The Neutral-to-the-Left Mixture Model

by

Sean La

B.Sc., Simon Fraser University, 2018

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

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL
STUDIES
(Statistics)

The University of British Columbia
(Vancouver)

August 2021

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

The Neutral-to-the-Left Mixture Model

submitted by **Sean La** in partial fulfillment of the requirements for the degree of **Master of Science** in **Statistics**.

Examining Committee:

Benjamin Bloem-Reddy, Statistics, UBC
Supervisor

Alexandre Bouchard-Côté, Statistics, UBC
Supervisory Committee Member

Abstract

A useful step in data analysis is clustering, in which observations are grouped together in a hopefully meaningful way. The mainstay model for Bayesian nonparametric clustering is the Dirichlet process mixture model, which has one key advantage of inferring the number of clusters automatically. However, the Dirichlet process mixture model has particular characteristics, such as linear growth in the size of clusters and exchangeability, that may not be suitable modelling choices for some data sets, so there is further research to be done into other Bayesian nonparametric models with characteristics that differ from that of the Dirichlet process mixture model while maintaining automatic inference of the number of clusters.

In this thesis, we introduce the *Neutral-to-the-Left mixture model*, a family of Bayesian nonparametric infinite mixture models which serves as a strict generalization of the Dirichlet process mixture model. This family of mixture models has two key parameters: the distribution of arrival times of new clusters, and the parameters of the stick breaking distribution, whose customization allows the user to inject prior beliefs regarding the structure of the clusters into the model. We describe collapsed Gibbs and Metropolis–Hastings samplers to infer the posterior distribution of clusterings given data. We consider one particular parameterization of the Neutral-to-the-Left mixture model with characteristics that are distinct from that of the Dirichlet process mixture model, evaluate its performance on simulated data, and compare these to results from a Dirichlet process mixture model. Finally, we explore the utility of the Neutral-to-the-Left mixture model on real data by applying the model to cluster tweets.

Lay Summary

This thesis describes the development of a statistical model to cluster data, the NTL mixture model, where the user has the ability to customize the characteristics of the clusters given by the model to match their understanding of the data. We consider one way to customize the statistical method so that it is meant for data points with time stamps, and so that under the model, clusters appear and then disappear in time. We develops algorithms to fit the model to data, and study the statistical method when used on simulated data. Finally, we apply the statistical method to cluster President Joe Biden's tweets.

Preface

This thesis is solely authored, unpublished work by the author, Sean La. The research topic and experiments were jointly designed with Profs. Benjamin Bloem-Reddy and Alexandre Bouchard-Côté. The code used to produce the experimental results for this thesis was created by Sean La, as well as the authors of the various software packages used as dependencies in the code. The code of the experiments can be found at the following link: https://github.com/realseanla/ntl-mixture-model.

Table of Contents

List of Tables

List of Figures

Acknowledgments

They say it takes a village to raise a child – I think the same can be said about educating a graduate student. During my university studies, I have been blessed with the support of a wide variety of people, people who have inspired me by their wisdom. This thesis is a culmination of their kindness and compassion, and I owe a great debt to all those who have had helped me along the way.

First, I would like to thank a large and formless governmental institution, NSERC, for funding my research in the form of an Alexander Graham Bell Canada Graduate Scholarship.

Next, to my colleagues. I would like to thank my graduate studies supervisors, Benjamin Bloem-Reddy and Alexandre Bouchard-Côté, for having provided me quality advice and guidance regarding my research despite unprecedented global circumstances. To my fellow graduate students Kevin, Kenny, Gian Carlo, Conor, Miguel, Anthony, Matteo, Rachel, Lulu, Evan, and Vittorio. I will always remember and appreciate the fact that it's almost Friday.

Now, to my friends Kasra, Albert, Jason, Maxwell, Isaac, Justin, Emre, Matthew, Elijah, Shirley, Donica, Max, Angela, Caitlin, Mircea, Mustafa, John, Harry, Taylor, and everyone else who I may have forgotten, for making the day-to-day grind bearable.

And finally, to my family, who are the reasons why I do what I do. To Vicky, my cousin, you have have been more of a sister to me than anything, and I look forward to calling you Doctor in a few years. To my baby cousins, Sophia and Elliot, although you guys are starting to become taller than me, you will always be babies in my eyes. And last, but not least, to my mother, who climbed impassable mountains, crossed torrentuous oceans, and fought inexplicable monsters, so that

I, and all those who come after, may live better lives.

I think it's going to work out.

Chapter 1

Introduction

In the day-to-day work of the humble data analyst, they are often bombarded with dredges of data and are asked by the powers that be to create insight from an otherwise chaotic heap of numbers. One common way to reduce the complexity of the data to be analyzed is to create natural groups of the observations (also called *clusterings*) that hopefully will aid the analyst in their interpretation. When a clustering algorithm is used to group the data, the algorithm is often said to be "learning" a clustering in an "unsupervised" way, since ideally, the algorithm receives little input from the user on how to create the clusters, as automating the process of discovering clusters is what the analyst is interested in in the first place!

These unsupervised learning algorithms are a mainstay in the data analytics toolbox, and for decades there has been active research in the development of unsupervised learning algorithms. This thesis describes a general class of clustering algorithms, and in particular, an algorithm for discovering clusters with bounded expected size in nonexchangeable data, which are formulated in the language of Bayesian inference and Bayesian nonparametrics. The Bayesian nonparametric model that this family of clustering algorithms is based on is called the Neutral-to-the-Left (NTL) mixture model.

Formally in the language of statistics, clustering is the task of inferring the values of a set of latent discrete-valued random variables $\mathbf{Z}_n = (Z_i)_{i=1}^n$ given a set of real and possibly vector-valued observations $\mathbf{X}_n = (X_i)_{i=1}^n$. The unique values taken on by $\mathbf{Z}_n$ are said to be the "clusters" of the observations, in which $Z_i = Z_j$

implies that observations i and j are in the same cluster. Signal indicating which set of observations are contained in the same cluster is modelled as being manifested in the observations $\mathbf{X}_n$, with $Z_i = Z_j$ implying that X_i is "similar" to X_j, and $Z_i \neq Z_j$ implying that X_i is "dissimilar" to X_j. The exact mathematical definition of what constitutes "similarity" depends on the specific statistical model considered at hand, with one common choice being that X_i and X_j are distributed according to the same probability distribution if i and j are in the same cluster.

Our goal with the NTL mixture model is to describe a class of statistical models of clusterings where the number of clusters is unknown, and which allow the user to inject prior knowledge of the characteristics of the clusters into the model. Specifically, in this thesis, we explore one parameterization of the NTL mixture model meant for nonexchangeable data which induces groupings containing clusters whose expected sizes are bounded, and which only commonly occur within short subsequences throughout the whole sequence of observations. If considering temporally-ordered data, this phenomena can be interpreted as there being small clusters that appear for only a short duration in time. We develop two algorithms for inferring the posterior distribution of cluster assignments $\mathbf{Z}_n$ given observations $\mathbf{X}_n$. We then demonstrate that sampling algorithms for the NTL mixture model can more adequately recover the data-generating clustering in comparison to the well-known Dirichlet process mixture model (DPMM) when the data is generated from the prior of the specified parameterization of the NTL mixture model. Finally, we showcase the utility of the NTL mixture model by using it to cluster a Twitter dataset with timestamps.

In Section 1.1, we describe work related to the NTL mixture model. In chapter 2, we give background regarding Bayesian nonparametrics, the Dirichlet process (DP) and its applications in clustering, and describe the Beta Neutral-to-the-Left model of random graphs, the statistical model that the NTL mixture model is based on. Chapter 3 contains descriptions and derivations of algorithms to infer the various parameters of the NTL mixture model. Chapter 4 showcases the utility of the NTL mixture model on both simulated and real data. Lastly, Chapter 5 concludes the thesis and describes various directions for future research.

1.1 Related Work

The precursor to the NTL mixture model, the Beta Neutral-to-the-Left (BNTL) for random graphs (then named $(\alpha\text{-}T)$-*graphs*), was originally proposed by Bloem-Reddy and Orbanz [1]. Under certain parameterizations, they show that the BNTL model can induce graphs which are sparse, which is related to the notion of microclustering mentioned in the next paragraph. Bloem-Reddy et al. [2] described both Bayesian and frequentist inference algorithms for BNTL graphs, which some of our inference algorithms are based on.

The specific parameterization of the NTL mixture model considered in this thesis leads to clusters whose expected sizes are *a priori* finite, which is related to the notion of *microclustering*, a concept first introduced by Betancourt et al. [3]. A clustering is said to be a *microclustering* if the size of the largest cluster grows sublinearly in the total number of observations in the model. The models described by Betancourt et al. are suspected to exhibit microclustering in general, but the property is only demonstrated to hold in special cases, and is shown to hold experimentally otherwise. Betancourt et al. [4] then describe a clustering model which they show theoretically exhibits microclustering.

Griffin and Steel [5] describe a model for changepoints in time series data based on the Dirichlet process. In their *stick-breaking autoregressive process*, at each point t in continuous time $[0,T]$, observations are distributed according to a mixture

$$G_t = \tilde{G}_{N(t)}$$

where $N(t)$ is a Poisson process over the interval $[0,t]$, and the $\tilde{G}_s$ is defined recursively as

$$\tilde{G}_s = (1-V_s)\tilde{G}_{s-1} + V_s\delta_{\theta_s}.$$

Therefore, the appearance of changepoints in the time series data is modelled as the arrival of new atoms θ_s, where the arrival times are dictated by some Poisson process. This recursive stick breaking procedure bears a striking resemblance to the NTL mixture model to be introduced in later sections, with the key difference being that the autoregressive stick breaking process models observations in continuous time, whereas the parameterization of the NTL mixture model we consider in this

thesis is meant for discrete time.

Chapter 2

Background

Bayesian nonparametrics is a subfield of Bayesian statistics that deals with the development, inference, and application of *Bayesian nonparametric models* – statistical models that have an unbounded number of parameters which are inferred using Bayesian techniques [6]. Bayesian nonparametric models are seen to be more flexible than conventional finite-parameter parametric statistical models, since the potential infinitude of parameters allows nonparametric models to grow in complexity as the number of observations increases.

