_i^n and ρ_i^t, where ρ_i^j is the probability of matches at i for sample j. Similar to the one sample case, the probability of ρ_i^j is a linear combination of the probabilities of the joint-genotype such as,

$$\rho_i^j = \zeta_i^j \theta_i^n + (1 - \zeta_i^j)\theta_i^t$$

Let Y_i^j be the number of matches we observe at nucleotide location i and N_i^j be the total number of reads found at i for each sample j, then, Y_i^j follows a Binomial distribution given by,

$$Y_i^j \sim \text{Bin}(N_i^j, \rho_i^j)$$

We aim to estimate $\theta_i^n, \theta_i^t, \zeta_i^n$ and ζ_i^t. Note that θ_i^t and θ_i^n are shared by observations at the same location, and the mixing proportion ζ_i^j, may be shared across observations at different locations. In this fashion, we are able to determine the joint-genotype at each position while considering uniform or shared contamination rates.

8.2.1 Properties of ρ

Let $\tilde{\rho}_i^j$ be the empirical value for ρ_i^j, calculated by the fraction of observed matches at location i to the total number of reads at that location. Given $\tilde{\rho}_i^j$, we can evaluate all possible empirical $\tilde{\zeta}_i^j$ values for each i by enumerating the joint-genotype (θ_i^n, θ_i^t) and solving for $\tilde{\zeta}_i^j$ as follows

$$\tilde{\zeta}_i^t = \frac{\tilde{\rho}_i^t - \theta_i^t}{\theta_i^n - \theta_i^t}, \qquad \tilde{\zeta}_i^n = \frac{\tilde{\rho}_i^n - \theta_i^t}{\theta_i^n - \theta_i^t} \tag{8.2}$$

We assume that tumor samples have contamination levels of no more than half and normal samples will have contamination levels of no less than half. This is because the label of the samples are almost always known by the scientists performing cell extractions and/or sequencing of these cells.

Given the empirical parameter value, $\tilde{\rho}_i^n$ and $\tilde{\rho}_i^t$, and under our assumption that $\zeta_i^n - \zeta_i^t >= 0$, we can determine the characteristics of each contamination rate with varying pair of genotype.

$$\begin{aligned} \rho_i^n - \rho_i^t &= (\zeta_i^n - \zeta_i^t)\theta_i^n + (\zeta_i^t - \zeta_i^n)\theta_i^t & (8.3) \\ &= (\zeta_i^n - \zeta_i^t)(\theta_i^n - \theta_i^t) & (8.4) \end{aligned}$$

Given fixed θ_i^n and θ_i^t, there are 3 potential values for the ρ_i^n and ρ_i^t:

1. If $\theta_i^n < \theta_i^t$, then $\rho_i^n < \rho_i^t$ since $\zeta_i^n - \zeta_i^t >= 0$
2. If $\theta_i^n > \theta_i^t$, then $\rho_i^n > \rho_i^t$ since $\zeta_i^n - \zeta_i^t >= 0$
3. If $\theta_i^n = \theta_i^t$, then $\rho_i^n - \rho_i^t = 0$ or $\rho_i^n = \rho_i^t$

This suggests that if $\tilde{\rho}_i^n = \tilde{\rho}_i^t$, the model is unidentifiable with respect to the contamination rate, as any values of ζ's would satisfy the given pair of empirical parameters. The pair of observations with this characteristic would be of no use to the model because they fail to give any information of contamination. Including these observations into the analysis can bias the result. We later utilize this information to filter out unidentifiable observations and use only those that provides a unique solution to the system of linear equation above.

As previously mentioned, our goal is to allow the sharing of contamination levels across all base positions in the genome. We can do this by clustering ζ_i^j such that sites within the same cluster share the same contamination estimate. Similarly, we can also allow sites to perform clustering over the 9 values of the joint-genotype, (θ^n, θ^t). In this work, we take to both finite and infinite mixture models to represent these clusterings. In the finite setting, we assume countable number of components over ζ^n, ζ^t, and (θ^n, θ^t), by placing a Dirichlet prior over these parameters. We extend this framework to allow for infinite number of components over ζ^n and ζ^t, by placing a Dirichlet Process prior over these parameters. The DP prior will allow for flexible number of clusters with shared values drawn from a carefully chosen based distribution. We describe both models in the next two sections.

8.2.2 Mixture models with Dirichlet Prior (M1)

The models proposed by Jun et al. (2012) and Bergmann et al. (2016) involve dividing the parameter space when estimating the contamination level of a given sample. In this first model, we adopt the same approach by dividing the parameter space of ζ_i^j into fine grid and placing a Dirichlet prior on the distribution of ζ_i^j and the joint-genotype (θ_i^n, θ_i^t). Sampling from these parameters now involved sampling from a series of multinomial distribution which is straightforward to do.

Let $\boldsymbol{\alpha}$ be the concentration parameters associated for the joint-genotypes, and $\boldsymbol{\alpha_n}$ and $\boldsymbol{\alpha_t}$ the associated concentration parameters for ζ_i^n and ζ_i^t. The dimension of $\boldsymbol{\alpha_n}$

and $\boldsymbol{\alpha_t}$ depends on the number of components (grid) use in the model, while that of $\boldsymbol{\alpha}$ is 9, corresponding to the number of possible joint-genotype. Since the label of each sample is known (i.e. tumor or normal sample), we can impose prior knowledge on the value of the concentration parameters of the Dirichlet distribution to reflect this information. The complete generative process of the model is given by the following

$$\begin{aligned} \pi_i &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\ u_i &\sim \text{Multinomial}(\pi_i) \\ (\theta_i^n, \theta_i^t) &= (\theta^n, \theta^t)_{u_i} \end{aligned}$$

$$\begin{array}{ll} G_n \sim \text{Dirichlet}(\boldsymbol{\alpha_n}) & G_t \sim \text{Dirichlet}(\boldsymbol{\alpha_t}) \\ v_i^n \sim \text{Multinomial}(G_n) & v_i^t \sim \text{Multinomial}(G_t) \\ \zeta_i^n = \zeta_{v_i^n}^n & \zeta_i^t = \zeta_{v_i^t}^t \\ Y_i^n \sim \text{Bin}(N_i^n, \zeta_i^n \theta_i^n + (1 - \zeta_i^n)\theta_i^t) & Y_i^t \sim \text{Bin}(N_i^t, \zeta_i^t \theta_i^n + (1 - \zeta_i^t)\theta_i^t) \end{array}$$

This model can produce biases in its estimation due to the divisions of the parameter space. An increase in accuracy could potentially be obtained through smaller grid sizes but will lead to computational inefficiency in the estimation as more grid values have to be evaluated before choosing the optimal point. The same criticism has been raised to other methods of similar nature. Nevertheless, this model provides a simple representation to address our problem, as well as providing good verification for more complex models such as the DP mixture model which we discuss next.

8.2.3 Mixture models with Dirichlet Process prior (M2)

We extend M1 to allow for flexible clustering of the contamination rate, an alternative to dividing the parameter space into fine grids. We do this by placing a Dirichlet Process prior over both ζ^n and ζ^t through the Chinese restaurant process representation of the DP (see Section 2.4.1).

Let H_j be the base distribution of the individual Dirichlet Process for $j = \{n, t\}$. For example, we may place a Beta$(2, 5)$ or a Unif(0, 0.5) for H_t over ζ^t, to reflect the tumor sample label. This is also important to prevent the label switching property of the DP which we explain in more detail in Section 8.4. The complete data generating process for model M2 is as follows

$$
\begin{aligned}
\pi_i &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
u_i &\sim \text{Multinomial}(\pi_i) \\
(\theta_i^n, \theta_i^t) &= (\theta^n, \theta^t)_{u_i}^*
\end{aligned}
$$

$$
\begin{array}{ll}
G_n \sim DP(\alpha_n, H_n) & G_t \sim DP(\alpha_t, H_t) \\
\zeta_i^n \sim G_n & \zeta_i^t \sim G_t \\
Y_i^n \sim \text{Bin}(N_i^n, \zeta_i^n\theta_i^n + (1-\zeta_i^n)\theta_i^t) & Y_i^t \sim \text{Bin}(N_i^t, \zeta_i^t\theta_i^n + (1-\zeta_i^t)\theta_i^t)
\end{array}
$$

For both models M1 and M2, the likelihood equation is then the product of the likelihood of m locations as follows

$$
\begin{aligned}
L(\boldsymbol{\zeta^n}, \boldsymbol{\zeta^t}, \boldsymbol{\theta^n}, \boldsymbol{\theta^t}|\tilde{\mathbf{Y}}) &= \prod_{i=1}^{m} \text{Bin}(N_i^n, \zeta_i^n\theta_i^n + (1-\zeta_i^n)\theta_i^t)\text{Bin}(N_i^t, \zeta_i^t\theta_i^n + (1-\zeta_i^t)\theta_i^t) \\
&= \prod_{i=1}^{m} \binom{N_i^n}{y_i^n}(\zeta_i^n\theta_i^n + (1-\zeta_i^n)\theta_i^t)^{y_i^n}(1-\zeta_i^n\theta_i^n + (1-\zeta_i^n)\theta_i^t)^{N_i^n - y_i^n} \times \\
&\qquad \binom{N_i^t}{y_i^t}(\zeta_i^t\theta_i^n + (1-\zeta_i^t)\theta_i^t)^{y_i^t}(1-\zeta_i^t\theta_i^n + (1-\zeta_i^t)\theta_i^t)^{N_i^t - y_i^t}
\end{aligned}
$$

We take the Bayesian approach to estimating the parameters of both models through Markov chain Monte Carlo methods. Specifically, we adopt the Gibbs sampling algorithms from Section 2.5.

8.3 Markov chain Monte Carlo sampling

We employ the standard Gibbs sampling procedure to draw from the posterior distribution of the parameters. For each model, we derive the conditional distribution from the joint distribution of the corresponding model. The following sections show the detail of the update steps involve in the Gibbs sampling procedures.

8.3.1 Gibbs sampling for M1

The Gibbs sampler for M1 involves evaluating the conditional probability of ζ_i^n and ζ_i^t at each point of the grid in the parameter space, and evaluating the probability of $(\theta^n, \theta^t)_i$ across its 9 values. The following Equation provides the conditional distribution of sampling all 3 parameters at location i:

$$P(\zeta^n_{v^n_i}, \zeta^t_{v^t_i}, (\theta^n, \theta^t)_{u_i} | (\boldsymbol{\theta^n}, \boldsymbol{\theta^t})_{\boldsymbol{-i}}, \boldsymbol{\zeta^n_{-i}}, \boldsymbol{\zeta^t_{-i}}, \boldsymbol{Y^n}, \boldsymbol{Y^t}, \boldsymbol{N^n}, \boldsymbol{N^t}, \boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \boldsymbol{\alpha})$$

$$\propto P(v_i^n = k, v_i^t = l, u_i = j | \boldsymbol{u_{-i}}, \boldsymbol{v^n_{-i}}, \boldsymbol{v^t_{-i}}, \boldsymbol{Y^n}, \boldsymbol{Y^t}, \boldsymbol{N^n}, \boldsymbol{N^t}, \boldsymbol{\alpha_n}, \boldsymbol{\alpha_t}, \boldsymbol{\alpha}) \tag{8.5}$$

$$\begin{aligned} &\propto P(Y_i^n = y_i^n | v_i^n = k, u_i = j, N_i^n) P(Y_i^t = y_i^t | v_i^t = l, u_i = j, N_i^t) \times \\ &\quad P(v_i^n = k | \boldsymbol{v^n_{-i}}, \boldsymbol{\alpha_n}) P(v_i^t = l | \boldsymbol{v^t_{-i}}, \boldsymbol{\alpha_t}) P(u_i = j | \boldsymbol{u_{-i}}, \boldsymbol{\alpha}) \end{aligned} \tag{8.6}$$

where $k = \{1, \ldots, K\}, l = \{1, \ldots, L\}, j = \{1, \ldots, 9\}$, and the value of $\{K, L\}$ depends on the total number of grid spaces use in the model. To update the parameters, the algorithm samples from $K \times L \times 9$ vector of probabilities. After integrating out the respective G_n and G_t, the conditional distribution of the latent class indicators are given by:

$$P(v_i^n = k | \boldsymbol{v^n_{-i}}, \boldsymbol{\alpha_n}) = \frac{m_{k,-i} + \alpha_{nk}}{m - 1 + \sum_{r=1}^{K} \alpha_{nr}}. \tag{8.7}$$

Similarly,

$$P(v_i^t = l | \boldsymbol{v^t_{-i}}, \boldsymbol{\alpha_t}) = \frac{m_{l,-i} + \alpha_{tl}}{m - 1 + \sum_{r=1}^{L} \alpha_{tr}} \tag{8.8}$$

where $m_{k,-i}$ is the number of observations in cluster k minus observation i. Lastly, for the joint-genotype we have:

$$P(u_i = j | \boldsymbol{u_{-i}}, \boldsymbol{\alpha}) = \frac{m_{j,-i} + \alpha_j}{m - 1 + \sum_{r=1}^{9} \alpha_r}. \tag{8.9}$$

Equations (8.6) to (8.9) provide the main building block for an iteration of our Gibbs sampler. The complete steps are given by Algorithm 5. This procedure can be repeated to produce the desired number of samples after the chain converges.