From a probabilistic point of view, nonparametric priors are *stochastic processes* – collections of random variables $(X_t)_{t\in\mathscr{I}}$ which are indexed by a parameter t (often interpreted as "time") that takes on values from a possibly infinite index set $\mathscr{I}$. The distributions of the observed random variables X_t are assumed to be endowed with some correlation structure that allows for tractable inference, such as, for example, the existence of latent random variables so that subsets of the observations X_t are iid conditionally on the latent variables.

Some examples of Bayesian nonparametric models include Gaussian processes for regression tasks [7] and the Indian buffet process [8] for latent feature allocation. Perhaps the most famous Bayesian nonparametric model is the Dirichlet process, which is often used in mixture models and variations thereof.

2.1 The Dirichlet Process

The Dirichlet process (DP) is a stochastic process whose sample paths are discrete probability distributions themselves. There are three main ways to characterize the DP, which we detail below.

First, an implicit definition [9]. Formally, consider a measurable set $\mathscr{S}$, a probability measure H on $\mathscr{S}$, and a real number $\alpha > 0$. The Dirichlet process $\mathrm{Dir}(\alpha, H)$ is a stochastic process so that $X \sim \mathrm{Dir}(\alpha, H)$ satisfies

$$(X(B_1), X(B_2), \ldots, X(B_n)) \sim \mathrm{Dirichlet}(\alpha H(B_1), \ldots, \alpha H(B_n))$$

for any finite disjoint partition $(B_i)_{i=1}^n$ of $\mathscr{S}$. Although this definition does not give a construction of a realization of $\mathrm{Dir}(\alpha, H)$, we can rest assured that such a stochastic process exists by invoking *Komologorov's extension theorem* [10], which states that it is enough to define a stochastic process by a suitably consistent set of marginal distributions on finite subsets of indices.

Although it suffices to define the DP by its finite dimensional marginal distribution, this is not so helpful from a practical point of view, as the implicit definition gives us no information regarding how to compute quantities related to the DP, such as e.g. drawing realizations from it. We can explicitly characterize a realization of a probability distribution from the DP using the stick breaking representation [11]. It can be shown that a realization of the DP is a discrete distribution of the form

$$P(X \in \cdot \,|\, (\beta)_{i=1}^{\infty}, (x_i)_{i=1}^{\infty}) = \sum_{i=1}^{\infty} \beta_i \delta_{x_i}(\cdot),$$

where $x_i \overset{\text{iid}}{\sim} H$. The mixture weights $(\beta_i)_{i=1}^{\infty}$ can be expressed in the following form:

$$\beta_1 = \psi_1, \; \beta_i = \psi_i \prod_{j=1}^{i-1} (1 - \psi_j), \; i > 1$$

where $\psi_j \overset{\text{iid}}{\sim} \mathrm{Beta}(1, \alpha)$ for some $\alpha > 0$. The distribution over $(\beta_i)_{i=1}^{\infty}$ is called the Griffiths–Engen–McCloskey (GEM) distribution, which we denote by $\mathrm{GEM}(\alpha)$ for $\alpha > 0$ [9]. This representation is reminiscent of breaking a stick of unit length into smaller pieces, where we start off by breaking off a piece with length ψ_1,

and then proceed to break off more sticks from the remaining piece with fractional lengths ψ_j.

The final representation of the DP concerns the characterization of the conditional distribution of draws from the DP [9]. Consider a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, a base probability measure H on this space whose support may possibly be uncountable, and a real value $\alpha > 0$. Consider a stochastic process characterized by the following procedure.

Define $n_x = \#\{i \in \{1,\ldots,n\} : X_i = x\}$. For $n \geq 0$:

1. Draw a new value $X_{n+1} \overset{iid}{\sim} H$ with probability $\frac{\alpha}{n+\alpha}$.
2. Set $X_{n+1} = x$ with probability $\frac{n_x}{n+\alpha}$ for some previously assigned value x.

The sequence $(X_n)_{n\geq 1}$ is then a sequence of draws from a realization of the DP. This can be proved by observing the fact that the distribution of $(X_n)_{n\geq 1}$ is exchangeable (although it is easy to see from the generative procedure that it is not independent) [9]. Therefore, by de Finetti's theorem, we have that there exists a random probability measure P so that the observations X_i are conditionally independent given P [9]. More concisely, we have

$$P \sim \text{Dir}(\alpha, H)$$
$$X_i \overset{\text{iid}}{\sim} P \text{ for } i \geq 1.$$

Although de Finetti's theorem states the existence of P, it does not give an explicit construction of P.

This characterization of conditional draws from the DP is often described by two metaphors: the Chinese Restaurant Process, and the Polya Urn model [12].

In the Chinese Restaurant Process [13], consider an infinite sequence of tables within a Chinese restaurant, which are labelled with values $n \in \mathbb{Z}$. Consider also an infinite sequence of customers who wish to sit at these tables, with Z_i denoting the table that customer i chooses to sit at. For a particular customer i, they will sit at table n with probability proportional to the number of customers already sitting at that table, and will sit at a new table n' with probability proportional to α.

On the other hand, in the Polya Urn model (also known as the Blackwell-MacQueen urn scheme) [14], consider an urn that initially has α balls coloured

white (assuming that α is an integer, of course). At each step in the process, draw a ball from the urn. If the ball is white, add a ball of a new colour into the urn. Otherwise, add a new ball with the same colour of the drawn ball into the urn.

2.2 Applications of the Dirichlet Process

2.2.1 Dirichlet Process Mixture Model

Owing to the fact that realizations of the DP are discrete probability measures with probability 1, the classic application of the DP in data analysis is clustering. Clustering is the task of creating partitions of a data set so that data points within groups are more similar than data points across groups. There are a multitude of algorithms for cluster analysis whose theoretical foundations are based on ideas from a wide variety of fields such as graph theory and statistics. One crucial task in cluster analysis is determining the number of distinct groups. Some algorithms, such as correlation clustering, determine this quantity as part of the procedure [15]. However, other popular algorithms, such as k-means clustering and hierarchical clustering, require that the user provides the number of clusters as a parameter [16]. For low dimensional data, the number of distinct clusters is sometimes obvious after visualizing the data. But in more complicated scenarios, such as high dimensional data, the number of clusters is often times something that the analyst wants to infer.

In a Dirichlet process mixture model (DPMM), each cluster corresponds to an atom in the distribution $P(X \in \cdot) = \sum_{i=1}^{\infty} \beta_i \delta_{x_i}$, where in particular, x_i is interpreted as the "parameter" of the cluster [6]. For example, if we model the observations as being Gaussian conditioned on their cluster assignment, we might set $x_i := (\mu_i, \Sigma_i)$, where μ_i is the mean of the (possibly multivariate) Gaussian distribution, and Σ_i is the covariance. The base measure H could then be a Normal-Inverse-Wishart distribution, which would serve as a conjugate prior for the Gaussian likelihood of the data [17]. In practice, H often serves as the possibly conjugate prior of the likelihood, so that the posterior distribution over the parameters of the components can be conveniently Gibbs sampled. Moreover, the posterior distribution of the set of realized components of the DP can be easily sampled from, owing to the fact

that the DP is self-conjugate [18]; that is, given $P \sim \text{Dir}(\alpha, H)$, $X_1, \ldots, X_n \sim P$, we have that

$$P|X_1, \ldots, X_n \sim \text{Dir}\left(\alpha + n, \frac{\alpha}{n+\alpha}H + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta_{X_i}\right).$$

The Hierarchical Dirichlet Process

Data analysts may face situations where they would like to create multiple clusterings using the DP in such a way so that the clusterings share the same set of atoms $S \subseteq H$, albeit with different weights across clusterings.

The hierarchical Dirichlet process (HDP), first described by Teh et al. [19] fulfills this need by modelling the multiple clusterings G_j as being realizations of Dirichlet processes $\text{Dir}(\alpha_j, G)$ that share a base measure G, which itself is the realization of a Dirichlet process $\text{Dir}(\alpha, H)$.

More precisely, the generative model described by a HDP is given by the following:

$$\begin{aligned} G &\sim \text{Dir}(\alpha, H) \\ G_j &\sim \text{Dir}(\alpha_j, G) \text{ for } j = 1, \ldots, k \\ X_{kj} &\sim G_j \text{ for } k = 1, \ldots, n_j. \end{aligned} \tag{2.1}$$

Because $G = \sum_{i=1}^{\infty} \beta_i \delta_{x_i}$ is a discrete distribution, the G_j distributions will also be discrete, and each G_j will share the same set of atoms from G, and place non-zero probabilities on the atoms x_i. Therefore, we have a multiple clustering model where the clusterings share the same parameters, but with different mixture weights.

Dirichlet Process Hidden Markov Multiple Changepoint Model

One common task when dealing with streaming time series data is detecting instances when the underlying distribution of the data changes, i.e. changepoint detection. For a particular data set, the analyst may be interested in detecting multiple changepoints within their data. This multiple changepoint detection problem can be formulated as a clustering problem, subject to the constraint that only a single cluster is present at any time in the time series data. If one wishes to construct a Bayesian nonparametric method for changepoint detection in time series data, the Dirichlet process is the natural prior to use for such a task.

Ko et al. [20] introduce the Dirichlet Process Hidden Markov Multiple Changepoint Model, which allows for the detection of an arbitrary number of changepoints in the data. Suppose that $s_1, s_2, \ldots, s_{t+1}$ correspond to the latent states of a sequence of observations $X_1, \ldots, X_{t+1}$. The most natural representation of the model is the Chinese Restaurant Process-style assignment conditional probability

$$P(s_{t+1} = j | s_t = i, s_{t-1}, \ldots, s_1) = \begin{cases} \frac{n_i - 1 + \beta}{n_i - 1 + \beta + \alpha} \text{ if } j = i, \\ \frac{\alpha}{n_i - 1 + \beta + \alpha} \text{ if } j = i + 1 \end{cases} \tag{2.2}$$

where $\alpha, \beta > 0$, and n_i is the number of observations assigned to state i. The above conditional probability can be seen as almost a special case of the DP conditional predictive rule when there is only a single existing cluster. This conditional predictive rule differs slightly, however, from the conditional probability for the conventional DP in that the numerator of the conditional probability of assigning to an existing state is associated with an arbitrary initial mass of $\beta - 1$ instead of the usual mass of 1 in the DP, though of course we can set $\beta := 2$ to match it with the DP.