Algorithm 5: Gibbs sampling for Model M1

```
Initialize $v_i^n$, $v_i^t$, and $u_i$ for each $i = 1, \dots, m$;
for $i = 1, \dots, m$ do
    Compute Equation (8.7) and (8.8) for all $k = 1, \dots, K$ and $l = 1, \dots, L$;
    Compute Equation (8.9) for all $j = 1, \dots, 9$;
    Compute $\text{Bin}(y_i^n|N_i^n, \rho_i^n), \text{Bin}(y_i^t|N_i^t, \rho_i^t)$;
    Compute conditional probability as in Equation (8.6);
    Update $v_i^n$, $v_i^t$, and $u_i$
end
Repeat
```

8.3.2 Gibbs sampling for M2

In this second model, we place a Dirichlet process prior over the contamination level ζ_i^j. Consequentially, sampling from the conditional distribution becomes more involved. We adopt the Chinese Restaurant Process representation of the DP (see: Section 2.4.1) and the corresponding Gibbs sampler (see: Section 2.5). We write $\rho_{ic}^j = \zeta_c^j \theta_i^n + \zeta_c^j \theta_i^t$, the probability of observation i given the parameter of component c, ζ_c^j. To update the cluster assignment corresponding to ζ_i^j, we sample from the conditional distribution given by the following:

$$P(c_i^n = c | c_{-i}^n, y_i^n, \zeta_1^n, \cdots, \zeta_h^n) = \begin{cases} b\frac{m_{-i,c}}{m-1+\alpha}\text{Bin}(y_i^n|N_i^n, \rho_{ic}^n), & \text{for } 1 \le c \le k^- \\ b\frac{\alpha/s}{m-1+\alpha}\text{Bin}(y_i^n|N_i^n, \rho_{ic}^n), & \text{for } k^- < c \le h \end{cases} \tag{8.10}$$

$$P(c_i^t = c | c_{-i}^t, y_i^t, \zeta_1^t, \cdots, \zeta_h^t) = \begin{cases} b\frac{m_{-i,c}}{m-1+\alpha}\text{Bin}(y_i^t|N_i^t, \rho_{ic}^t), & \text{for } 1 \le c \le k^- \\ b\frac{\alpha/s}{m-1+\alpha}\text{Bin}(y_i^t|N_i^t, \rho_{ic}^t), & \text{for } k^- < c \le h \end{cases} \tag{8.11}$$

where $m_{-i,c}$ are the number of observations in cluster c not including observation i, α is the concentration parameter of the Dirichlet Process, and s is the number of latent components.

To keep the distribution of ζ^n and ζ^t invariant, we implement the Metropolis-Hastings algorithm at every update step of ζ^n and ζ^t. Let $\text{Unif}(\alpha_0^j, \beta_0^j)$ be the transition probability for $j = \{n, t\}$. Assuming symmetric transition probabilities, we accept a candidate draw for the ζ^j with the following probability:

$$\begin{aligned} r &= \frac{\prod_{i=1}^{m_c} P(y_i|\zeta_c^*)P(\zeta_c^*)}{\prod_{i=1}^{m_c} P(y_i|\zeta_c^{r-1})P(\zeta_c^{r-1})} && (8.12) \\ &= \frac{\prod_{i=1}^{m_c} \text{Bin}(y_i^j|N_i^j, \zeta_c^*\theta_i^n + (1-\zeta_c^*)\theta_i^t)\text{Unif}(\zeta_c^*|\alpha_0^j, \beta_0^j)}{\prod_{i=1}^{m_c} \text{Bin}(y_i^j|N_i^j, \zeta_c^{r-1}\theta_i^n + (1-\zeta_c^{r-1})\theta_i^t)\text{Unif}(\zeta_c^{r-1}|\alpha_0^j, \beta_0^j)} && (8.13) \end{aligned}$$

where ζ^* denotes the candidate ζ, $\zeta^{(r-1)}$ denotes the current ζ, m_c is the number of observations in cluster c, and subscript c is the cluster indicator. The update step of Equation (8.13) is performed for all $c \in \{1, \cdots, k^-\}$.

Let u_i be elements of $\{1, \ldots, 9\}$, indicating cluster assignment for observation i associated with the 9 joint-genotypes. Similar to that of M1, to sample the cluster indicator we have,

$$P(u_i = j|\boldsymbol{u_{-i}}, \boldsymbol{\alpha}) = \frac{m_{j,-i} + \alpha_j}{m - 1 + \sum_{r=1}^{9} \alpha_r} \qquad (8.14)$$

We can then sample from the conditional distribution from the 9 joint-genotype (θ_i^n, θ_i^t) as follows

$$\begin{aligned} P(\theta_i^t = k, \theta_i^n = l|\boldsymbol{\theta^t_{-i}}, \boldsymbol{\theta^n_{-i}}, \zeta_i^n, \zeta_i^t, y_i^n, y_i^t, N_i^n, N_i^t) \propto P(y_i^n|u_i = j, N_i^n, \zeta_i^n)P(y_i^t|u_i = j, N_i^t, \zeta_i^t)\times \\ P(u_i = j|\boldsymbol{u_{-i}}, \boldsymbol{\alpha}) \end{aligned}$$

Finally we get,

$$P(u_i = j|\boldsymbol{u_{-i}}, \zeta_i^n, \zeta_i^t, y_i^n, y_i^t, N_i^n, N_i^t) = \text{Bin}(N_i^n, \rho_i^n)\text{Bin}(N_i^t, \rho_i^t)\frac{m_{j,-i} + \alpha_j}{m - 1 + \sum_{r=1}^{9} \alpha_r} \qquad (8.15)$$

8.4 Identifiability

The label switching problem arises when using Markov chain Monte Carlo to draw posterior samples in mixture models or any other latent variable models. It occurs

Algorithm 6: Gibbs sampling for Model M2

```
Initialize $\zeta_i^n$, $\zeta_i^t$, $c_i^n$, $c_i^t$, and $u_i$ for each $i = 1, \ldots, m$;
for $d \in n, t$ do
    for $i = 1, \ldots, m$ do
        Draw a new value for $c_i^d$ from $\{1, \ldots, k\}$ according to Equation (8.10)
          and (8.11)
    end
    Draw a new value for $\boldsymbol{\zeta}^d$ with acceptance probability given by Equation
      (8.13).
end
for $i = 1, \ldots, m$ do
    Draw $u_i$ from Equation (8.15)
end
Repeat
```

when unsupervised assignment of labels to distinct groups is infeasible due to the permutation invariance property of the likelihood. For instance, in a 2-component mixture model, even if the 2 clusters are clearly separated, the labels $(1, 2)$ and $(2, 1)$ have equal posterior probability. Both Celeux et al. (2000) and Stephens (2000) provides deeper explanations to the label switching problems. Formally, assume finite number of components, K, in the probabilistic models given in Equation (2.3). Inference for this model is drawn from the posterior distribution of the parameters $P(\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{c}|\boldsymbol{X})$. Define any permutation of the parameters as,

$$\nu(\boldsymbol{\pi}, \boldsymbol{\theta}) = ((\pi_{\nu(1)}, \ldots, \pi_{\nu(k)}), (\theta_{\nu(1)}, \ldots, \theta_{\nu(k)})) \tag{8.16}$$

Given a set of parameter values for $(\boldsymbol{\pi}, \boldsymbol{\theta})$, the order of which these values are associated to which components, does not affect the likelihood of the mixture model. In other words, the likelihood is invariant with respect to the labels of the components, such that the likelihood $L(\boldsymbol{\pi}, \boldsymbol{\theta}|\boldsymbol{X}) = L(\nu(\boldsymbol{\pi}, \boldsymbol{\theta})|\boldsymbol{X})$. When exchangeable priors are imposed on the parameters, the resulting posterior distributions are invariant under all permutations ν. Without a specific constraint on the priors, the chain of the MCMC method will explore all possible arrangements of the posterior, resulting in symmetric multi-modal distribution. This multimodality of the posterior is problematic if the task at hand is to cluster observations into meaningful groups, as the marginal posterior distribution is the same for each component.

Researchers have also noted that label switching is important to MCMC convergence (Jasra et al., 2005). If the modes are well separated, the sampler may get stuck in exploring only one mode and is unable to jump to other modes. When this happens, posterior inference over each component could be meaningful, however, convergence of the chain may be questionable as it has fail to explore all modes of the posterior distribution.

Label switching is a problem in our setting because if we allow the contamination rates to sample from $[0, 1]$, the posterior will explore ζ values that potentially switches the label of the tumor and normal sample. As an example, a $\zeta_i^t = 0.25$ with joint-genotype $(\theta_i^n, \theta_i^t) = (1, 0)$ result in the same binomial probability as $\zeta_i^t = 0.75$ and $(\theta_i^n, \theta_i^t) = (0, 1)$. This leads to inaccuracy in both the estimation of the contamination rate, as well as in the detection of somatic mutations.

Several methods in the literature have been proposed to counter the problem of label switching. The most natural method is to adapt the idea from the frequentist point of view, which is to impose an artificial contraint on the parameter estimates (i.e. $\pi_1 < \pi_2 < \ldots < \pi_K$) to break the symmetry (Diebolt and Robert, 1994; Chung et al., 2004). Other methods includes: proposing a random permutation of the labels through a Metropolis-Hastings step in the Gibbs sampling procedure (Papastamoulis and Iliopoulos, 2010; Jasra et al., 2005; Frühwirth-Schnatter, 2001), finding a permutation that minimizes a certain loss function (Celeux, 1998; Stephens, 2000; Nobile and Fearnside, 2007; Rodriguez and Walker, 2014), and incorporating uncertainties through a probabilistic relabeling strategy (Yao, 2012; Jasra, 2006).

Our approach to counter the label switching problem and provide a more accurate estimate to the contamination, is to use an informative prior over the contamination level. More specifically, we placed a Unif$(0, 0.3)$ for the tumor sample and Unif$(0.7, 1)$ for the normal sample.

8.5 Summary

We introduced two mixture models, M1 and M2, to estimate contamination levels and detect somatic mutations in a matched tumor and normal sample. We do this to allow for sharing of contamination levels across the genome. Both models differ in the prior placed over for the contamination level, M1 uses the Dirichlet prior and M2 utilizes the Dirichlet Process prior. We defined the models and described the MCMC algorithms associated with each one. Additionally, we reviewed the label switching problem inherent to mixture models and outlined our strategy to tackle this problem.

9. EXPERIMENTS ON SYNTHETIC AND REAL DATA

9.1 Introduction

We evaluated our models described in Chapter 8 in a variety of experimental settings. We considered three different scenarios; the first involves a uniform contaminaton level across all locations in the genome, the second deals with varying level of contaminations across the genome, and the third involves *in silico* data of DNA sequencing from a matched normal and tumor sample. Lastly, we applied our models to publicly available data. We compared both M1 and M2 in the first two experiments and, as M2 is more efficient with large datasets, implemented only M2 in the last two experiments.

9.2 Simulation studies

For each experiment, we use a grid size of 0.01 over the parameter space of both ζ^n and ζ^t. To reflect the known sample label in M1, we constrained the parameter space of ζ^n to $[0.7, 1]$ and ζ^t to $[0, 0.3]$. In model M2, we do this by setting the base distribution for ζ^n, H_n, to be $\text{Unif}(0.7, 1)$, and that of ζ^t, H_t, to be $\text{Unif}(0, 0.3)$. For both models, the prior for the joint-genotype is a symmetric Dirichlet prior with $\alpha = 1$. The MCMC simulations were run with 5000 iterations and a burn-in of 2500. Random initializations were used at the start of the chain.

9.2.1 Uniform contamination

In this first experiment, we consider uniform (i.e. fixed) contamination rates for all i along the genome. We generated synthetic data of 200 observations from a binomial distribution, using randomly selected joint-genotypes (θ_i^n, θ_i^t), and a known

Table 9.1.: Mean absolute errors and standard deviations for both M1 and M2 on uniform contamination rate.

Prior	ρ^t	ρ^n	ζ^t	ζ^n
Dirichlet	0.0072 ± 0.0001	0.0078 ± 0.0002	0.0184 ± 0.0003	0.0194 ± 0.0003
DP	0.0019 ± 0.0002	0.0021 ± 0.0003	0.0040 ± 0.0004	0.0044 ± 0.0005

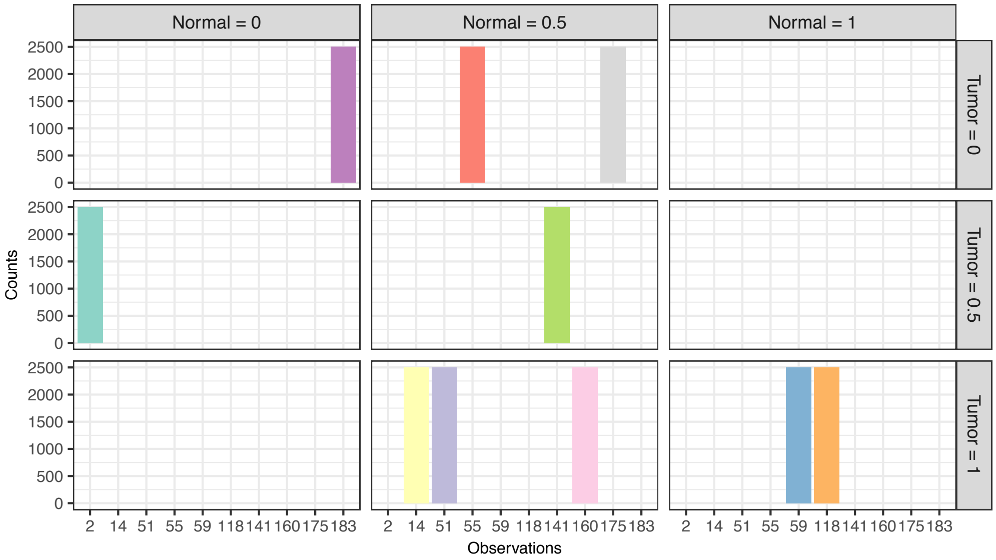


Figure 9.1.: Posterior distribution of the joint-genotype, (θ^n, θ^t), for M1 (result is similar for M2) on data with uniform contamination rate. To draw inference from this figure, observation 59 appears to have a joint-genotype of (1, 1) which based on Table 8.1, suggests a wild-type mutation.

contamination level of $\zeta_i^t = 0.1$ and $\zeta_i^n = 0.9$ for all $i = 1, \cdots, 200$. The depth of reads, N_i^j, is distributed as a Poisson random variable with mean 200.

Figure 9.1 shows the distribution of the joint-genotype for 2500 MCMC samples of 10 random observations. Both models provided consistent results and were able to recover the correct joint-genotypes for all 200 observations. Figure 9.2 and Figure 9.3 show the posterior distribution of ρ^j and ζ^j for both models. We see that variance of the estimates is higher when using a Dirichlet prior over a DP prior, indicated by a

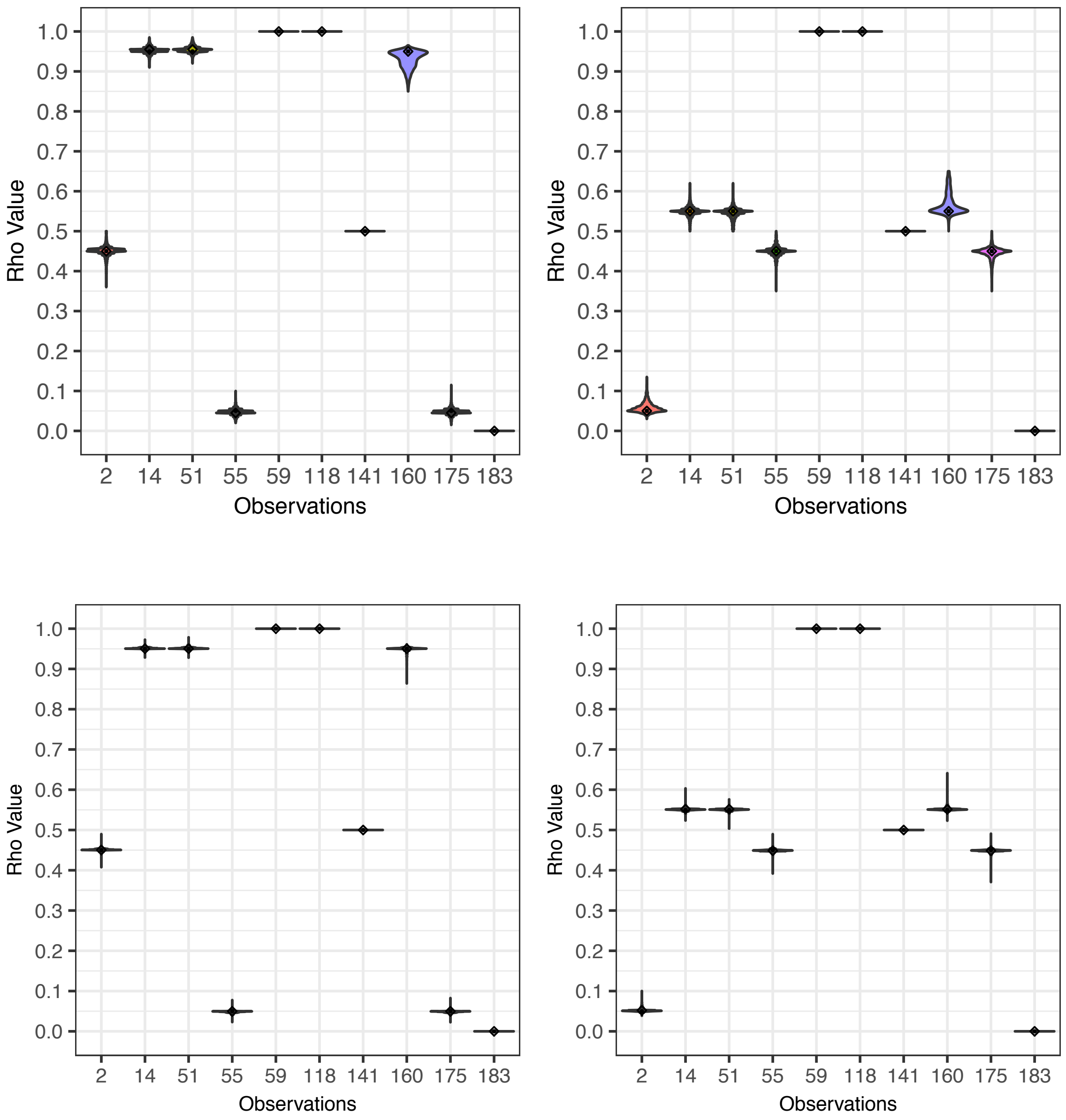


Figure 9.2.: Posterior distribution of ρ's on 10 randomly selected observations for model M1 (top) and M2 (bottom) on data with uniform contamination rate. In each subfigure, the left shows the posterior probabilities of the tumor sample, ρ^t, and the right shows that of the normal sample, ρ^n. Filled circles show the real ρ values of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distribution.

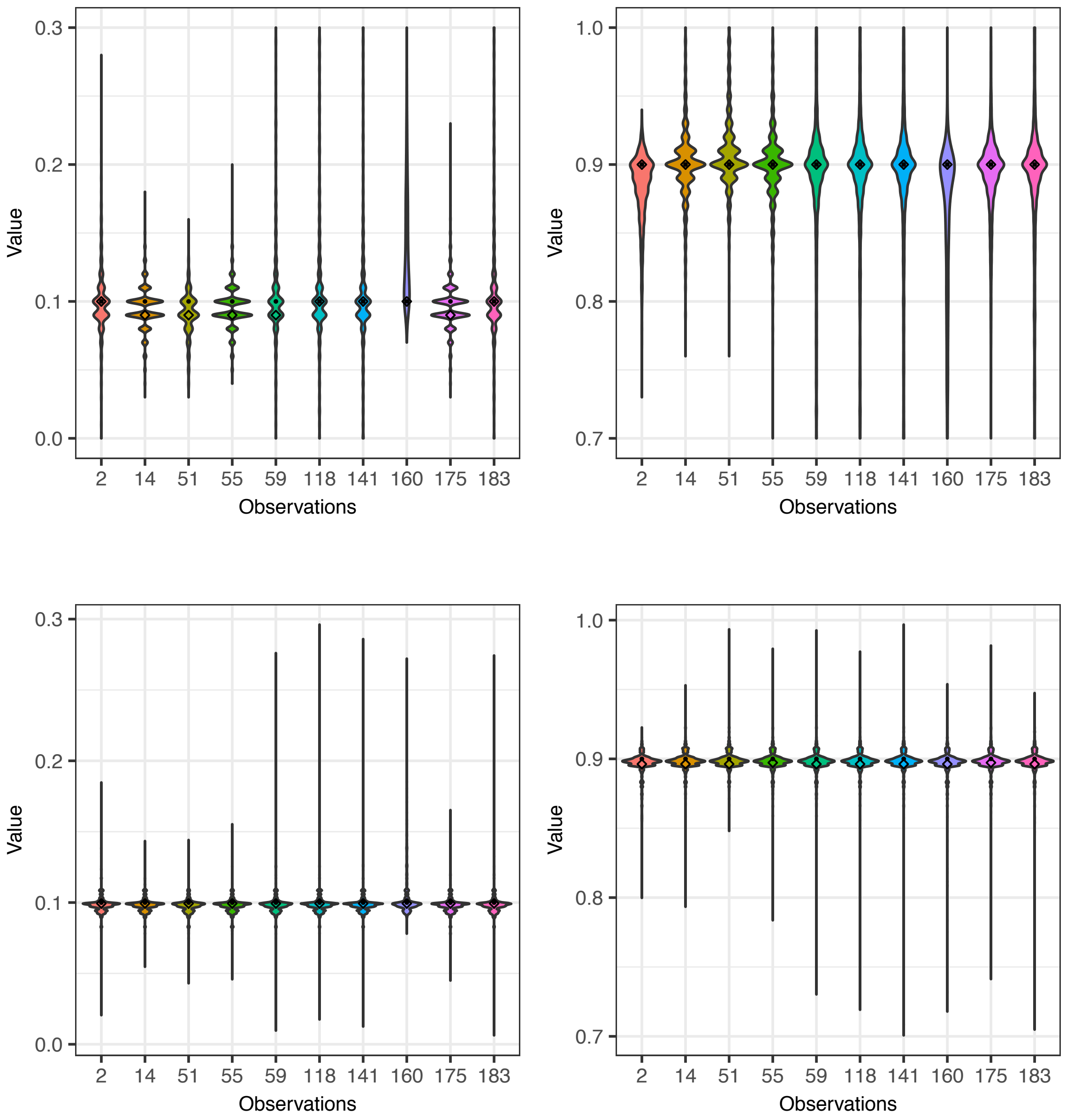


Figure 9.3.: Posterior distribution of ζ's on 10 randomly selected observations for model M1 (top) and M2 (bottom) on data with uniform contamination rate. In each subfigure, the left shows the posterior distributions of the contamination rate on the tumor sample, ζ^t, and the right shows that of the normal sample, ζ^n. Filled circles show the real ζ values of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distribution.

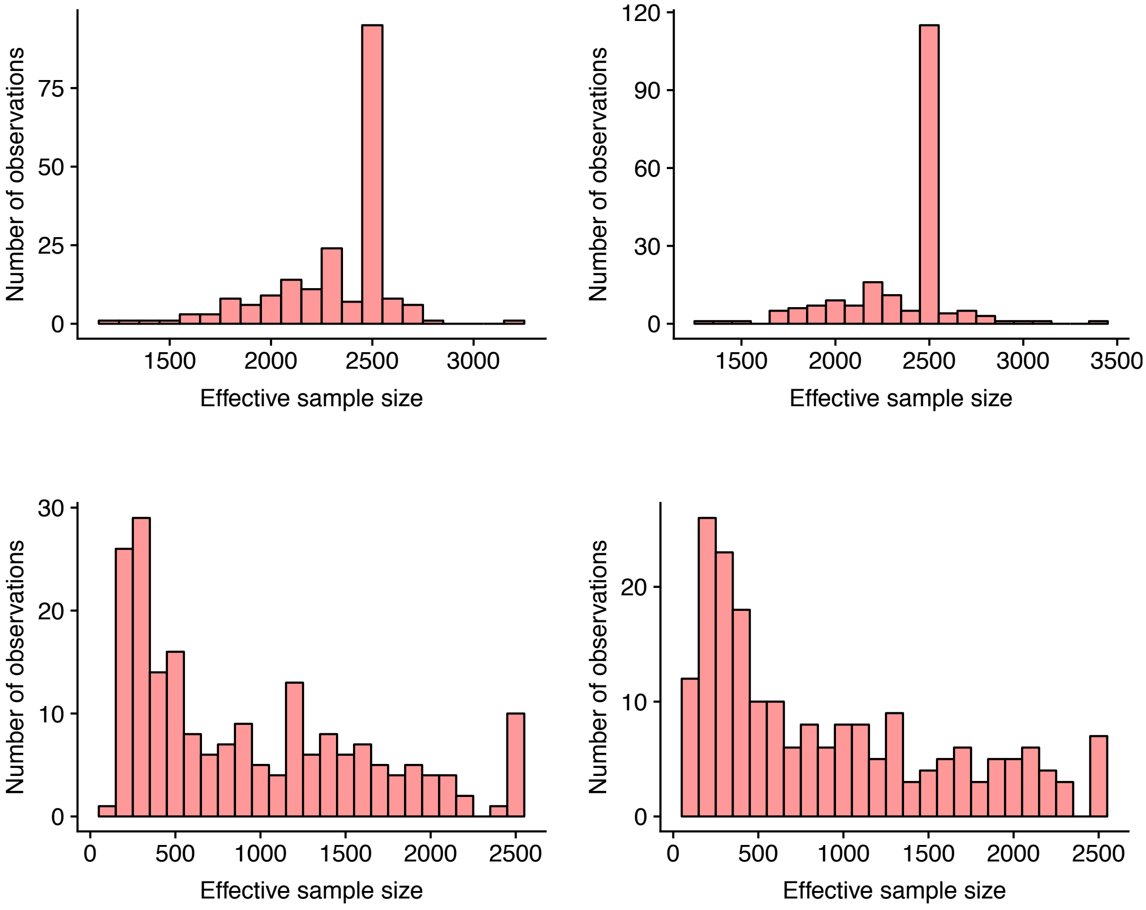


Figure 9.4.: Effective sample size for the MCMC samples of ζ^t (left) and ζ^n (right) for M1 (top) and M2 (bottom) on data with uniform contamination rate.

wider violin plot. Both models have explored the entire parameter space of each ζ and the ESS, given by Figure 9.4, suggests good mixing, with most observations in M1 having low correlation. On the other hand, the effective sample size for observations in M2 have greater variation, lying between 76 to 2500 samples.

To measure the accuracy and efficiency of both models and algorithms, we ran 20 different initializations on the same data. At every iteration, we take the absolute difference between the estimate and the real value as a measure of error. We use the posterior mode as parameter estimates for M1, and the posterior mean for M2. We use the Mean Absolute Error (MAE) to evaluate the performance of every initializations, and then computed the mean and standard deviation of the MAE across all initializations. Table 9.1 suggests that both models accurately estimated the binomial probabilities ρ^j for each sample $j \in \{n, t\}$ and all initializations were able to correctly identify the joint-genotype, (θ_i^n, θ_i^t), for all observations $i = 1, \cdots, m$.

Additionally, we measured the efficiency of the two models across the different initializations. The mean time in seconds for M1 is 209 per 100 MCMC samples, where the 5'th and 95'th quantile lies at 207.79 and 210.33 seconds, respectively. On the other hand, the average time for M2 is 5.72 seconds per 100 iterations, with the same quantiles at 5.64 and 5.97 seconds. These result suggest that the model with a Dirichlet prior (M1) performs significantly slower than that of the Dirichlet Process prior (M2). This is because M1 requires more extensive computation as the likelihood at each grid of the parameter space must be evaluated. Efficiency could potentially be improved by using wider grid sizes, at the cost of accuracy to the parameter estimates.