2.3 The Beta Neutral-to-the-Left Model of Sparse Graphs

Graphs are useful tools for representing multiple relationships between objects, such as social networks or interactions between proteins within a biological network. If one is interested in creating a statistical model of graphs, a naive first pass approach is to assume exchangeability over vertices. However, this seemingly reasonable assumption leads to an insidious side effect – the Aldous–Hoover theorem implies that such graphs represented by vertex exchangeable models are necessarily dense (the number of edges grows on the order of at least quadratically in the number of vertices) [21]. However, many real-world networks exhibit sparsity, in which the number of edges grows at most linearly in the number of vertices. One useful property that can be represented by sparse graphs is power law on the distribution of the degrees of vertices, which is also empirically present in many graphs seen in the wild. A graph exhibiting power law in vertex degrees has the prop-

erty that the number of vertices whose degree is d is proportional to some negative power of d (i.e., given a graph G, we have that $\#\{v \in V(G) : \deg(v) = d\} \propto d^{-\eta}$, where η is some positive real number).

Some common statistical models of random graphs assume that the vertices of the graph are exchangeable, which restricts the class of random graphs representable by these models to be dense (or empty) [22]. Orbanz and Roy [21] posed the (then) open problem of constructing a statistical model of sparse random graphs that maintains some notion of probabilistic symmetry in the model.

Bloem-Reddy and Orbanz [1] introduce a class of random graphs which exhibit sparsity and power law behaviours, while maintaining a useful form of probabilistic symmetry in the form of *left-neutrality*, which is then called Beta Neutral-to-the-Left (BNTL) models in [2]. In this model, graphs are represented as a sequence of edge ends $(Z_1, Z_2, \ldots)$, where $Z_i \in \mathbb{N}$. BNTL models have a representation reminiscent of the stick-breaking presentation of the DP. Suppose that Λ is some base distribution which describes the arrival times of vertices in the graph. We have that a BNTL model has the following generative model for the edge assignments $(Z_1, Z_2, \ldots)$:

$$\begin{aligned} T_j &\sim \Lambda, \\ \psi_j | T_j &\overset{\text{iid}}{\sim} \text{Beta}(1-\alpha, T_j - 1 - (j-1)\alpha) \text{ for } j \geq 1 \\ P_{j,K_n} &= \psi_j \prod_{\ell=j+1}^{K_n} (1-\psi_\ell) \\ Z_n &\sim \begin{cases} \delta_{K_n}(\cdot) & \text{for } n = T_{K_n} \\ \text{Categorical}(P_{j,K_n}) & \text{otherwise} \end{cases} \end{aligned} \tag{2.3}$$

where K_n is the number of vertices in the graph when the nth edge end has been assigned a vertex Z_n.

This model exhibits *neutrality-to-the-left* in the recursive stick breaking increments $(P_{j,K_n})_{j=1}^{K_n}$; that is, we have that the random variables R_{j,K_n} defined by

$$R_{j,K_n} := \frac{P_{j,K_n}}{\sum_{i=1}^{j} P_{i,K_n}} = \frac{\psi_j \prod_{\ell=j+1}^{K_n}(1-\psi_\ell)}{\sum_{i=1}^{j} \psi_i \prod_{\ell=i+1}^{K_n}(1-\psi_\ell)} = \psi_j$$

are mutually independent for all $j = 1, \ldots, K_n$.

The probability of a vertex in a BNTL model has a recursive stick breaking representation similar to the probability of a cluster location in the stick-breaking representation for the DP. The two representations differ in the *direction* of the recursive stick breaking, however; in the DP stick breaking representation, the probability of all clusters younger than a cluster j are constrained to be no greater than the complement probability $(1 - \psi_j)$. On the other hand, in the BNTL model, the probability of all vertices *older* than a vertex j can be no greater than the complement probability $(1 - \psi_j)$; therefore the probability of a particular vertex degrades over time as new vertices arrive in the graph.

Another interesting characteristic of the BNTL stick breaking model that differentiates it from the DP stick-breaking representation is that the distribution of the arrival times of vertices is an explicit parameter of the model; the arrival of vertices can be made faster or slower than the DP by the choice of arrival time distribution. Indeed, this explicit parameterization of the arrival time distribution in the BNTL model allows for modelling of sparse graphs – setting the interarrival times of vertices (defined to be $\Delta_j := T_j - T_{j-1}$) to be distributed via $\text{Geom}(p)$, for example, yields graphs generated by the BNTL model that are sparse.

Although the BNTL process differs from the DP in the direction of its stick breaking representation, another surprising fact is that the DP is actually a special case of the BNTL process. Conditioned on a sequence of strictly increasing vertex arrival times $\mathbb{T} = (T_1, T_2, \ldots) \sim \Lambda$, [2] shows that the predictive rule for the assignment of an incoming edge end is

$$P(Z_{n+1} \in \cdot | \mathbf{Z}_n, \mathbb{T}) = \mathbf{1}(n+1 = T_{K_n+1}) \delta_{K_n+1}(\cdot) + \mathbf{1}(n+1 < T_{K_n+1}) \sum_{j=1}^{K_n} \frac{d_{j,n} - \tau}{n - K_n \tau} \delta_j(\cdot).$$

Recall that the predictive rule for the cluster assignment of a new observation for the Dirichlet Process is expressed as

$$P(Z_{n+1} \in \cdot | \mathbf{Z}_n) = \frac{\alpha}{n+\alpha} \delta_{K_n+1}(\cdot) + \frac{n}{n+\alpha} \sum_{j=1}^{K_n} \frac{n_j}{n} \delta_j(\cdot)$$

We see that the predictive rule for the DP is a special case of the BNTL predictive

rule with $\tau := 0$ and Λ set to be arrival time distribution of the DP.

Although originally formulated to address sparsity of graphs, the BNTL model has a natural connection with random partitions of the natural numbers. Indeed, every random graph induced by the generating procedure of the BNTL corresponds one-to-one to a partition of the natural numbers. Graphs with $\eta > 2$ correspond to partitions where the size of blocks grow sublinearly in the number of observations, and graphs with $\eta \in (1,2)$ correspond to partitions where the size of blocks grow linearly in the number of observations.

Chapter 3

The Neutral-to-the-Left Mixture Model

In this section, we introduce a novel infinite mixture model motivated by the Beta Neutral-to-the-Left (BNTL) model for graphs introduced by Bloem-Reddy and Orbanz [1]. Recall that the BNTL model for graphs can be described by a neutral-to-the-left stick breaking procedure where the arrival time distribution of new components (vertices in the context of graphs) is an explicit parameter of the model. The Dirichlet process can be seen as a special case of the BNTL process with a specific parameterization of the arrival time distribution. Given that the Dirichlet process is extensively used as a prior in infinite mixture models, a natural question to ask is how can the BNTL process be used as a prior in Bayesian nonparametric clustering?

We answer this question by introducing an extension of the BNTL model called the Neutral-to-the-Left (NTL) mixture model, a family of infinite mixture models that are parameterized by the distribution of stick breaking weights and arrival time distribution of clusters, and serves as a generalization of the Dirichlet process mixture model (DPMM). We consider one such parameterization which exhibits clustering behaviour distinct from that of the popular DPMM.

3.1 Generative model

We define a Neutral-to-the-Left (NTL) mixture model to be given by the following generative process:

$$
\begin{aligned}
T_j &\sim \Lambda, \\
\psi_j | T_j &\overset{\text{iid}}{\sim} \text{Beta}(a(j,T_j), b(j,T_j)) \text{ for } j \geq 2 \\
P_{j,K_n} &= \psi_j \prod_{\ell=j+1}^{K_n} (1-\psi_\ell) \text{ (where } \psi_1 = 1) \\
Z_n &\sim \begin{cases} \delta_{K_n}(\cdot) & \text{for } n = T_{K_n} \\ \text{Categorical}(P_{j,K_n}) & \text{otherwise} \end{cases} \\
\theta_j &\sim F(\cdot) \\
X_n | Z_n &\sim f(\cdot | \theta_{Z_n}).
\end{aligned}
\tag{3.1}
$$

The generative process described in 3.1 differs slightly from the model described in model 2.3 in that we assume each observation i is associated with datum X_i, and that the parameters of the Beta distribution of the ψ_j weights are also explicit parameters of the model, along with the arrival time distribution Λ. The parameters of the Beta distribution in 3.1 are functions of both the index j of the current cluster and the arrival time of the cluster T_j. For example, setting $a(j,T_j) := 1-\alpha$, $b(j,T_j) = T_j - 1 - (j-1)\alpha$ recovers the stick breaking distribution of model 2.3.

3.2 Parameterizations of the Neutral-to-the-Left Mixture Model

As mentioned previously, the NTL mixture model has the following set of parameters which can be modified by the user as they see fit: $a(j,T_j)$ and $b(j,T_j)$, the parameters of the distribution of the stick breaking weights, and Λ, the distribution of arrival times of the clusters in the model.

We wish to emphasize that the choice of values of these parameters can lead to clusterings with vastly different characteristics. For example, it is shown by Bloem-Reddy et al. [2] that choosing $a(j,T_j) = 1$, $b(j,T_j) = T_j - 1$, and Λ equal to

the arrival time distribution of the Dirichlet process leads to clusterings which are exchangeable and whose largest cluster size grow linearly in the number of observations. On the other hand, in the next subsection, we consider a parameterization with characteristics that differ from that of the DPMM.

3.2.1 Conjectured microclustering property of a parameterization of the NTL mixture model

In this subsection, we introduce a parameterization of the NTL mixture model which is not exchangeable and yields clusters whose expected sizes are bounded, and thus do not grow linearly. Because of this, we hypothesize that the following parameterization exhibits the *microclustering* property, where $M_n/n \xrightarrow{p} 0$ in which M_n is the size of the largest cluster in the model [4]. We leave the proof of this conjecture for future work. The following proposition suggests that the conjecture may be true.

Proposition 1. *Define the random variable S_{j,K_n} to be the number of observations assigned to cluster j when there are K_n total clusters in a realization of the NTL mixture model. Consider constant $a(j,T_j) \equiv a$, $b(j,T_j) \equiv b$ where $a,b > 0$, and $\Delta_j \overset{iid}{\sim} \Lambda$ for $j \geq 2$ where Λ is some probability distribution over the positive integers with finite first moment. Then*

$$\lim_{K_n \to \infty} \mathbb{E}[S_{j,K_n}] = \mathbb{E}[\Delta_2].$$

Proof. We have that

$$\begin{aligned}
\mathbb{E}[S_{j,K_n}] &= 1 + \sum_{i=j}^{K_n} \mathbb{E}\left[(\Delta_{i+1} - 1)\psi_j \prod_{\ell=j+1}^{i} (1-\psi_\ell)\right] \\
&= 1 + \sum_{i=j}^{K_n} (\mathbb{E}[\Delta_2] - 1)\left(\frac{a}{a+b}\right)\left(\frac{b}{a+b}\right)^{i-j} \quad \text{(by independence of } \psi_k, \Delta_j\text{)} \\
&= 1 + (\mathbb{E}[\Delta_2] - 1)\left(\frac{a}{a+b}\right)\sum_{i=j}^{K_n}\left(\frac{b}{a+b}\right)^{i-j}.
\end{aligned}$$

Notice that the expected number of observations assigned to cluster j has a term

which is a finite geometric series of $\frac{b}{a+b}$. Taking the limit as $K_n \to \infty$ gives

$$\begin{aligned}
\lim_{K_n \to \infty} \mathbb{E}[S_{j,K_n}] &= 1 + (\mathbb{E}[\Delta_2] - 1) \left(\frac{a}{a+b} \right) \lim_{K_n \to \infty} \sum_{i=j}^{K_n} \left(\frac{b}{a+b} \right)^{i-j} \\
&= 1 + (\mathbb{E}[\Delta_2] - 1) \left(\frac{a}{a+b} \right) \left(\frac{1}{1 - \frac{b}{a+b}} \right) \\
&= 1 + (\mathbb{E}[\Delta_2] - 1) \left(\frac{a}{a+b} \right) \left(\frac{a+b}{a} \right) \\
&= \mathbb{E}[\Delta_2].
\end{aligned}$$

Therefore, we see that the expected number of observations assigned to any cluster is finite as the number of clusters, and in turn, the number of observations, goes to infinity. □

We can represent a clustering model as (a collection of) distributions over random partitions Π_n over $[n] := \{1, 2, \ldots, n\}$. In particular, it has been shown that microclustering models cannot exhibit infinite exchangeability in the observations [4]. Therefore, microclustering models must sacrifice at least one of two properties:

1. *finite exchangeability* – permutation invariance over $[n]$, or
2. *projectivity* – equality of Π_n in distribution to Π_m restricted to the first n elements for $1 \leq n < m$.

Therefore, if this parameterization of the NTL mixture model does satisfy the microclustering property, the NTL mixture model must sacrifice *finite exchangeability*, since it is easy to see that $P(\mathbf{Z})$ is not permutation invariant in the order of $\mathbf{Z}$.

3.3 Inference

In this section, we consider the problem of sampling the posterior distribution of the cluster assignments $\mathbf{Z}$ given observations $\mathbf{X}$ for a particular parameterization of the NTL mixture model.

We first describe the structure of the joint distribution of the NTL mixture model in the general setting, which follows from a similar derivation for the BNTL model in Bloem-Reddy et al. [2].

Let $\mathbb{T}_{K_n} = (T_1, T_2, \ldots, T_{K_n})$, $\Psi_{K_n} = (\psi_1, \psi_2, \ldots, \psi_{K_n})$ (with the constraint that $\psi_1 = 1$), $\mathbf{Z}_n = (Z_1, Z_2, \ldots, Z_n)$, and $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$, and $\Theta_{K_n} = (\theta_1, \theta_2, \ldots, \theta_{K_n})$. We will assume that $f(x|\theta)$ is some likelihood, with conjugate prior $F(\theta)$. Let Λ_ϕ be the arrival time distribution with parameter ϕ, with prior distribution $G(\phi)$ on ϕ. We also assume that the Λ_ϕ is a Markov chain over the interarrivals, with each interarrival possibly depending only on the arrival time immediately preceding it, so that

$$\Lambda_\phi(\mathbb{T}_{K_n}) = \delta_1(T_1) \prod_{s=2}^{K_n} p_s^\phi(\Delta_s | T_{s-1}). \tag{3.2}$$

Here, $p_s^\phi(\Delta_s|T_{s-1})$ is the interarrival time distribution for the sth cluster, with ϕ the parameter of the distributions, and $G(\phi)$ is possibly conjugate to p_s^ϕ. This structure of the arrival time distribution has the capacity to describe a variety of arrival time behaviours. For example, the arrival time distribution of the DP can expressed in the form of equation 3.2 [23]. We may also wish to consider the more simpler situation of *iid* interarrivals, so that $p_s^\phi(\Delta_s|T_{s-1}) = p_\phi(\Delta_s)$, which we consider later on in this section.

Notice that knowledge of $\mathbf{Z}_n$ gives complete information for $\mathbb{T}_{K_n}$, since we have that

$$T_s = \min\{i : Z_i = s\} \tag{3.3}$$

under generative model 3.1. Therefore, given $\mathbf{Z}_n$, $\mathbf{X}_n$, Θ_{K_n}, Ψ_{K_n}, and ϕ, the general form of the joint likelihood of an NTL mixture model is given by

$$\begin{aligned} P(\mathbf{X}_n, \mathbf{Z}_n, \Theta_{K_n}, \Psi_{K_n}, \phi) = &\left(\prod_{s=1}^{K_n} \left[\prod_{i:z_i=s} f(X_i|\theta_s) \right] F(\theta_s) \right) \\ &\times \left[\prod_{s=2}^{K_n} \left(\frac{\psi_s^{n(s)-1+a(s,T_s)-1} (1-\psi)^{n(<s)+b(s,T_s)-1}}{B(a(s,T_s), b(s,T_s))} \right) p_s^\phi(\Delta_s|T_{s-1}) \right] G(\phi) \end{aligned}$$

where

- K_n is the number of clusters,

- $n(s) = \#\{i : z_i = s\}$, and
- $n(< T_s) \equiv n(< s) = \left(\sum_{s':T_{s'}<T_s} n(s')\right) - (T_s - 1)$.

We consider the particular parameterization of *iid* interarrival times distributed to a geometric distribution, so that $p_j^{\phi}(\cdot|T_{j-1}) = p_\phi(\cdot) = \text{Geom}(\cdot|\phi)$, and with conjugate prior $G(\cdot) = \text{Beta}(\cdot|a_\phi, b_\phi)$ on ϕ. Here, we use the convention of the geometric distribution having support only on the positive integers. We also set constant parameters $a(s,T_s) \equiv a > 0$, $b(s,T_s) \equiv b > 0$. This leads to the following expression for the joint likelihood

$$P(\mathbf{X}_n, \mathbf{Z}_n, \Theta_{K_n}, \Psi_{K_n}, \phi) = \left(\prod_{s=1}^{K_n}\left[\prod_{i:z_i=s} f(X_i|\theta_s)\right] F(\theta_s)\right)\left[\prod_{s=2}^{K_n}\left(\frac{\psi_s^{n(s)-1+a-1}(1-\psi)^{n(<s)+b-1}}{B(a,b)}\right)\right] \times \left[\frac{\phi^{K_n-1+a_\phi-1}(1-\phi)^{n-K_n+b_\phi-1}}{B(a_\phi,b_\phi)}\right].$$

Integrating out Ψ_{K_n} and ϕ in the joint likelihood leads to

$$P(\mathbf{X}_n, \mathbf{Z}_n, \Theta_{K_n}) = \left(\prod_{s=1}^{K_n}\left[\prod_{i:z_i=s} f(X_i|\theta_s)\right] F(\theta_s)\right)\left[\prod_{s=2}^{K_n}\left(\frac{B(n(s)-1+a_s, n(<s)+b_s)}{B(a_s,b_s)}\right)\right] \times \left[\frac{B(K_n-1+a_\phi, n-K_n+b_\phi)}{B(a_\phi,b_\phi)}\right].$$

3.3.1 A Collapsed Gibbs Sampling Algorithm

We now derive a collapsed Gibbs sampler to estimate the posterior distribution $P(\mathbf{Z}_n|\mathbf{X}_n)$ for this parameterization of the NTL mixture model. Using the abuse of notation $\mathbf{Z} := \mathbf{Z}_n$ and $\mathbf{X} := \mathbf{X}_n$, we would like to sample

$$P(Z_i = s|\mathbf{Z}_{-i}, \mathbf{X}) \;\propto\; P(X_i|Z_i = s, \mathbf{Z}_{-i}, \mathbf{X}_{-i})P(Z_i = s|\mathbf{Z}_{-i}),$$

where $\mathbf{Z}_{-i}$ and $\mathbf{X}_{-i}$ are the tuples of cluster assignments and observations from $\mathbf{Z}$ and $\mathbf{X}$ except for the ith components. The term $P(X_i|Z_i = s, \mathbf{Z}_{-i}, \mathbf{X}_{-i})$ is a posterior predictive distribution, which has a known form when $F(\theta)$ is a conjugate prior to $f(X|\theta)$. We describe various choices of f and F in a later subsection.

The more complicated term we deal with now is

$$P(Z_i = s|\mathbf{Z}_{-i}) = \int\int P(Z_i = s|\mathbf{Z}_{-i}, \Psi_{K_n}, \phi) P(\Psi_{K_n}|\mathbf{Z}_{-i}) P(\phi|\mathbf{Z}_{-i}) d\Psi_{K_n} d\phi.$$

In what follows, let $n_{-i}(s)$ be the number of observations assigned to cluster s when observation i is removed from the model, $n_{-i}(< s)$ be the number of observations assigned to the complement of cluster s when i is removed from the model, and K_n^{-i} is the number of clusters in the model when i is removed. Moreover, define $n(< i)$ for $i = 1, \ldots, n$ by

$$n(< i) = \left(\sum_{s: T_s < i} n(s) \right) - (i-1).$$

As before, let $n_{-j}(< i)$ be the above quantity but under the situation where observation j is removed. Define

$$C(\mathbf{Z}_{-i}) = \left[\prod_{s': s' > 1} \frac{B(n(s') - 1 + a, n(< s') + b)}{B(a,b)} \right] \left[\frac{B(K_n^{-i} - 1 + a_\phi, n - K_n^{-i} - 1 + b_\phi)}{B(a_\phi, b_\phi)} \right]$$

which is a factor which all cases of the conditional probability $P(Z_i = s|\mathbf{Z}_{-i})$ share.