9.2.2 Varying contaminations

Our second experiment involved another synthetic data set where the contamination rates vary by locations. Contamination levels were generated by randomly selecting a set of discrete, equally spaced ζ^t values in $[0, 0.3]$ and ζ^n values in $[0.7, 1]$ with grid size of 0.01. Observations in each sample were drawn from a binomial distribution with randomly assigned joint-genotype (θ^n, θ^t), and the corresponding ζ value for each i. The total number of reads, N_i, were drawn from a $\text{Unif}(100, 200)$, and rounded down to the nearest integer.

As with the previous experiment, the results show a consistency between the two models. Figure 9.5 shows the posterior distribution of the match probabilities of 10 randomly sampled observations. It indicates that the true binomial probabilities ρ^j, where $j \in \{n, t\}$, were recovered by the models as it lies within the range of the posterior distributions. Naturally, a DP prior produces a much smoother distribution

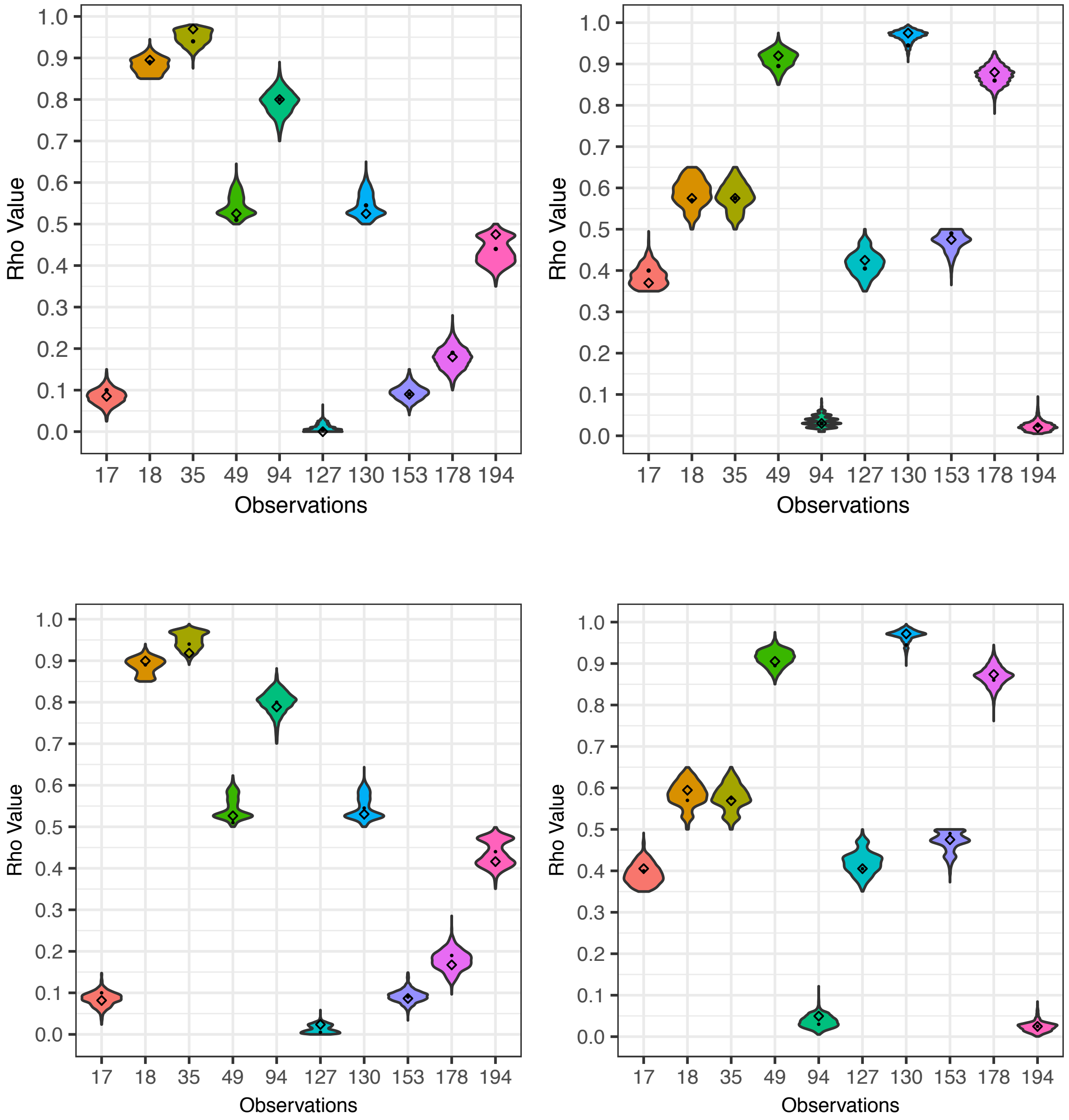


Figure 9.5.: Posterior distribution of ρ's on 10 randomly selected observations for model M1 (top) and M2 (bottom) on varying contamination rate across locations. In each subfigure, the left shows the posterior probabilities of the observations on tumor sample, ρ^t, and the right shows that of the normal sample, ρ^n. Filled circles show the real ρ values of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distribution.

Table 9.2.: Mean Absolute Errors and standard deviations for both M1 and M2 on varying contamination rate.

Prior	ρ^t	ρ^n	ζ^t	ζ^n
Dirichlet	0.01717 ± 0.00003	0.01755 ± 0.00002	0.05432 ± 0.0007	0.05535 ± 0.0007
DP	0.01671 ± 0.00009	0.01783 ± 0.00063	0.05249 ± 0.00022	0.05470 ± 0.00049

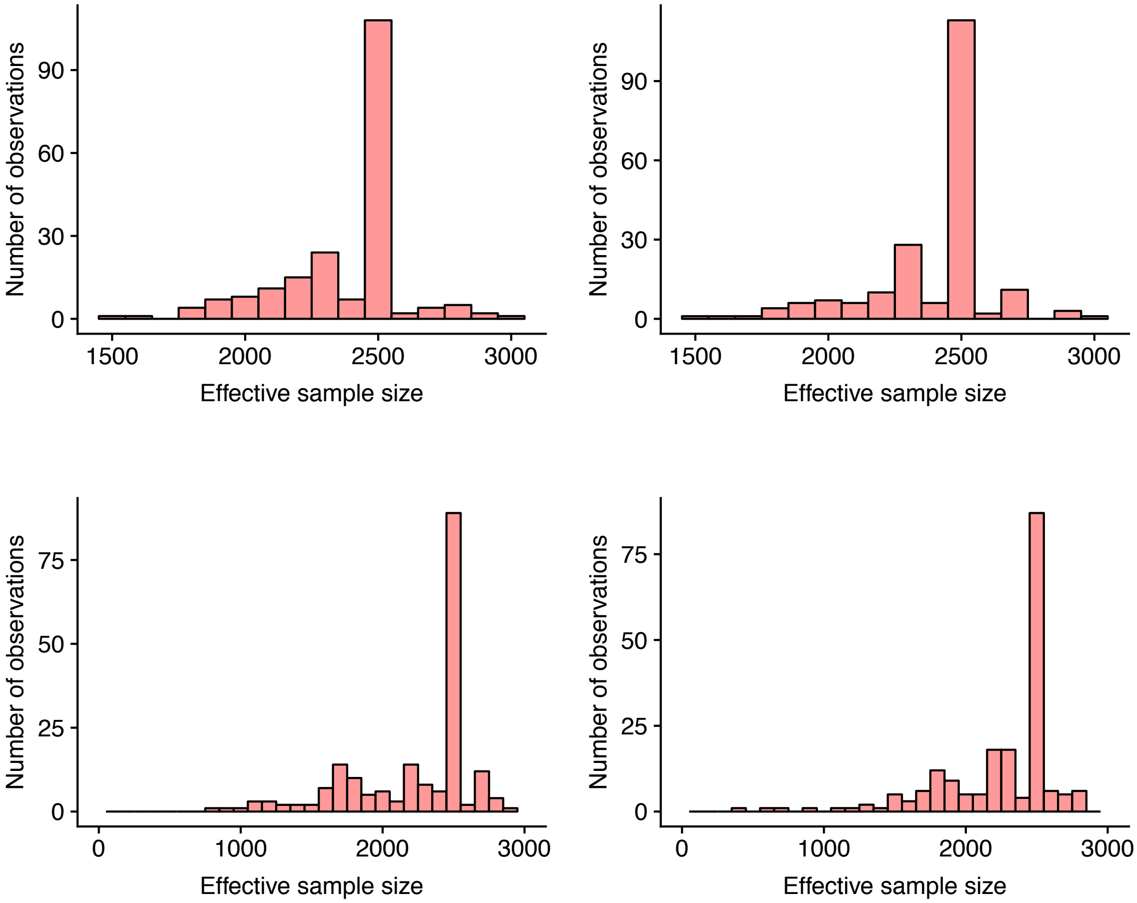


Figure 9.6.: Effective sample size for the MCMC samples of ζ^t (left) and ζ^n (right) for both M1 (top) and M2 (bottom)

(e.g. observations 94 and 127 in Figure 9.7), as the component parameters were drawn from a continuous base distributions. Figure 9.6 suggests that both algorithms have mixed well and have explored the entire parameter space, in terms of the parameters ζ^t and ζ^n.

Similar to the previous experiment, we ran 20 different initializations of the chain on the same data and summarized the accuracy and efficiency of both algorithms in Table 9.2. Again, the algorithms were able to accurately estimate the binomial probabilities up to one hundredth of a decimal, with the DP having slightly higher deviations. Both models exhibit higher error in estimating the contamination when compared to that of the uniform contaminations. This is due to the fact that the possible values for the contamination is now higher. All initializations were able to correctly identify the joint-genotype for every observations, despite having higher errors in the contamination estimate. This suggests that there are potentially multiple solutions to the contamination rate given a fixed joint-genotype. In terms of efficiency, again, the DP prior performed much faster than the Dirichlet prior, averaging at 8 seconds per 100 iterations compare to 76 seconds for the Dirichlet prior.

9.2.3 *In silico* data

Our final experiment involved matched *in silico* tumor and normal samples which more closely represents the distribution found in the real data. The contamination level is known to be at 0.1 for the tumor sample, and 0.9 for the normal sample. The raw data contains 86336 common locations amongst both samples.

Biologically, most locations will have similar distribution across both tumor and normal samples, as most locations will not be affected by the tumor. Data of these kind are known to be really noisy, with only a handful of observations containing insightful information. The left figure in Figure 9.9 indicates that the empirical probability (the observed matches over the depth of reads) of the locations are highly correlated across the two samples; quantitatively, the Pearson correlation is 0.98,

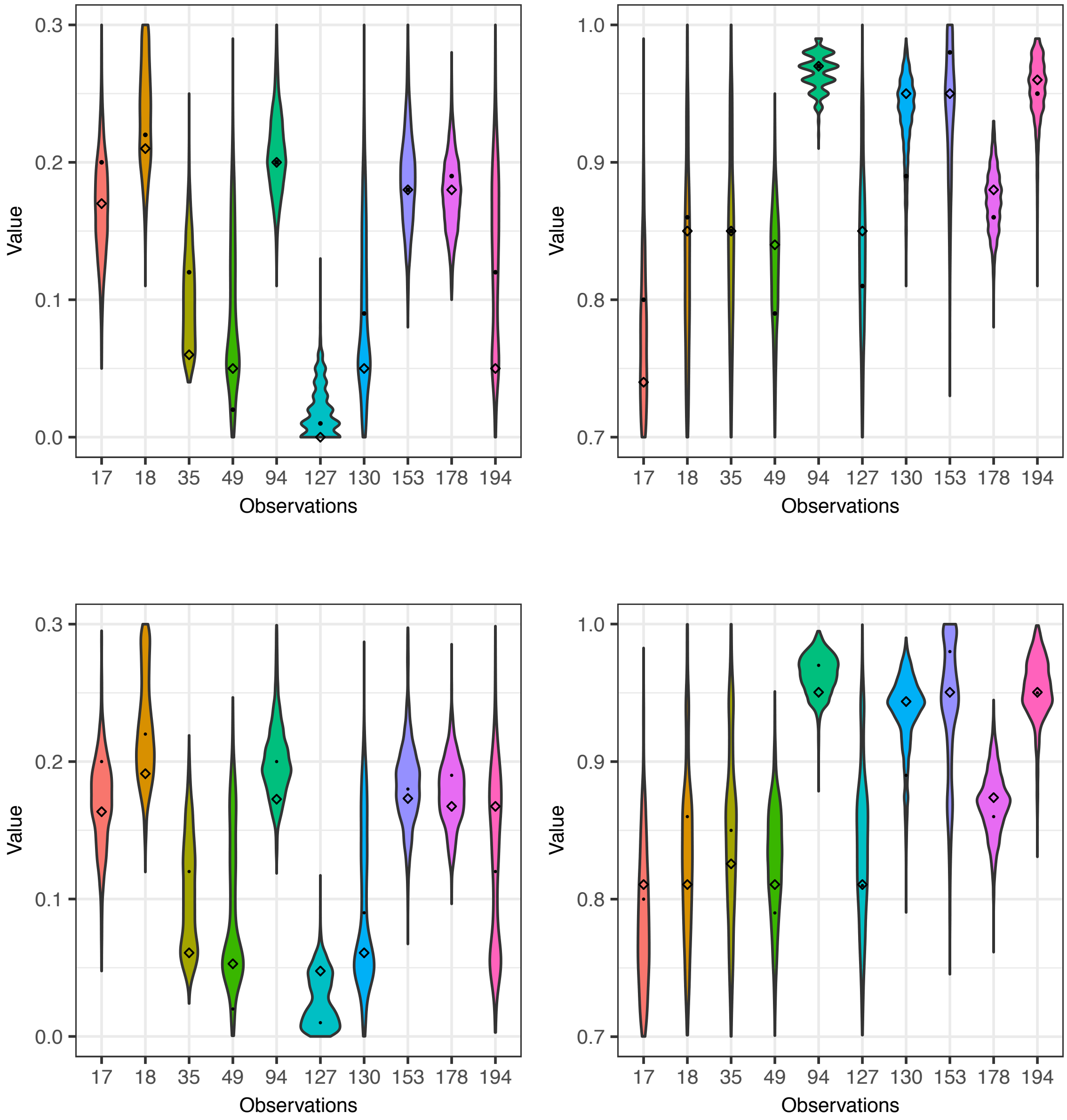


Figure 9.7.: Posterior distribution of ζ's on 10 randomly selected observations for model M1 (top) and M2 (bottom) on varying contamination rates across locations. In each subfigure, the left shows the posterior distributions of the observations on tumor sample, ζ^t, and the right shows that of the normal sample, ζ^n. Filled circles show the real ζ values of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distribution.

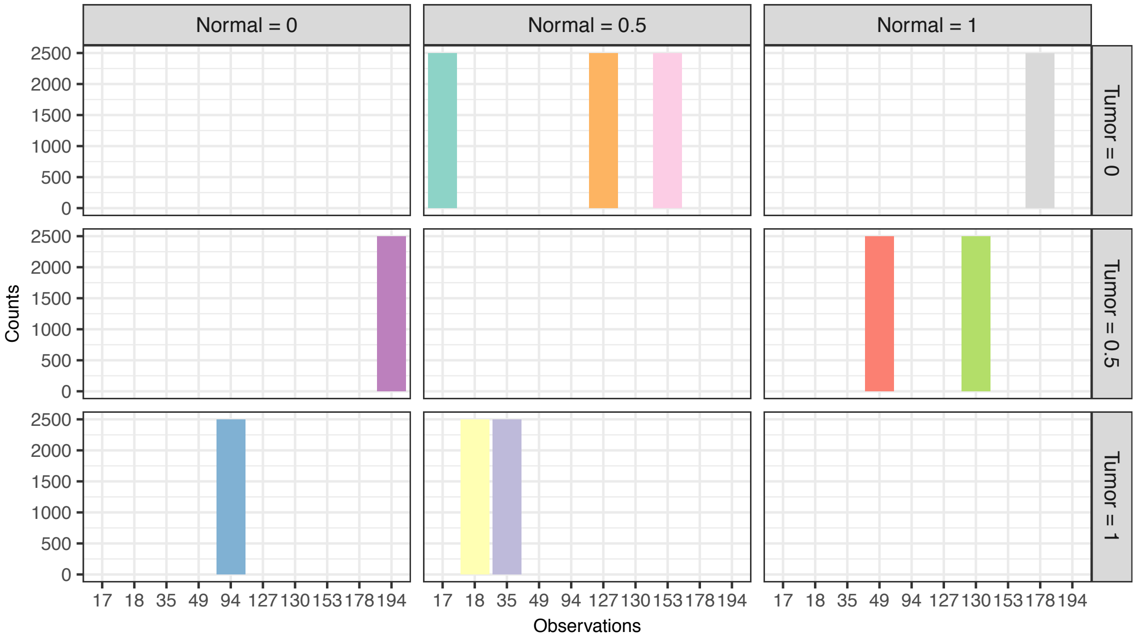


Figure 9.8.: Posterior distribution of the joint-genotype, (θ^n, θ^t), for M1 (result is similar for M2) on varying contamination rate across locations. To draw inference from this figure, observation 178 appear to have a joint-genotype of (1, 0), which, based on Table 8.1, suggest a somatic mutation.

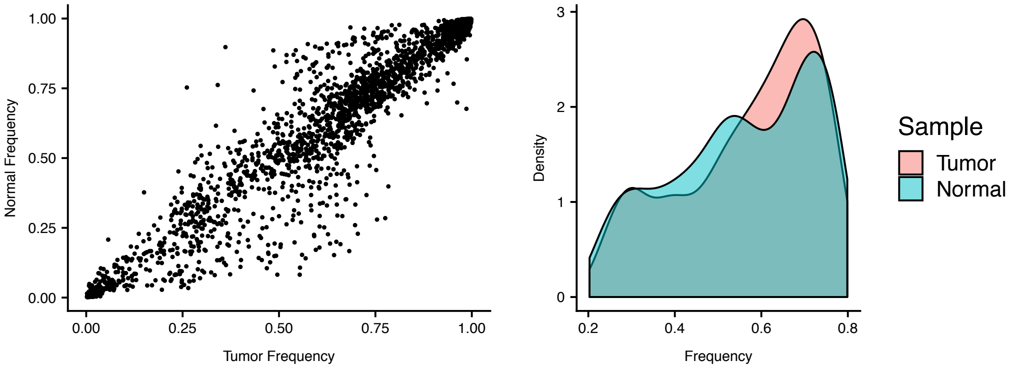


Figure 9.9.: Empirical probability of each location for tumor and normal samples (left) and density of the empirical probability after preprocessing step (right).

suggesting a near perfect positive correlation. For this reason, preprocessing of the raw data are commonly done and are necessary to find observations that contain valuable signals.

We filter for locations that have pair of frequencies (or empirical probabilities) between 0.2 and 0.8; our final data set contains 986 observations. Even after filtering was done, we see that the density of the two samples are still relatively similar (Figure 9.9), albeit with significant Kolmogorov-Smirnov test result. We use the same set of prior and hyperparameters as our previous two experiments, and ran the MCMC algorithm on 10,000 iterations with burn-in of 5,000. As shown by the previous two experiments, M2 is computationally more efficient. For this reason, we only implemented our model with a DP prior (M2) in this experiment.

Figure 9.11 shows the contamination estimate of the data. While there are few observations with signal contamination at the known level of 0.1 and 0.9, most observations have estimates around 0.3 and 0.7, which is both the upper and lower bound of the parameter space of ζ. This could indicate true contaminations at that level, or that the real estimate falls outside the parameter space, and the model finds the closest value that still satisfies the prior information.

Further investigation towards individual observations and its possible contamination estimate suggests that most observations are unidentifiable. As shown in Equation 8.4, when the probability of the binomial distributions are approximately equal across the two samples, the model is unidentifiable as there are infinitely many solutions to the contamination rate. This is apparent in Figure 9.12 and Table 9.4 with most observations having joint-genotype estimate of $(0.5, 0.5)$.

Furthermore, given the frequency of an observation, the possible ζ estimate does not satisfy our models assumptions. To illustrate, observation 367 in Figure 9.10 have a frequency of 0.575 in the tumor sample and 0.574 in the normal sample. We can compute its possible contamination levels through solving two simple linear equations and enumerating its joint-genotype. This is shown in Table 9.3. Recall the assumption of our models is that the contamination rates for the tumor sample should be less

Table 9.3.: Possible contamination rate given the frequencies at location 367

(θ^n, θ^t)	(0, 1)	(0.5, 1)	(1, 0)	(1, 0.5)	Frequency
ζ^t	0.425	0.850	0.575	0.150	0.575
ζ^n	0.426	0.852	0.574	0.128	0.574

Table 9.4.: Result of joint-genotypes on *in silico* data

θ^n/θ^t	AA	AB	BB
AA	0 (Wild-type)	195 (Somatic)	53 (Somatic)
AB	175 (LOH)	437 (Germline)	77 (LOH)
BB	35 (Error)	62 (Error)	0 (Germline)

than 0.5 and above 0.5 for the normal sample. Table 9.3 suggests that for this specific observation our model assumption is violated, or that it has the same joint-genotype, since given the frequencies, there is no possible solution to the contamination levels that would meet our assumptions, even when deviations in the empirical probabilities are considered.

The outcome seen in Table 9.4 may be biologically inaccurate, as we would not expect most of our DNA to be Germline. This means that it is also impractical for use in determining the contamination rate of a sample. These results indicate a need for more conservative preprocessing strategies to better retrieve relevent information from the data.

9.3 Application

The matched normal and tumor data used in this section is obtained from the NID Genomic Data Commons[1], a public benchmark data for detecting mutation or variation in the genome. These data are provided for the purpose of comparative

[1] `https://gdc.cancer.gov/resources-tcga-users/tcga-mutation-calling-benchmark-4-files`

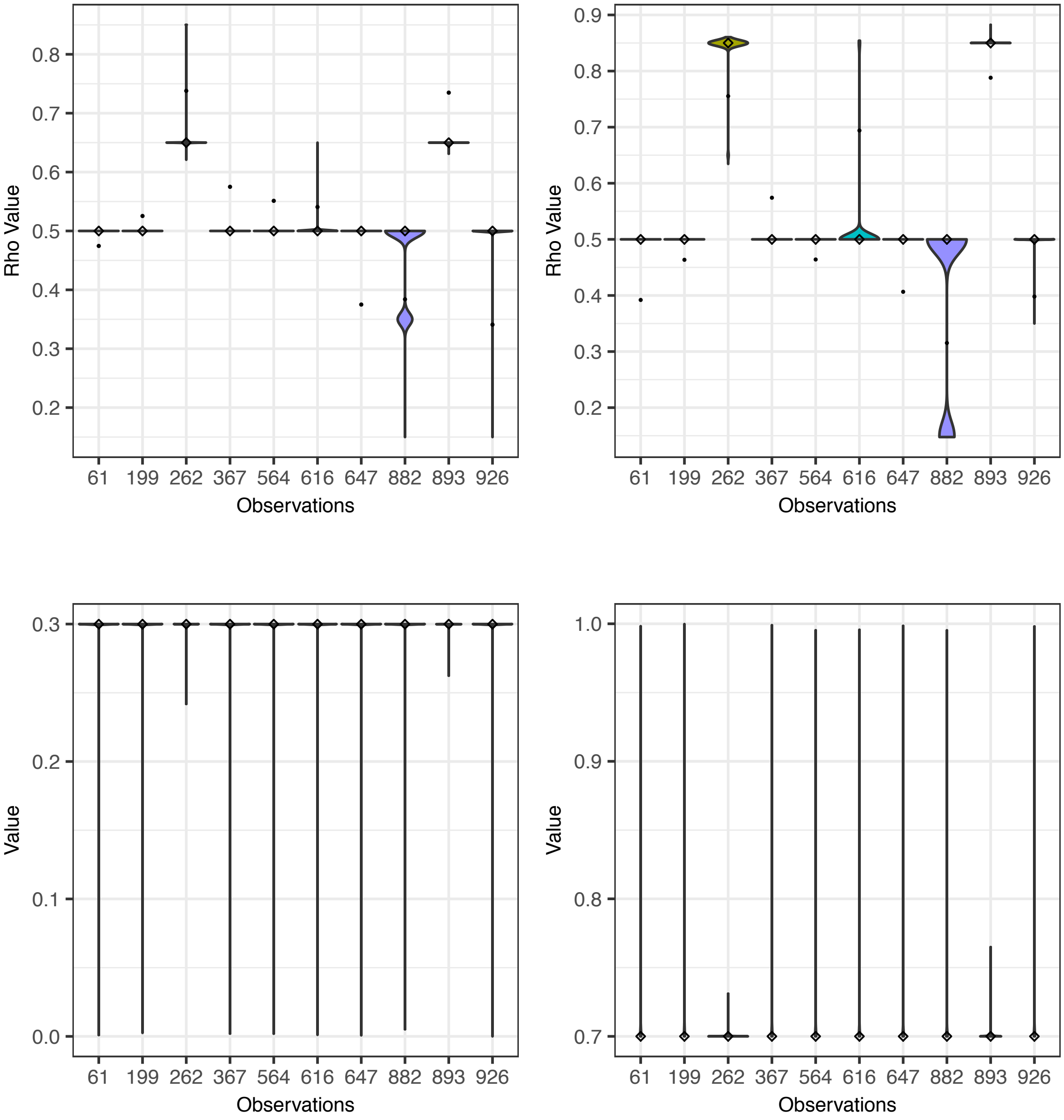


Figure 9.10.: Posterior distribution of ρ's (top) and ζ's (bottom) on 10 randomly selected observations for model M2 (top) on *in silico* data. The left shows the posterior of the observations on tumor sample and the right shows that of the normal sample. Filled circles show the empirical probability of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distribution.

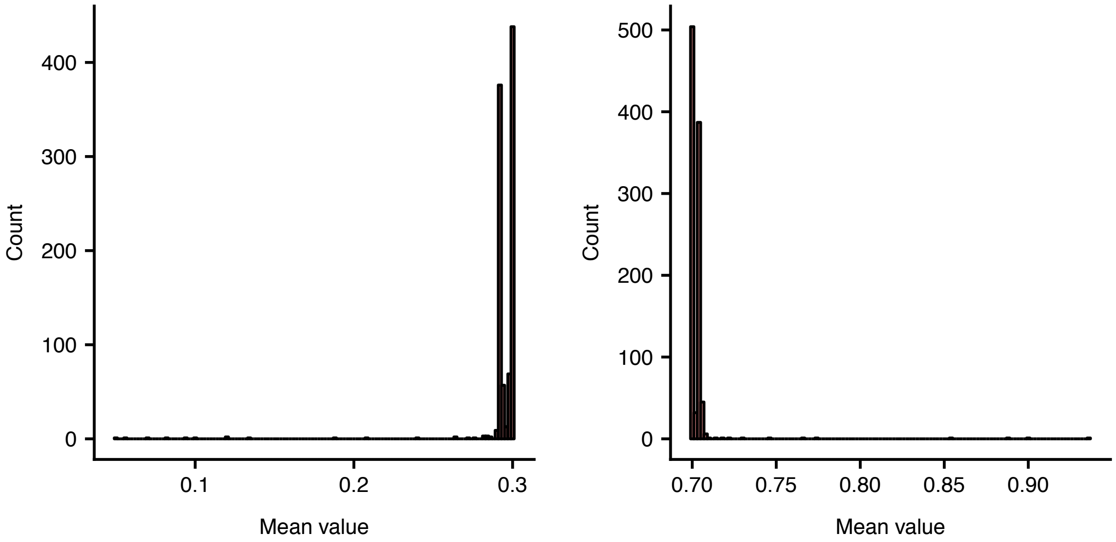


Figure 9.11.: Estimated contamination levels for tumor (left) and normal (right) on *in silico* data.