We enumerate three cases for the update term $P(Z_i = s|\mathbf{Z}_{-i}) \propto P(Z_i = s, \mathbf{Z}_{-i})$.

1. **Case 1: Z_i is assigned to a new cluster.**

   If $i > 1$, we have

$$\begin{aligned} & P(Z_i = s, \mathbf{Z}_{-i}, \Psi_{K_n}, \phi) \\ &= \left(\frac{\psi_s^{a-1}(1-\psi_s)^{n_{-i}(<s)+b-1}}{B(a,b)} \right) \left(\prod_{s': s' > 1} \frac{\psi_{s'}^{n_{-i}(s')-1+a-1}(1-\psi_{s'})^{n_{-i}(<s')+b-1}}{B(a,b)} \right) \\ &\quad \times \left(\frac{\phi^{(K_n^{-i}+1)-1+a_\phi-1}(1-\phi)^{n-K_n^{-i}-1+b_\phi-1}}{B(a_\phi, b_\phi)} \right) \end{aligned}$$

Integrating out ϕ and Ψ_{K_n} gives

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}) \\
=&\left(\frac{B(a, n_{-i}(<s)+b)}{B(a,b)}\right)\left(\prod_{s':s'>1}\frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)}\right) \\
&\times\left(\frac{B((K_n^{-i}+1)-1+a_\phi, n-K_n^{-i}-1+b_\phi)}{B(a_\phi,b_\phi)}\right) \\
=&\left(\frac{B(a, n_{-i}(<s)+b)}{B(a,b)}\right)\left(\prod_{s':s'>1}\frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)}\right) \\
&\times\left(\frac{B(K_n^{-i}-1+a_\phi, n-K_n^{-i}-1+b_\phi)}{B(a_\phi,b_\phi)}\right)\left(\frac{K_n^{-i}-1+a_\phi}{n-2+a_\phi+b_\phi}\right) \\
=&\left(\frac{B(a, n_{-i}(<s)+b)}{B(a,b)}\right)\left(\frac{K_n^{-i}-1+a_\phi}{n-2+a_\phi+b_\phi}\right) \\
&\times\left(\prod_{s':s'>1}\frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)}\right)\left(\frac{B(K_n^{-i}-1+a_\phi, n-K_n^{-i}-1+b_\phi)}{B(a_\phi,b_\phi)}\right) \\
=&C(\mathbf{Z}_{-i})\left(\frac{B(a, n_{-i}(<s)+b)}{B(a,b)}\right)\left(\frac{K_n^{-i}-1+a_\phi}{n-2+a_\phi+b_\phi}\right).
\end{aligned}
$$

Otherwise, if $i = 1$, we have

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}, \Psi_{K_n}, \phi) \\
=&\left(\prod_{s':s'\neq s}\frac{\psi_{s'}^{n_{-i}(s')+a-1}(1-\psi_{s'})^{n_{-i}(<s')+b-1}}{B(a,b)}\right)\left(\frac{\phi^{(K_n^{-i}+1)-1+a_\phi-1}(1-\phi)^{n-K_n^{-i}-1+b_\phi-1}}{B(a_\phi,b_\phi)}\right).
\end{aligned}
$$

With similar reasoning as in the subcase $i > 1$, integrating out Ψ and ϕ gives

$$
P(Z_i = s, \mathbf{Z}_{-i}) = C(\mathbf{Z}_{-i})\left(\frac{K_n^{-i}-1+a_\phi}{n-2+a_\phi+b_\phi}\right).
$$

2. **Case 2: Z_i is assigned to an existing cluster s with $T_s < i$.**

We have that

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}, \Psi_{K_n}, \phi) \\
&= \left(\frac{\psi_s^{n_{-i}(s)+a-1}(1-\psi_s)^{n_{-i}(<s)+b-1}}{B(a,b)} \right) \left[\prod_{\substack{s': \\ T_s<T_{s'}<i}} \frac{\psi_{s'}^{n(s')-1+a-1}(1-\psi_{s'})^{n(<s')+1+b-1}}{B(a,b)} \right] \\
&\quad \times \left[\prod_{\substack{s': \\ T_{s'}<T_s \text{ or } T_{s'}>i}} \frac{\psi_{s'}^{n(s')-1+a-1}(1-\psi_{s'})^{n(<s')+b-1}}{B(a,b)} \right] \left(\frac{\phi^{K_n^{-1}-1+a_\phi-1}(1-\phi)^{n-K_n^{-i}+b_\phi-1}}{B(a_\phi,b_\phi)} \right).
\end{aligned}
$$

Integrating out Ψ_{K_n} and ϕ gives

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}) \\
&= \left[\frac{B(n_{-i}(s)+a, n_{-i}(<s)+b)}{B(a,b)} \right] \left[\prod_{\substack{s': \\ T_s<T_{s'}<i}} \frac{B(n(s')-1+a, n(<s')+1+b)}{B(a,b)} \right] \\
&\quad \times \left[\prod_{\substack{s': \\ T_{s'}<T_s \text{ or } T_{s'}>i}} \frac{B(n(s')-1+a, n(<s')+b)}{B(a,b)} \right] \left[\frac{B(K_n^{-1}-1+a_\phi, n-K_n^{-i}+b_\phi)}{B(a_\phi,b_\phi)} \right] \\
&= \left(\prod_{s':s'>1} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)} \right) \left(\frac{B(K_n^{-i}-1+a_\phi, n-K_n^{-i}-1+b_\phi)}{B(a_\phi,b_\phi)} \right) \\
&\quad \times \left(\frac{n_{-i}(s)-1+a_s}{n_{-i}(s)+n_{-i}(<s)-1+a_s+b_s} \right) \left[\prod_{s':T_s<T_{s'}<i} \frac{n_{-i}(<s')+b_{s'}}{n_{-i}(s')+n_{-i}(<s')-1+a_{s'}+b_{s'}} \right] \\
&\quad \times \left(\frac{n-K_n^{-i}-1+b_\phi}{n-2+a_\phi+b_\phi} \right) \\
&= C(\mathbf{Z}_{-i}) \left(\frac{n_{-i}(s)-1+a_s}{n_{-i}(s)+n_{-i}(<s)-1+a_s+b_s} \right) \left[\prod_{s':T_s<T_{s'}<i} \frac{n_{-i}(<s')+b_{s'}}{n_{-i}(s')+n_{-i}(<s')-1+a_{s'}+b_{s'}} \right] \\
&\quad \times \left(\frac{n-K_n^{-i}-1+b_\phi}{n-2+a_\phi+b_\phi} \right)
\end{aligned}
$$

3. **Case 3: Z_i is assigned to an existing cluster s with $i < T_s$.**

If $i > 1$, then we have

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}, \Psi_{K_n}, \phi) \\
=&\left(\frac{\psi_s^{n(s)+a-1}, (1-\psi_s)^{n_{-i}(<i)+b-1}}{B(a,b)} \right) \left[\prod_{s':i<T_{s'}<T_s} \frac{\psi_{s'}^{n_{-i}(s')-1+a-1}(1-\psi_{s'})^{n_{-i}(<s')+n_{-i}(s)+b-1}}{B(a,b)} \right] \\
&\times \left[\prod_{s':T_{s'}<i \text{ or} T_{s'}>T_s} \frac{\psi_{s'}^{n_{-i}(s')-1+a-1}(1-\psi_{s'})^{n_{-i}(<s')+b-1}}{B(a,b)} \right] \left(\frac{\phi^{K_n^{-i}-1+a_\phi-1}(1-\phi)^{n-K_n^{-i}+b_\phi-1}}{B(a_\phi,b_\phi)} \right).
\end{aligned}
$$

Integrating out Ψ_{K_n} and ϕ gives

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}) \\
=&\left(\frac{B(n(s)+a, n_{-i}(<i)+b)}{B(a,b)} \right) \left[\prod_{s':T_{s'}<i \text{ or } T_{s'}>T_s} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)} \right] \\
&\times \left[\prod_{s':i<T_{s'}<T_s} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+n_{-i}(s)+b)}{B(a,b)} \right] \left(\frac{B(K_n^{-i}-1+a_\phi, n-K_n^{-i}+b_\phi)}{B(a_\phi,b_\phi)} \right) \\
=&\left[\frac{n-1-K_n+b_\phi}{n-2+a_\phi+b_\phi} \right] \left[\frac{B(n_i(s)+a, n_{-i}(<i)+b)}{B(n_{-i}(s)-1+a_s, n_{-i}(<s)+b_s)} \right] \\
&\times \left(\prod_{s':i<T_{s'}<T_s} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+n_{-i}(s)+b)}{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)} \right) \\
&\times \left(\prod_{s':s'>1} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}{B(a,b)} \right) \left(\frac{B(K_n^{-i}-1+a_\phi, n-K_n^{-i}-1+b_\phi)}{B(a_\phi,b_\phi)} \right) \\
=&C(\mathbf{Z}_{-i}) \left[\frac{n-1-K_n+b_\phi}{n-2+a_\phi+b_\phi} \right] \left[\frac{B(n_{-i}(s)+a, n_{-i}(<i)+b)}{B(n_{-i}(s)-1+a_s, n_{-i}(<s)+b_s)} \right] \\
&\times \prod_{s':i<T_{s'}<T_s} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+n_{-i}(s)+b)}{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)}.
\end{aligned}
$$

Otherwise, suppose $i = 1$.