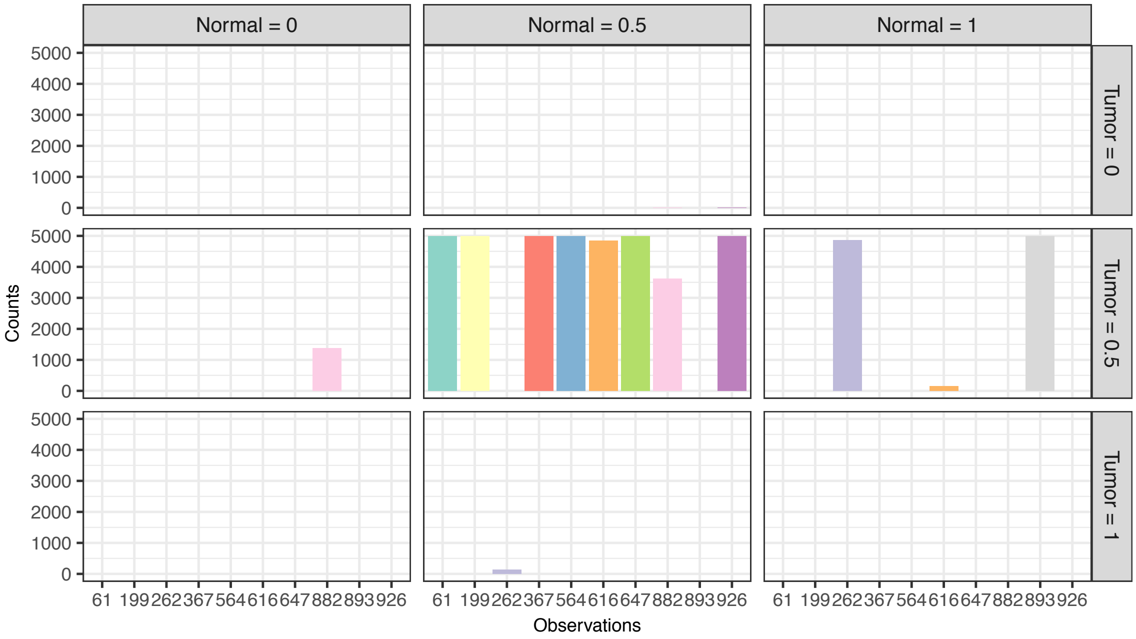


Figure 9.12.: Posterior distribution of the joint-genotype, (θ^n, θ^t), for M2 on *in silico* data. To draw inference from this figure, observation 61 appear to have a joint-genotype of (0.5, 0.5), which, based on Table 8.1, suggest a germline mutation.

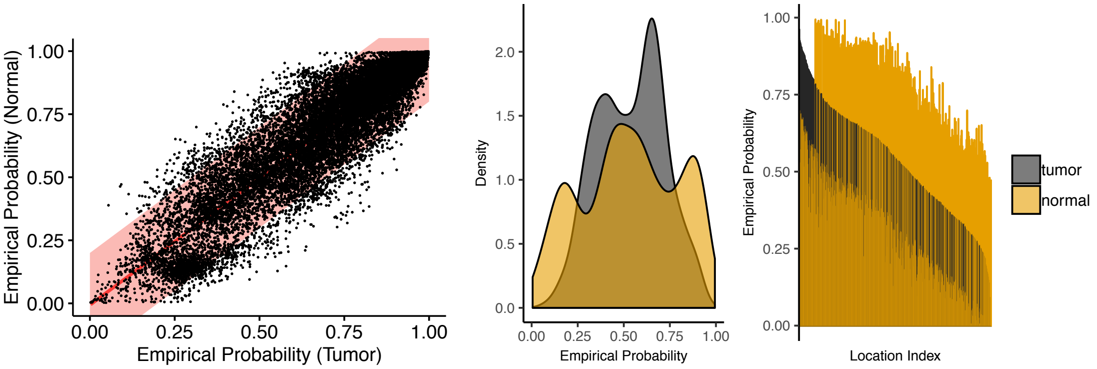


Figure 9.13.: Scatter plot of empirical probability of normal against tumor sample (left); points in shaded area are filtered out, the regression line is shown in red, and the Pearson correlation is 0.94. Density of the empirical probability after preprocessing (center), and ordered from highest to lowest, the empirical probability of the paired sample based on location (not actual locations in the genome).

evaluations of mutation calling algorithms, as the cell lines are publicly available for validation. While mixtures of the pure and normal are provided, at the time of the analysis, we were only able to secure the pure samples. Nevertheless, we implemented our model to the pure tumor and normal samples, and expect the contamination rate to be 0 for ζ^t and 1 for ζ^n. Each observation in both data set consists of a nucleotide location in the genome, the total number of reads mapped to this location, and the number of reads matching the reference genome from both the forward and backward strand of the DNA.

The raw data contains 23,822 common genome wide locations across the two samples. The empirical probability of location i in each sample, $\tilde{\rho}_i^t$ and $\tilde{\rho}_i^n$, is given by the ratio between the number of matched reads to the total number of mapped reads. We plot these values against one another to see the changes in the mutation profile between the two samples. Similar to the *in silico* data from our previous experiment, the left most figure in Figure 9.13 indicates very little shift in the frequency, as most locations lie along the diagonal regression line, suggesting a perfect linear relationship. Unlike the *in silico* data, there are scarcely any observations that cluster in the left bottom corner, with most lying in the top right corner (associated with a nearly perfect match), indicating that most locations closely match the reference genome. We hypothesize that for locations along the regression line (in red), the tumor causes little change to the genomic landscape, and therefore, provide weak signal of contaminations. Furthermore, given that the data contains pure normal and tumor samples, observations along the regression line will lead to the unidentifiability of the contamination estimate. To ensure identifiability, we included only points that lie as far away from the regression line as possible. For this reason, we took a more conservative strategy in the preprocessing step, compared to what was done on the *in silico* data.

We added an intercept term of 0.2 and -0.2 to the regression line and discarded all observations that lie between the two lines (shaded pink in Figure 9.13). The final data contains 1199 common locations, and the density of its empirical probability is given in Figure 9.13. The densities show that the normal sample contains more high and low frequencies, whereas the tumor sample contains more frequencies around half. The bar graph in Figure 9.13 presents these probabilities by location, sorted in decreasing order based on the tumor frequencies. This figure shows the locations along the genome that the tumor were able to surpress the number of matches to the reference (indicated by higher normal frequency).

Similar to that of the *in silico* experiment, we model this data using a DP prior (M2), and ran the MCMC algorithm for 10,000 iterations with a burn-in of 5,000. We set the same hyperparameter for the DP, but use a different value for the joint-genotype; to more closely reflect the mutational landscape, we adopted the hyperparameters used by Roth et al. (2012) given by Table 9.5.

The result shows a similar trend to that of the *in silico* data. Although there are a few contamination estimates at 0 and 1, most estimates of the contamination peaks at the upper boundary on the tumor sample (0.3) and lower boundary on the normal sample (0.7) of its parameter space (Figure 9.15). The posterior of ζ^j in Figure 9.14 shows that the algorithm has explored the parameter space but most estimates are around the boundary. More over, the scatter plot in Figure 9.13 is inconsistent with the known contamination rate of the data. With a ζ^t of 0 and ζ^n of 1, we should expect to see clusters at (θ_n, θ_t) where $\theta \in \{0, 0.5, 1\}$. However, we only observed clusters primarily at (1, 1) and along the regression lines, which again, will lead to unidentifiable estimates. Furthermore, the high number of errors in Table 9.6 is another indication of either inconsistency between the data and model assumptions, or better preprocessing step is needed to filter out locations before model implementation.

Table 9.5.: Hyperparameter for the Dirichlet prior on the joint-genotypes of pure normal and tumor data

θ^n/θ^t	AA	AB	BB
AA	10^6	10^2	10^2
AB	10^2	10^4	10^2
BB	1	1	10^4

Table 9.6.: Result of joint-genotypes on pure normal and tumor data

θ^n/θ^t	AA	AB	BB
AA	0 (Wild-type)	320 (Somatic)	186 (Somatic)
AB	159 (LOH)	50 (Germline)	61 (LOH)
BB	166 (Error)	257 (Error)	0 (Germline)

9.4 Summary

We tested both algorithms in a variety of experimental settings: uniform contamination, varying contamination, and *in silico* data with uniform contamination. In addition, we implemented our model on a pure matched tumor and normal sam-

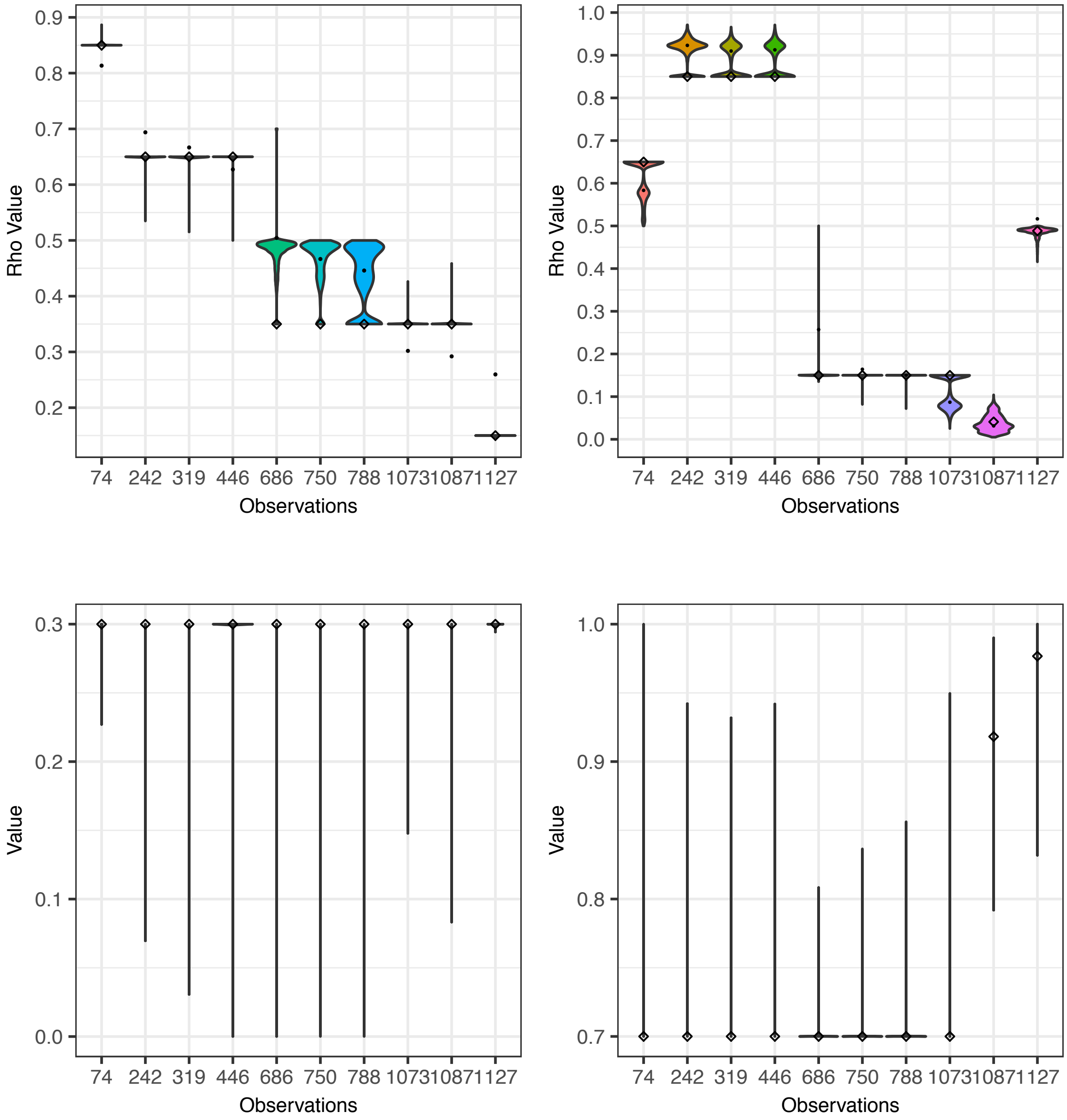


Figure 9.14.: Posterior distribution of ρ's (top) and ζ's (bottom) on 10 randomly selected observations for model M2 (top) for pure tumor and normal sample. The left shows the posterior of the observations on tumor sample and the right shows that of the normal sample. Filled circles show the empirical probability values of the observations, while diamonds represent the mode (for M1) or mean (for M2) of the posterior distributions.

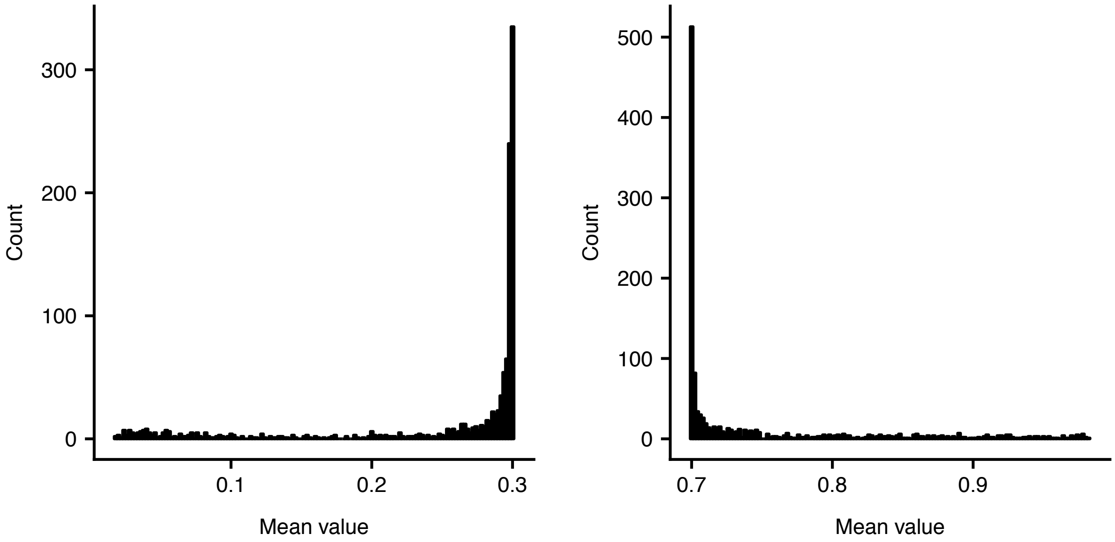


Figure 9.15.: Estimated contamination levels for tumor (left) and normal (right) on public data.

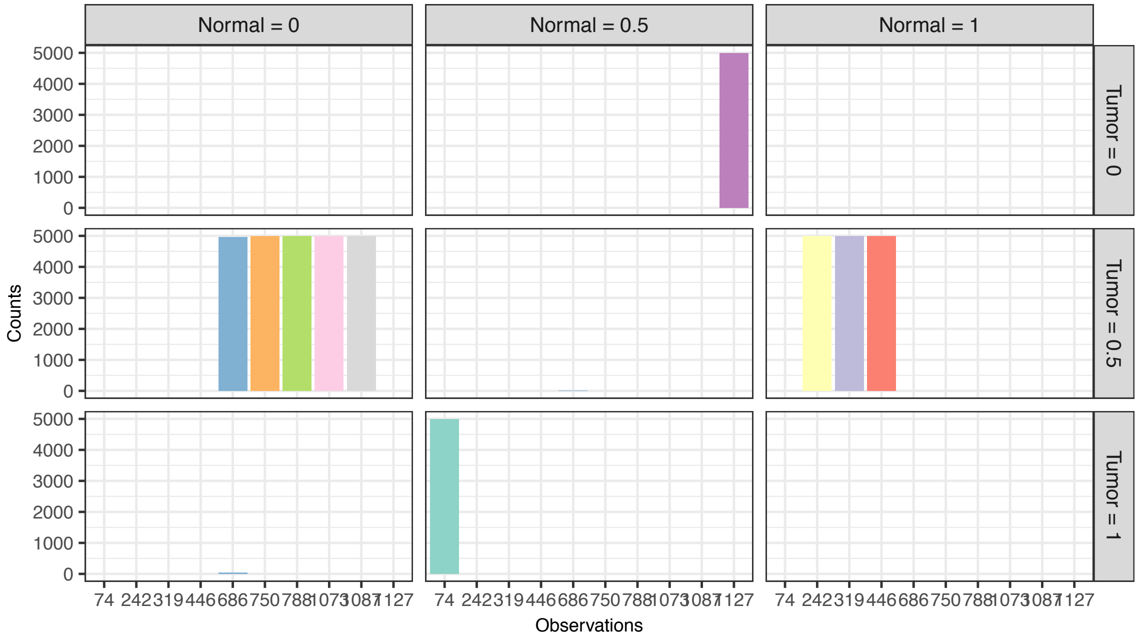


Figure 9.16.: Posterior distribution of the joint-genotype, (θ^n, θ^t), for M2 on pure normal and tumor data. To draw inference from this figure, observation 74 appear to have a joint-genotype of (0.5, 1), which, based on Table 8.1, suggest a LOH.

ple data, obtained from publicly available databases. For this data, we assume the contamination level to be 0 and 1, respectively.

In the first two experiments, we generated the data by randomly assigning appropriate contamination levels and joint-genotype for each observation in the dataset. Both models performed very well in the first two scenarios, with small error on the posterior estimate of all the parameters in the models. In term terms of efficiency, both experiments show that M2 is significantly faster by a factor of 100. Predictably, the grid size used in M1, highly impacts the accuracy and efficiency of the algorithm. The posterior distribution of the parameters for model M2 shows a smoother distribution, compare to that of M1.

When the models were implemented on *in silico* data, the estimate from the posterior distribution of both ζ^t and ζ^n were incompatible to the known contamination of the data. The same results were seen when the models were applied on the pure normal and tumor samples. In fact, most contamination estimates across the two samples were found to be at the boundary of the parameter space.

There are two potential reasons behind these discrepancies. First, preprocessing is an important step in evaluating modeling results, especially in the context of genomic dataset. In similar applications, for example, outside annotation is used to filter and exclude certain sites within the genome. Other methods in this area utilizes additional information to filter for sites along the genome. Specifically, Larson and Fridley (2013) and Su et al. (2012) only uses location along the genome that are heterozygous (i.e. AB sites); both Jun et al. (2012) and Cibulskis et al. (2011) first identifies homozygous loci (i.e. AA and BB sites) based on separate array data, using only these sites for purity estimation; lastly, Bergmann et al. (2016) uses a separate tool to filter for a pre-selected highly informative genomic markers. Additionally, these methods utilizes other information in their model that we did not consider, such as the probability of sequencing error per read (Jun et al., 2012; Bergmann et al., 2016; Cibulskis et al., 2011), as well as base calling probabilities.

Second, our modeling assumptions may be inconsistent with that of the real data due to technical noise. In particular, empirical analysis has shown sequencing experiments tend to incorrectly represent heterozygous loci mapped at rates < 0.5 (Su et al., 2012; Larson and Fridley, 2013), which impacts accurate estimation of tumor purities. This fact is inconsistent with our assumptions in placing a probability of $\theta^n = \theta^t = 0.5$ to heterozygous genotype AB. Understanding these potential issues as well as including technical variation within the modeling assumptions are promising directions for further investigation.

10. SUMMARY AND FUTURE WORK

10.1 Summary

This dissertation focused on solving two specific problems involving nonparametric Bayesian mixture models. First, we investigated the problem of density estimation for observations on constrained spaces. Data are often bounded by complex spatial domains such as the city of Chicago. Modeling these type of data without considering the domain, result in underestimation of observations lying at the edge of the boundary, as the model attempts to account for the sharp drop in the number of observations lying beyond the boundary. To mitigate this issue, we adopted the rejection sampling algorithm and showed that the observations can be generated by proposing from an unconstrained distribution and accepting proposals that fall in the constrained set. Based on the different proposal distributions used, we proposed two models: the truncated mixtures of Gaussians (TMoG) and mixtures of truncated Gaussians (MoTG). Chapter 3 described these models in more detail by providing a brief review of rejection sampling and how it can be used to generate data in more complex domains. This chapter also included comparisons between the two models by means of simulation and a toy example.

The resulting probability density of the two models, involve intractable normalizing constants that causes posterior sampling to be doubly-intractable. By characterizing both our models through a rejection sampling scheme, we utilized a data augmentation method that avoids computing intractable normalizing constants. Our methodology is most useful for nontrivial constraint sets, such as the boundaries of a city. It is also useful in simpler settings like the simplex, or the unit disc, when existing mixture models are not flexible enough to represent rich correlation structure or to capture sharp changes in probability across boundaries. In Chapter 4, we presented existing MCMC samplers for the data augmentation scheme and described how these samplers can be costly and exhibit poor mixing. We introduced a user defined parameter, ρ, that specifies the probability given by the prior to the constrained sets of interest. We translated this probability into threshold that quantifies the limit to the number of rejections per observations in the sampler. Also in this chapter, we provided justifications to this modified sampler and described how it is necessary to reduce variance and improve efficiency.

We tested our methods on various experimental settings in Chapter 5. In the first experiment involving a one-dimensional Gaussian, we illustrated how the likelihood of observations near the border can be underestimated when the constrained set is not accounted for. Our second experiment provided an example where higher thresholds may be required for data with sharp modes near the boundary. In the bivariate Gaussian experiment, we compared both models and showed that MoTG can fail with higher thresholds due to the coupling between the prior and number of rejections. In our last experiment, we evaluated the predictive performance of both samplers to a setting with more complex constraint and showed that under our scheme, the likelihood of observations near the edges improved significantly. Across all our experiments, we showed that TMoG is significantly faster and performs better than MoTG. Chapter 6 continues our implementation of both models to applications on two real datasets: flow-cytometry data and Chicago homicide data. We included posterior predictive checking for model diagnostics, and observed that the results on the comparison of both models to these data were consistent to our findings in Chapter 5.

In chapter 7, we provided a sensitivity analysis of our models and algorithms to the hyperparameters of the Dirichlet Process prior that were placed over the weights of the mixing components. We showed that smaller number of components result in lower predictive performance of observations near the boundary. We extended the rejection sampling scheme to include an outer limit on the perimeter of the constraint set to bound the location of the proposal draws from the prior. We showed through experimental testing that this modification further improved mixing while maintaining its efficiency.

Chapter 8 presented our approach to the second problem this dissertation addresses. DNA sequencing from tumor samples are susceptible to contamination by normal cells that leads to inaccuracies in the detection of mutations driving tumor growth. We motivated the problem in Chapter 8 and introduced the two models we developed to estimate the contamination rates in a matched normal and tumor sample. The first model involved a finite mixture model by placing a Dirichlet prior over the contamination parameter. We extended this model to the framework of infinite mixtures by placing a Dirichlet Process prior over the contamination parameter. Our models allow for varying contamination levels along the genome, as well as for the detection of somatic mutations. We discuss the models in detail and presented the associated MCMC algorithms for both models.

In Chapter 9, we implemented both models and algorithms on synthetic as well as real data. Our first experiment involved fixed contamination rate along the genome. We showed that the models can accurately estimate the observed probability, the contamination level, as well as the genotype of each location in the genome. In the second experiment, we implemented our models and algorithms to a setting with varying levels of contamination along the genome. The result was consistent to that of the first experiment where the sampler were able to accurately recover the correct parameter estimates for all observations. Across 20 different initializations, we showed that the model with the Diriclet Process prior was significantly faster than its counterpart.

In the application of our models and algorithms to public data with known contamination, we showed that there were some incompatibilities between the posterior estimates to the given contamination level. We described the two main reasons between these discrepancies. The first reason was the need for better preprocessing strategies such as filtering for only heterozygous or homozygous sites using separate array data or other stand alone tools. The second reason was the need to account for technical noise in our modeling assumptions. Multiple studies have shown that sequencing experiments can incorrectly represent the probability of a heterozygous loci at a site. Further understanding of the technical variation in the data can potentially improve modeling outcomes.

10.2 Future work

In this section, we outline the possible extensions to the models and algorithms we proposed to the two problems addressed in this dissertation.

10.2.1 Density estimation of data on constrained spaces

For the density estimation of data on constrained spaces, there are a couple of future directions that can be explored.

1. Improving the computational efficiency of the MCMC algorithms by employing a less crude way of checking whether a proposal falls in the constrained set. In our application to the Chicago homicide data, each proposal drawn from the prior is inspected across all 77 neighborhood limits of Chicago before it can be ruled as a rejection. This was done mainly because the data provided involved separate coordinate bounds for all 77 neigborhoods. One obvious way to reduce this overhead is to use only the outer coordinates of Chicago.

2. Future studies can extend our work to continuous state space models with restricted domains. In Chapter 3, we mentioned a specific dataset of coyotes in Rhode Island[1]. We can include time in addition to the geographical location to model the movements of these coyotes, that are restricted to the island. Each iteration of the Gibbs sampling procedure will include an additional step to sample from the conditional distribution over time. Many studies have model animal mobilities using continuous state space models (Hanks et al., 2015; John-

[1] `http://theconservationagency.org/narragansett-bay-coyote-study/`

son et al., 2008). However, these methods do not account for the limitation in the movement of the animals due to spatial restriction, which again, can underestimate the density near the boundary.

3. Our thresholding scheme serves to regularize the proposal distribution, limiting how much mass it assigns outside the constraint. It is interesting to look at more refined approaches to this, such as increasing the likelihood of truncation with distance from the constraint. We have briefly attempted to explore this idea in Chapter 7 but further theoretical and empirical studies are needed to fully understand the behavior of the proposal distribution in this setting.

4. The thresholded sampler presented in Chapter 4 describes a specific case of marginally specifiying the prior q. It would be of interest to implement the general marginal specification introduced by Kessler et al. (2015) in our sampler.

5. We can explore other Bayesian inference techniques, such as variational Bayes (Blei et al., 2017), to accurately estimate the probability density of observations on constrained spaces. It would also be promising to generate observations on complex constrained spaces using scalable framework such as the Variational Auto-encoder (Kingma and Welling, 2013).

10.2.2 Clustering mutations and estimating contaminations in matched normal and tumor samples

For the problem of estimating contamination levels in a matched normal and tumor sample, there are some immediate avenues of future improvements that we can explore.

1. The next focus would be to implement better strategies in preprocessing the data before applying the algorithms. As mentioned in Chapter 9, we want to filter out heterozygous location along the genome (Larson and Fridley, 2013; Su et al., 2012) to provide better identifiable estimation of both the contamination

and the joint-genotype. Further, we can modify our assumptions to the genotype probability to reflect technical noise that exist in these type of data (i.e. the probability of genotype AB is < 0.5 (Su et al., 2012; Larson and Fridley, 2013)). We can allow variation in the genotype probabilities by placing a prior over the parameters (Roth et al., 2012).