We have that

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}, \Psi_{K_n}, \phi) \\
&= \left[\prod_{s': i < T_{s'} < T_s} \frac{\psi_{s'}^{n_{-i}(s')-1+a-1} (1-\psi_{s'})^{n_{-i}(<s')+n_{-i}(s)+b-1}}{B(a,b)} \right] \\
&\quad \times \left[\prod_{s': T_{s'} < i \text{ or} T_{s'} > T_s} \frac{\psi_{s'}^{n_{-i}(s')-1+a-1} (1-\psi_{s'})^{n_{-i}(<s')+b-1}}{B(a,b)} \right] \\
&\quad \times \left(\frac{\phi^{K_n^{-i}-1+a_\phi-1} (1-\phi)^{n-K_n^{-i}+b_\phi-1}}{B(a_\phi, b_\phi)} \right).
\end{aligned}
$$

By similar reasoning as in subcase $i > 1$, integrating out Ψ_{K_n} and ϕ gives

$$
\begin{aligned}
&P(Z_i = s, \mathbf{Z}_{-i}) \\
&= C(\mathbf{Z}_{-i}) \left[\frac{n-1-K_n+b_\phi}{n-2+a_\phi+b_\phi} \right] \left[\frac{1}{B(n_{-i}(s)-1+a_s, n_{-i}(<s)+b_s)} \right] \\
&\quad \times \left[\prod_{s': i < T_{s'} < T_s} \frac{B(n_{-i}(s')-1+a, n_{-i}(<s')+n_{-i}(s)+b)}{B(n_{-i}(s')-1+a, n_{-i}(<s')+b)} \right]
\end{aligned}
$$

This provides a complete collapsed Gibbs sampling algorithm for inferring the posterior $P(\mathbf{Z}|\mathbf{X})$.

All three cases (and subcases therein) of the term $P(Z_i = s, \mathbf{Z}_{-i})$ share a constant term $C(\mathbf{Z}_{-i})$. Therefore, when computing the Gibbs proposal $P(Z_i = s|\mathbf{Z}_{-i}) \propto P(Z_i = s, \mathbf{Z}_{-i})$, we need only to compute all terms other than the $C(\mathbf{Z}_{-i})$ which reduces the time complexity of the collapsed Gibbs sampler. Notice that in case 2 of the posterior sampling of Z_i in the collapsed Gibbs sampler, calculating $P(Z_i = s|\mathbf{Z}_{-i})$ requires no more than K_n multiplications (since we have a factor for each state s' so that $T_s < T_{s'} < i$). Therefore, a single sweep of the Gibbs sampling algorithm has time complexity on the order of $O(nK_n^2)$.

3.3.2 A Metropolis–Hastings Algorithm

Although the collapsed Gibbs sampling algorithm described in the previous section has the advantage of having each move of the algorithm being always accepted [24], the algorithm suffers from large time complexity.

The collapsed Gibbs sampler's time complexity of $O(nK_n^2)$ per sweep pales in comparison to the time complexity of a single sweep of a collapsed Gibbs sampling algorithm for the DPMM, which has time complexity $O(nK_n)$ owing to the fact that we may write

$$P(Z_i = s|\mathbf{Z}_{-i}) \propto \begin{cases} n_{-i}(s) & \text{if } s \text{ is an existing cluster} \\ \alpha & \text{if } s \text{ if a new cluster} \end{cases}$$

under the DP model.

We design a Metropolis–Hastings algorithm with an efficient proposal that addresses the poor time complexity of the collapsed Gibbs sampler that is motivated by the following observation. Figure 3.1 depicts an example of 100 observations generated from the NTL mixture model prior.

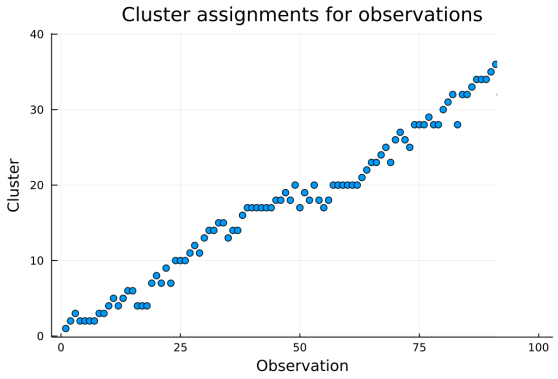


Figure 3.1: Cluster assignments of 100 observations generated from the NTL mixture model prior.

Notice that clusters tend to contain observations from around the same time period, with observations farther away in time very rarely being clustered together. Then if we were to design a proposal distribution for an accept-reject Metropolis–Hastings algorithm, one reasonable proposal distribution would be a discrete uniform distribution over clusters centered around the previously assigned cluster.

Formally, we use the following proposal distribution q in the Metropolis–Hastings algorithm to be described shortly. Let S_{-i} be an ordered sequence of clusters such that the following hold.

1. If $T_{Z_i} < i$ or i is the only observation assigned to cluster Z_i, we have that S_{-i} contains

   (a) clusters s with birth time $T_s < i$ (older than observation i) from the model when observation i is removed are included, and

   (b) a new cluster s' corresponding to $T_{s'} = i$.

2. If $T_{Z_i} = i$ and there is more than one observation at cluster Z_i, then $S_{-i} = \{Z_i\}$.

Assume that the clusters s in S_{-i} are ordered according to their birth times T_s (where in the situation that Z_i is assigned a new cluster, then the new cluster s' corresponding to $Z_i = s'$ has birth time $T_{s'} = i$). Using indices that start at 1, let $S_{-i}[d,e]$ to be the subsequence of S_{-i} that starts from the dth element and ends at the eth element, both inclusive. Let $\text{index}(Z_i)$ be the index of the cluster Z_i in S_{-i}. Then define the proposal q to be

$$q(Z_i' = s|Z_i) \propto \begin{cases} 1 & \text{if } s \in S_{-i}[\max\{1, \text{index}(Z_i) - w\}, \min\{\#S_{-i}, \text{index}(Z_i) + w\}] \\ 0 & \text{otherwise} \end{cases}$$

with w being some user-specified positive integer. Intuitively, q is a proposal distribution which is uniform around a window of width w centered at the previous cluster Z_i, where observations which are younger than their assigned clusters, or are the only observation assigned to their clusters, are allowed to be moved, and they can only move to other younger clusters, or a new cluster at i.

The acceptance ratio α can then be expressed as

$$\alpha = \min\left(1, \frac{P(X_i|Z_i', \mathbf{Z}_{-i}, \mathbf{X}_{-i}) P(Z_i'|\mathbf{Z}_{-i}) q(Z_i|Z_i')}{P(X_i|Z_i, \mathbf{Z}_{-i}, \mathbf{X}_{-i}) P(Z_i|\mathbf{Z}_{-i}) q(Z_i'|Z_i)}\right). \tag{3.4}$$

Each of $P(Z_i'|\mathbf{Z}_{-i})$ and $P(Z_i|\mathbf{Z}_{-i})$ require only $O(K_n)$ computations to calculate and $q(Z_i'|Z_i)$ can be calculated in constant time, so a single sweep of the Metropolis–Hastings algorithm has quadratic time complexity $O(nK_n)$, an improvement over the $O(nK_n^2)$ time complexity of the collapsed Gibbs sampling algorithm described in the previous section.

Irreducibility of the Metropolis–Hastings sampler

We demonstrate that any two clusterings $\mathbf{Z}$ and $\mathbf{Z}'$ are accessible using the Metropolis–Hastings sampler described earlier – that is, for any two clusterings $\mathbf{Z}$ and $\mathbf{Z}'$, if the chain is at a state $\mathbf{Z}$, then it has positive probability of eventually arriving to a state $\mathbf{Z}'$. This will imply irreducibility of the Markov chain induced by the Metropolis–Hastings sampler [25].

For the purpose of the proof, we will show the following two results.

1. Given arbitrary clusters s, s', $q(s|s') > 0$ if and only if $q(s'|s) > 0$.

2. Let $\mathbf{Z}^0$ be the trivial clustering where each observation is assigned to its own cluster, and $\mathbf{Z}$ an arbitrary clustering. There exists a procedure to transform configuration $\mathbf{Z}^0$ into configuration $\mathbf{Z}$.

The above two results will then imply that there exists a procedure to transform $\mathbf{Z}$ into $\mathbf{Z}^0$, by applying result 1 as many times as needed to the procedure described in result 2. Then, we may transform configuration $\mathbf{Z}$ into $\mathbf{Z}'$ by first transforming $\mathbf{Z}$ into $\mathbf{Z}^0$, and then transforming $\mathbf{Z}^0$ into $\mathbf{Z}'$, which will imply that $\mathbf{Z}'$ is accessible from $\mathbf{Z}$.

Proof of result 1. If $q(s'|s) > 0$, then s and s' are either both older than observation i, or one of them is a cluster with the single element i, since both $s, s' \in S_{-i}$. As well, since $s, s' \in S_{-i}[\max\{1, \text{index}(s) - w\}, \min\{\#S_{-i}, \text{index}(s) + w\}]$, then s' is no more than w indices away from s in S_{-i}. Therefore, $s \in S_{-i}[\max\{1, \text{index}(s') - w\}, \min\{\#S_{-i}, \text{index}(s') + w\}]$. This implies that $q(s|s') > 0$. The converse holds similarly. □

Proof of result 2. We describe the following procedure to transform clustering $\mathbf{Z}^0 = (Z_1^0, Z_2^0, \ldots, Z_n^0) = (1, 2, \ldots, n)$ into the clustering $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_n)$.

First, some preliminaries. We say that a cluster Z_i exists in $\mathbf{Z}^0$ if there exists j so that $T_{Z_i} = T_{Z_j^0}$ (i.e., there exists a cluster Z_j^0 in $\mathbf{Z}^0$ that has the same birth time as cluster Z_i), where T_{Z_i} is defined as in equation 3.3. Further, it is clear that if $(T_{Z_1}, T_{Z_2}, \ldots, T_{Z_n}) = (T_{Z_1^0}, T_{Z_2^0}, \ldots, T_{Z_n^0})$, then $\mathbf{Z}$ and $\mathbf{Z}^0$ describe the same clustering. Lastly, notice that before the start of the procedure to be described, $i = T_{Z_i^0}$ for every i.

For $i = 1, \ldots, n$, perform the following.

1. If $i = T_{Z_i}$ (i.e., observation i is the first observation assigned to cluster Z_i in $\mathbf{Z}$), then leave Z_i^0 unchanged. This implies that $T_{Z_i} = i = T_{Z_i^0}$, so cluster Z_i exists in $\mathbf{Z}^0$.