2. We can explore the use of nonparametric mixtures to determine clonal and subclonal mutations occuring in the tumor environment (Jiao et al., 2014; Roth et al., 2014).

REFERENCES

Inmaculada B Aban, Mark M Meerschaert, and Anna K Panorska. Parameter estimation for the truncated pareto distribution. *Journal of the American Statistical Association*, 101(473):270–277, 2006.

Daniel H Alai, Zinoviy Landsman, and Michael Sherris. Lifetime dependence modelling using a truncated multivariate gamma distribution. *Insurance: Mathematics and Economics*, 52(3):542–549, 2013.

Daniel J Bauer. Observations on the use of growth mixture models in psychological research. *Multivariate Behavioral Research*, 42(4):757–786, 2007.

Ewa A Bergmann, Bo-Juen Chen, Kanika Arora, Vladimir Vacic, and Michael C Zody. Conpair: concordance and contamination estimator for matched tumor–normal pairs. *Bioinformatics*, 32(20):3196–3198, 2016.

Alexandros Beskos, Omiros Papaspiliopoulos, Gareth O. Roberts, and Paul Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382, 2006.

David Blackwell, James B MacQueen, et al. Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355, 1973.

David M Blei and John D Lafferty. Topic models. In *Text Mining*, pages 101–124. Chapman and Hall/CRC, 2009.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Ryan Remy Brinkman, Maura Gasparetto, Shang-Jung Jessica Lee, Albert J Ribickas, Janelle Perkins, William Janssen, Renee Smiley, and Clay Smith. High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease. *Biology of Blood and Marrow Transplantation*, 13(6): 691–700, 2007.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Kevin Cain, Siobán Harlow, Roderick Little, Bin Nan, Matheos Yosef, John Taffe, and Michael Elliott. Bias due to left truncation and left censoring in longitudinal studies of developmental and disease processes. *American Journal of Epidemiology*, 173:1078–84, 2011.

Antonio Canale and Pierpaolo De Blasi. Posterior asymptotics of nonparametric location-scale mixtures for multivariate density estimation. *Bernoulli*, 23(1):379–404, 02 2017. doi: 10.3150/15-BEJ746. URL `https://doi.org/10.3150/15-BEJ746`.

Sha Cao, Tao Sheng, Xin Chen, Qin Ma, and Chi Zhang. A probabilistic model-based bi-clustering method for single-cell transcriptomic data analysis. *bioRxiv*, page 181362, 2017.

George Casella and Edward I George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Gilles Celeux. Bayesian inference for mixture: The label switching problem. In *Compstat*, pages 227–232. Springer, 1998.

Gilles Celeux, Merrilee Hurn, and Christian P Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

Hwan Chung, Eric Loken, and Joseph L Schafer. Difficulties in drawing inferences with finite-mixture models: a simple example with a simple solution. *The American Statistician*, 58(2):152–158, 2004.

Kristian Cibulskis, Aaron McKenna, Tim Fennell, Eric Banks, Mark DePristo, and Gad Getz. Contest: estimating cross-contamination of human samples in next-generation sequencing data. *Bioinformatics*, 27(18):2601–2602, 2011.

Adelino R Ferreira da Silva. A dirichlet process mixture model for brain mri tissue classification. *Medical image analysis*, 11(2):169–182, 2007.

David B Dahl. Model-based clustering for expression data via a dirichlet process mixture model. *Bayesian inference for gene expression and proteomics*, 4:201–218, 2006.

Kyle J DeMars and Moriba K Jah. Probabilistic initial orbit determination using gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 36(5):1324–1335, 2013.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Jean Diebolt and Christian P Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(2):363–375, 1994.

David R Easterling, Gerald A Meehl, Camille Parmesan, Stanley A Changnon, Thomas R Karl, and Linda O Mearns. Climate extremes: observations, modeling, and impacts. *science*, 289(5487):2068–2074, 2000.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

Sylvia Frühwirth-Schnatter. Markov chain monte carlo estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association*, 96(453):194–209, 2001.

Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC, 2006.

Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

Walter R Gilks and Pascal Wild. Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348, 1992.

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

Yeyun Gong, Qi Zhang, and Xuanjing Huang. Hashtag recommendation using dirichlet process mixture models incorporating types of hashtags. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 401–410, 2015.

Isobel Claire Gormley and Thomas Brendan Murphy. Exploring voting blocs within the irish electorate: A mixture modeling approach. *Journal of the American Statistical Association*, 103(483):1014–1027, 2008.

JM Górriz, A Lassl, J Ramírez, D Salas-Gonzalez, CG Puntonet, and EW Lang. Automatic selection of rois in functional imaging using gaussian mixture models. *Neuroscience letters*, 460(2):108–111, 2009.

Ephraim M Hanks, Mevin B Hooten, Mat W Alldredge, et al. Continuous-time discrete-space models for animal movement. *The Annals of Applied Statistics*, 9(1): 145–165, 2015.

Jiangang Hao, Benjamin P Koester, Timothy A Mckay, Eli S Rykoff, Eduardo Rozo, August Evrard, James Annis, Matthew Becker, Michael Busha, David Gerdes, et al. Precision measurements of the cluster red sequence using an error-corrected gaussian mixture model. *The Astrophysical Journal*, 702(1):745, 2009.

Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

Ajay Jasra. *Bayesian inference for mixture models via Monte Carlo computation*. PhD thesis, Imperial College London (University of London), 2006.

Ajay Jasra, Chris C Holmes, and David A Stephens. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, pages 50–67, 2005.

Yuan Ji, Chunlei Wu, Ping Liu, Jing Wang, and Kevin R Coombes. Applications of beta-mixture models in bioinformatics. *Bioinformatics*, 21(9):2118–2122, 2005.

Wei Jiao, Shankar Vembu, Amit G Deshwar, Lincoln Stein, and Quaid Morris. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *BMC bioinformatics*, 15(1):35, 2014.

Devin S Johnson, Joshua M London, Mary-Anne Lea, and John W Durban. Continuous-time correlated random walk model for animal telemetry data. *Ecology*, 89(5):1208–1215, 2008.

Goo Jun, Matthew Flickinger, Kurt N Hetrick, Jane M Romm, Kimberly F Doheny, Gonçalo R Abecasis, Michael Boehnke, and Hyun Min Kang. Detecting and estimating contamination of human dna samples in sequencing and array-based genotype data. *The American Journal of Human Genetics*, 91(5):839–848, 2012.

Michael Kalkbrener and Natalie Packham. Correlation under stress in normal variance mixture models. *Mathematical Finance*, 25(2):426–456, 2015.

David C Kessler, Peter D Hoff, and David B Dunson. Marginally specified priors for non-parametric bayesian estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(1):35–58, 2015.

Sangwoo Kim, Kyowon Jeong, Kunal Bhutani, Jeong Ho Lee, Anand Patel, Eric Scott, Hojung Nam, Hayan Lee, Joseph G Gleeson, and Vineet Bafna. Virmid: accurate detection of somatic mutations with sample impurity inference. *Genome biology*, 14(8):R90, 2013.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Athanasios Kottas and Bruno Sansó. Bayesian mixture modeling for spatial poisson process intensities, with applications to extreme value analysis. *Journal of Statistical Planning and Inference*, 137(10):3151–3163, 2007.

Adrien Lagrange, Mathieu Fauvel, and Manuel Grizonnet. Large-scale feature selection with gaussian mixture models for the classification of high dimensional remote sensing images. *IEEE Transactions on Computational Imaging*, 3(2):230–242, 2017.

Nicholas B Larson and Brooke L Fridley. Purbayes: estimating tumor cellularity and subclonality in next-generation sequencing data. *Bioinformatics*, 29(15):1888–1889, 2013.

Wei Li, Saurabh Prasad, and James E Fowler. Hyperspectral image classification using gaussian mixture models and markov random fields. *IEEE Geoscience and Remote Sensing Letters*, 11(1):153–157, 2013.

Albert Y Lo. On a class of bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, pages 351–357, 1984.

Xiaolin Luo, Pavel V Shevchenko, and John B Donnelly. Addressing the impact of data truncation and parameter uncertainty on operational risk estimates. *arXiv preprint arXiv:0904.2910*, 2009.

Micha Mandel. Censoring and truncation-highlighting the differences. *The American Statistician*, 61(4):321–324, 2007. ISSN 00031305.

Jeffrey A Manning and Caren S Goldberg. Estimating population size using capture–recapture encounter histories created from point-coordinate locations of animals. *Methods in Ecology and Evolution*, 1(4):389–397, 2010.

Eleni Matechou, François Caron, et al. Modelling individual migration patterns using a bayesian nonparametric approach for capture–recapture data. *The Annals of Applied Statistics*, 11(1):21–40, 2017.

Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

Paul D McNicholas and Thomas Brendan Murphy. Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26 (21):2705–2712, 2010.

I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366. AUAI Press, 2006.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Mark EJ Newman and Elizabeth A Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

Agostino Nobile and Alastair T Fearnside. Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, 17(2): 147–162, 2007.

Camilla L Nord, Vincent Valton, John Wood, and Jonathan P Roiser. Power-up: a reanalysis of'power failure'in neuroscience using mixture modeling. *Journal of Neuroscience*, 37(34):8051–8061, 2017.

Panagiotis Papastamoulis and George Iliopoulos. An artificial allocations based solution to the label switching problem in bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2):313–331, 2010.

Toby A Patterson, Len Thomas, Chris Wilcox, Otso Ovaskainen, and Jason Matthiopoulos. State–space models of individual animal movement. *Trends in ecology & evolution*, 23(2):87–94, 2008.

Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

Jim Pitman. Exchangeable and partially exchangeable random partitions. *Probability theory and related fields*, 102(2):145–158, 1995.

Jim Pitman. Poisson–dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition. *Combinatorics, Probability and Computing*, 11(5):501–514, 2002.

Vinayak Rao, Lizhen Lin, and David B Dunson. Data augmentation for models based on rejection sampling. *Biometrika*, 103(2):319–335, 2016.

Carl Edward Rasmussen. The infinite gaussian mixture model. pages 554–560, 2000.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387212396.

Carlos E Rodriguez and Stephen G Walker. Label switching in bayesian mixture models: Deterministic relabeling strategies. *Journal of Computational and Graphical Statistics*, 23(1):25–45, 2014.

Jeroen VK Rombouts and Lars Stentoft. Option pricing with asymmetric heteroskedastic normal mixture models. *International Journal of Forecasting*, 31(3): 635–650, 2015.

Andrew Roth, Jiarui Ding, Ryan Morin, Anamaria Crisan, Gavin Ha, Ryan Giuliany, Ali Bashashati, Martin Hirst, Gulisa Turashvili, Arusha Oloumi, et al. Jointsnvmix: a probabilistic model for accurate detection of somatic mutations in normal/tumour paired next-generation sequencing data. *Bioinformatics*, 28(7):907–913, 2012.

Andrew Roth, Jaswinder Khattra, Damian Yap, Adrian Wan, Emma Laks, Justina Biele, Gavin Ha, Samuel Aparicio, Alexandre Bouchard-Côté, and Sohrab P Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nature methods*, 11(4):396, 2014.

Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

Le Si Quang, Olivier Gascuel, and Nicolas Lartillot. Empirical profile mixture models for phylogenetic reconstruction. *Bioinformatics*, 24(20):2317–2323, 2008.

Adrian FM Smith and Gareth O Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23, 1993.

Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

Xiaoping Su, Li Zhang, Jianping Zhang, Funda Meric-Bernstam, and John N Weinstein. Purityest: estimating purity of human tumor samples using next-generation sequencing data. *Bioinformatics*, 28(17):2265–2266, 2012.

Putu Ayu Sudyanti and Vinayak Rao. Flexible mixture modeling on constrained spaces. *arXiv preprint arXiv:1809.09238*, 2018.

Cenny Taslim, Tim Huang, and Shili Lin. Dime: R-package for identifying differential chip-seq based on an ensemble of mixture models. *Bioinformatics*, 27(11): 1569–1570, 2011.

Yee Whye Teh. Dirichlet process. *Encyclopedia of machine learning*, pages 280–287, 2010.

Robert L Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

Kiran Tomlinson and Layla Oesper. Examining tumor phylogeny inference in noisy sequencing data. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 36–43. IEEE, 2018.

Eric P Xing, Michael I Jordan, and Roded Sharan. Bayesian haplotype inference via the dirichlet process. *Journal of Computational Biology*, 14(3):267–284, 2007.

Vinod Kumar Yadav and Subhajyoti De. An assessment of computational methods for estimating purity and clonality using genomic data derived from heterogeneous tumor tissue samples. *Briefings in bioinformatics*, 16(2):232–241, 2014.

Weixin Yao. Model based labeling for mixture models. *Statistics and Computing*, 22(2):337–347, 2012.

Yingdong Zhao and Richard Simon. Gene expression deconvolution in clinical samples. *Genome medicine*, 2(12):93, 2010.

VITA

Ayu was born in the tropical island of Bali, Indonesia. She received a Bachelor of Science in Mathematics from Sepuluh Nopember Institute of Technology (ITS) Surabaya in 2010. Upon graduation, she was granted the Fulbright scholarship to do a Masters in Financial Mathematics at the North Carolina State University (NCSU) which she completed in 2013. She then joined the Department of Statistics at Purdue University that same year. She earned a Masters of Science in Mathematical Statistics in 2016 and received her PhD in August 2019. Her research interests include Bayesian inference, machine learning, computational statistics, and statistical applications to genomics and health data. During her doctoral study, she briefly worked as a Research Assistant for the Nursing department before moving to work for the Purdue Center for Cancer Research where she stayed for over 2 years. In 2017, she did her graduate internship at Amgens' Design and Development Center. After graduation, she would join the Core Machine Learning group at Overstock as a Machine Learning Scientist.