2. Otherwise, we have $i > T_{Z_i}$ (observation i is younger than cluster Z_i). The cluster Z_i is guaranteed to exist in $\mathbf{Z}^0$ by step 1 of this procedure. Then, repeatedly assign Z_i^0 to the youngest cluster s that is older than observation i (this is allowed since s is exactly one index away from Z_i^0 in S_{-i} at any point in this procedure), possibly keeping all other Z_j^0 fixed for multiple sweeps of the Metropolis-Hastings sampler, until $T_{Z_i^0} = T_{Z_i}$.

At the end of this procedure, we have $(T_{Z_1^0}, T_{Z_2^0}, \ldots, T_{Z_n^0}) = (T_{Z_1}, T_{Z_2}, \ldots, T_{Z_n})$, which is what we wanted to show. □

3.3.3 Choice of likelihood $f(\cdot|\theta)$ and prior $F(\theta)$

Depending on the nature of the observations at hand, there are a number of different parameterizations of the likelihood f and prior F. As mentioned previously, it is convenient to choose F to be conjugate to all parameters of f, so that in the collapsed Gibbs sampling algorithm described in the earlier section, the parameters Θ may be integrated out leaving only the posterior distribution of the cluster assignments $\mathbf{Z}$ to be sampled. However, this may not be possible for more complicated likelihoods f, so posterior sampling of at least some components of θ may be necessary.

We detail two examples of parameterizations of $f(\cdot|\theta)$ and $F(\theta)$ that we explore in detail later in the experiments section of this thesis.

For the first parameterization (whose details are based on [17]), suppose that the observations $(X_i)_{i=1}^n$ take on k-dimensional, continuous, and unbounded values. Then, a natural choice is to set f to be a multivariate Gaussian distribution with parameters $\theta_{Z_i} = (\mu_{Z_i}, \Sigma_{Z_i})$ so that

$$f(X_i|\mu_{Z_i}, \Sigma_{Z_i}) = \mathcal{N}(X_i|\mu_{Z_i}, \Sigma_{Z_i}) := \frac{\exp\left(-\frac{1}{2}(X_i - \mu_{Z_i})^T \Sigma_{Z_i}^{-1}(X_i - \mu_{Z_i})\right)}{\sqrt{(2\pi)^k |\Sigma_{Z_i}|}}$$

where $|\Sigma_{Z_i}|$ is the determinant of the square matrix Σ_{Z_i}. If $\Sigma_{Z_i} = \Sigma$ is known and

constant across all clusters, we may set the prior F of μ to be another multivariate Gaussian with mean μ_0 and covariance Σ_0. Under this parameterization, the posterior predictive $P(X_i|\mathbf{X}_{-i},\mathbf{Z})$ has the following convenient closed-form expression

$$\begin{aligned} P(X_i|\mathbf{X}_{-i},\mathbf{Z}) &= \int P(X_i|\mu_{Z_i},\Sigma)P(\mu_{Z_i}|\mu_0,\Sigma_0,\mathbf{X}_{-i},\mathbf{Z})d\mu_{Z_i} \\ &= \mathcal{N}(X_i|\mu'_{Z_i}(\mathbf{X}_{-i},\mathbf{Z}),\Sigma'_{Z_i}(\mathbf{X}_{-i},\mathbf{Z})+\Sigma) \end{aligned}$$

where $\mu'_{Z_i}(\mathbf{X}_{-i},\mathbf{Z})$ and $\Sigma'_{Z_i}(\mathbf{X}_{-i},\mathbf{Z})$ are the parameters of the posterior distribution $P(\mu_{Z_i}|\mu_0,\Sigma_0,\mathbf{X}_{-i},\mathbf{Z})$, which is another multivariate Gaussian distribution.

The second parameterization we consider is based on details from [26]. We may model $(X_i)_{i=1}^N$ as k-dimensional count data, so that $X_i = (X_i^1,\dots X_i^k)$ for $X_i^\ell \in \mathbb{N}$, with $m_i = \sum_{\ell=1}^k X_i^\ell$ total counts for observation i. In this case, we set $f(X_i|\theta_{Z_i})$ to be the probability mass function of the multinomial distribution

$$f(X_i|\theta_{Z_i}) = \text{Multinomial}(X_i|m_i,p_{Z_i}) = \frac{m_i!}{X_i^1!\dots X_i^k!}\left(p_{Z_i}^1\right)^{X_i^1}\dots\left(p_{Z_i}^k\right)^{X_i^k}$$

with parameters $\theta_s = p_s = (p_s^1,p_s^2,\dots,p_s^k)$ so that $p_s^1,\dots,p_s^k \geq 0$ and $\sum_{\ell=1}^k p_s^\ell = 1$, and $m_i = \sum_{\ell=1}^k X_i^\ell$ is the number of trials. The natural choice for the prior $F(p_s)$ is the probability density function of the Dirichlet distribution, which is given by

$$F(p_s) = \frac{1}{\mathbf{B}(\alpha)}\prod_{\ell=1}^k (p_s^\ell)^{\alpha^i-1}$$

where $\alpha = (\alpha^1,\dots,\alpha^k)$ for $\alpha^1,\dots,\alpha^k > 0$, and $\mathbf{B}(\alpha)$ is the multivariate Beta function. This also induces the posterior predictive $P(X_i|\mathbf{X}_{-i},\mathbf{Z})$ to be the probability mass function of the Dirichlet-multinomial distribution, which has analytic expression

$$\begin{aligned} P(X_i|\mathbf{X}_{-i},\mathbf{Z}) &= \text{DirMult}(X_i|\alpha_{Z_i}(\mathbf{X}_{-i},\mathbf{Z})) \\ &= \frac{\Gamma(\alpha^0_{Z_i}(\mathbf{X}_{-i},\mathbf{Z}))\Gamma(m_i+1)}{\Gamma(\alpha^0_{Z_i}(\mathbf{X}_{-i},\mathbf{Z})+m_i)}\prod_{\ell=1}^k \frac{\Gamma(X_i^\ell+\alpha^\ell_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}))}{\Gamma(\alpha^\ell_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}))\Gamma(X_i^\ell+1)} \end{aligned}$$

where $\alpha_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}) = (\alpha^1_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}),\dots,\alpha^k_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}))$ are the parameters of the posterior distribution $P(p_{Z_i}|\mathbf{X}_{-i},\mathbf{Z})$, which is a Dirichlet distribution, and $\alpha^0_{Z_i}(\mathbf{Z}_{-i},\mathbf{X}) =$

$\sum_{\ell=1}^{k} \alpha_{Z_i}^{\ell}(\mathbf{Z}_{-i}, \mathbf{X})$.

Chapter 4

Experiments

In this section, we study the NTL mixture model and its Metropolis-within-Gibbs samplers on both synthetic and real data. Experiments are done mainly using the programming language Julia [27], with the exception of the computation of two Bayes estimators, which are done using the programming language **R** [28]. Experiments were performed using a MacBook Pro (13-inch, 2020, Two Thunderbolt 3 ports) with a 1.4 GHz Quad-Core Intel Core i5 processor, and 8 GB of 2133 MHz LPDDR3 RAM. Code for the experiments can be found at the following link: https://github.com/realseanla/ntl-mixture-model.

4.1 Validation of Samplers

Like many hierarchical Bayesian models used in practice, the posterior distribution $P(\mathbf{Z}|\mathbf{X})$ of cluster assignments $\mathbf{Z}$ given data $\mathbf{X}$ is intractable for even moderately sized n due to the combinatorial explosion in the number of possible partitions of n elements, i.e. the Bell numbers. The Metropolis-within-Gibbs sampling algorithms described in the previous section allow us to approximate this posterior distribution. One challenge of developing MCMC algorithms is that it is not straightforward to determine whether the output of the algorithm is correct, since MCMC algorithms provide approximations of a posterior distribution which often cannot be expressed analytically or computed in reasonable time. Incorrectly programmed MCMC algorithms can still output results which may seem reasonable,

but are otherwise invalid. Therefore, before studying the efficacy of our method, it is important to first validate that the samplers are indeed targeting the correct posterior.

We apply one method of validating our MCMC algorithms described in the thesis of Briercliffe [29]. The idea is that for reasonably small n, we can determine the true posterior probabilities of all clusterings on n elements by enumerating all such clusterings and computing their joint likelihood, since we have

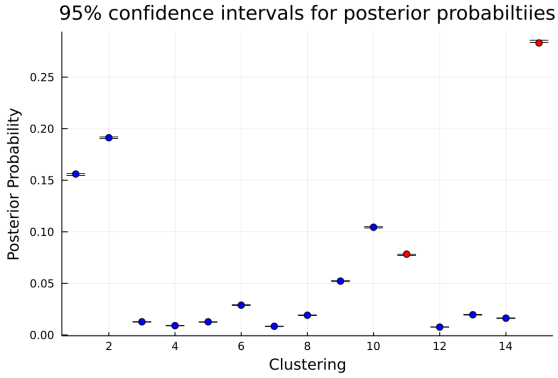
$$P(\mathbf{Z}|\mathbf{X}) = \frac{P(\mathbf{X},\mathbf{Z})}{\sum_{\mathbf{Z}'} P(\mathbf{X},\mathbf{Z}')}.$$

After computing these true posterior probabilities, we can then consider the problem of sampler validation from a frequentist point of view, where we construct a 95% confidence interval using the output of the MCMC algorithm. Standard errors of the estimates from the MCMC output are computed using the batch means method described in [30]. If the vast majority of the true posterior probabilities are captured by the confidence intervals, this is evidence that the samplers are correctly implemented.

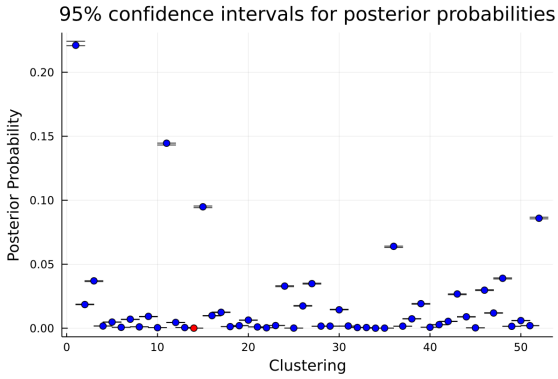
For $n \geq 3$, all cases of the collapsed Gibbs proposal described in section 3.3.1 will be computed at least once for sufficiently large number of iterations. In particular, consider an initial clustering $(Z_1 = 1,\ Z_2 = 2,\ Z_3 = 3)$ for $n = 3$. Sampling from the conditional probability $P(Z_1|\mathbf{X},\ Z_2 = 2,\ Z_3 = 3)$ covers the second subcase of case 1 and the second subcase of case 3. Sampling $P(Z_2|\mathbf{X},\ Z_1 = 1,\ Z_3 = 3)$ covers the first subcase of case 1, case 2, as well as the first subcase of case 3.

The implementation of the Metropolis–Hastings sampler used in this thesis uses the collapsed Gibbs proposal as a subroutine when calculating the acceptance ratio 3.4, so $n \geq 3$ is also sufficient to confirm validity of the Metropolis–Hastings sampler.

Figures 4.1 and 4.2 depict confidence intervals of the posterior probabilities on two clusterings on $n = 4$ and $n = 5$ elements, given by both the collapsed Gibbs and Metropolis–Hastings samplers. For both datasets, data were generated assuming a multinomial distribution at each cluster, with dimension $d = 10$ and $m = 5$ counts per observations. The parameters of the multinomial distributions were given Dirichlet distribution priors with scale parameter that is identically one for

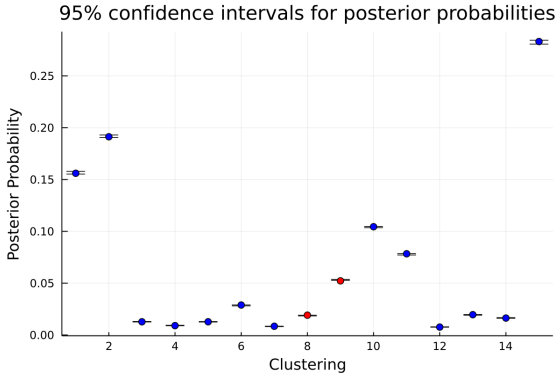


(a) $n = 4$ observations; 15 clusterings

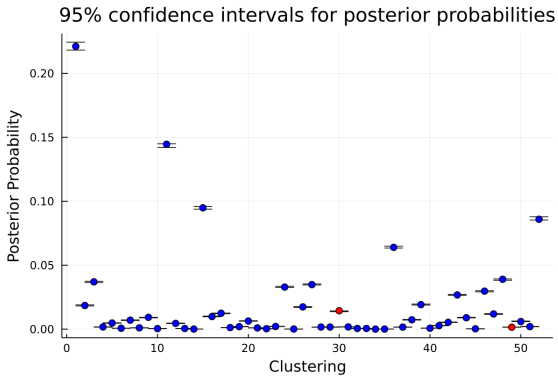


(b) $n = 5$ observations; 52 clusterings

Figure 4.1: Estimates of the true posterior probability of clusterings on (a) $n = 4$ and (b) $n = 5$ observations, given by the collapsed Gibbs sampler. Estimates are based on 1×10^6 iterations, with 1×10^5 burn in iterations. Blue dots indicate that the 95% confidence interval outputted by the sampler captures the true posterior probability, and red indicates that it does not. Note that the figures depict vertical 95% confidence intervals about each point, though almost all are too small to be visible at this scale.



(a) $n = 4$ observations; 15 clusterings



(b) $n = 5$ observations; 52 clusterings

Figure 4.2: Estimates of the true posterior probability of clusterings on (a) $n = 4$ and (b) $n = 5$ observations, given by the Metropolis–Hastings sampler. Estimates are based on 1×10^6 iterations, with 1×10^5 burn in iterations. Blue dots indicate that the 95% confidence interval outputted by the sampler captures the true posterior probability, and red indicates that it does not. Note that the figures depict vertical 95% confidence intervals about each point, though almost all are too small to be visible at this scale.

all components.

Over all experiments, 7 true posterior probabilities were not captured by their respective confidence intervals, with 3 probabilities not captured by the collapsed Gibbs sampler, and 4 probabilities not captured by the Metropolis–Hastings sampler. There are 67 clusterings in total over $n = 4$ and $n = 5$ observations, so we expect $67/20 \approx 3$ probabilities to not be captured by their 95% confidence intervals for each sampler. Therefore, this is evidence that both the collapsed Gibbs and Metropolis–Hastings samplers were correctly implemented.

4.2 Evaluation procedure

Apart from determining the time efficiency for each sampler (which will be measured in units of number of iterations per second), the efficacy of the NTL mixture model and its respective Metropolis-within-Gibbs samplers on simulated and real data are studied using the procedure outlined in the subsections that follow.

4.2.1 Convergence diagnostics

Asymptotically, the stationary distributions of the Markov chains outputted by the collapsed Gibbs and Metropolis–Hastings algorithms are guaranteed to be equal to the posterior distribution $P(\mathbf{Z}|\mathbf{X})$. Of course, we can never run either algorithm for infinite time in practice, so when using the NTL mixture model to analyze real data, we must assess whether the output of the algorithm we have used has reasonably converged.

Commonly, when a DPMM is used to cluster data using MCMC, three quantities are analyzed to assess convergence [31, 32]:

1. the number of clusters per iteration,
2. the joint log likelihood $\log P(\mathbf{Z}, \mathbf{X})$ for each iteration, and
3. the posterior value of the concentration parameter α.

Naturally, since the DPMM is a specific parameterization of the NTL mixture model, we can also analyze the above three quantities to assess convergence of

an NTL mixture model, with the caveat being that we analyze the posterior distribution of the $\phi \sim \text{Beta}(1,1)$ parameter of the *iid* interarrival time distribution $\Delta_j \sim \text{Geom}(\phi)$ instead of the α parameter, which does not appear in this parameterization of the NTL mixture model.

When analyzing the number of clusters, joint log likelihoods, and samples from the posterior distribution of ϕ, we look for stationarity in the trace plots. Nonstationarity (such as existence of trends) will suggest that more MCMC iterations are needed. We also ran several chains stemming from different initial cluster assignments, and analyzed the time series plots of the quantities described above. If, after a set number of iterations that is shared across chains, the chains are markedly different, this is an indication that the number of iterations is not sufficient.

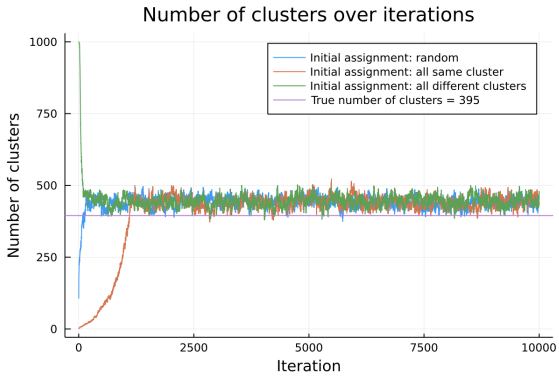
Notice that the posterior distribution of parameter values θ_s for each cluster s is not included in the above list, which is a quantity that would most likely be analyzed for other hierarchical Bayesian models with finite number of parameters. The reason this is the case is because of the assumed infinitude of the number of clusters in the model – since the number of clusters is likely to change across iterations, it is not always straightforward to determine which parameter values are associated to which cluster, especially since it is possible for some clusters to disappear in a given iteration.

Figure 4.3 depicts an example of convergence diagnostic plots for a clustering model fitted using an MCMC algorithm that was ran for 10000 iterations. The chains for the first and last initializations (random assignment, and assigning each observation to its own cluster) appear to converge rather quickly. However, the chain for the second initialization (assigning all observations to a single cluster only) converges after around 1250 iterations. Therefore, the first 1250 iterations should be discarded as burn-in, and only the last 8750 or so iterations can be reasonably used for downstream analysis.

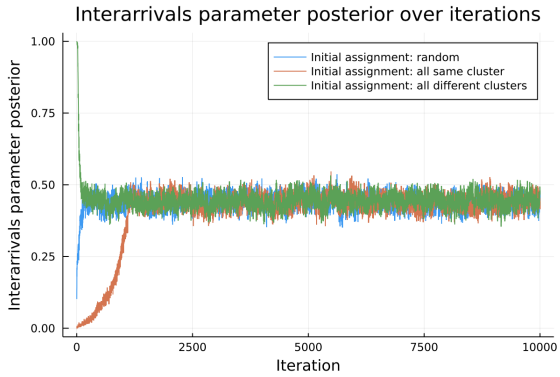
4.2.2 Assessing similarity of posterior clusterings to data-generating clustering

After assessing convergence of the Metropolis-within-Gibbs algorithms, it is also useful to understand the extent to which the algorithms output clusterings which resemble the data-generating clustering. One visual way to assess the performance

(a) Log likelihood



(b) Number of clusters



(c) Arrivals parameter posterior

Figure 4.3: Examples of convergence diagnostic plots for chains from three different initializations, with trace plots for (a) log likelihood, (b) number of clusters, and (c) arrival parameter posterior.

of clusterings is to compute the co-occurrence matrix, which is an $n \times n$ matrix M where entry $M_{i,j}$ is the proportion of posterior clusterings where observations i and j are within the same cluster. Using simulated data where the data-generating clusterings are known *a priori*, we can compare the co-occurrence matrix of the data-generating clustering with the co-occurrence matrix of the posterior clusterings, which will give us a visual depiction of the quality of our methods.

Another common method for assessing clustering performance that is used in practice is by computing the adjusted rand index (ARI) between the proposed clustering and the data-generating clustering [33]. The ARI of two clusterings takes on values bounded above by 1, with an ARI of 1 indicating perfect agreement between the two clusterings. However, since the MCMC algorithms described in the previous section output many different clusterings, the question arises of which clusterings do we compare to the data-generating clustering? One approach is to compute the ARI between the data-generating clustering and each clustering in the output Markov chain of the MCMC algorithms. This provides us a posterior distribution over the ARI with the data-generating clustering, which we can then further analyze by, for example, plotting the empirical distribution of the ARI values, or computing functions over the empirical ARI distribution such as the mean ARI.

4.2.3 Bayes estimates of clusterings and evaluation

Although the above procedure is useful for understanding the quality of draws from the posterior distribution of clusterings, in practice a single representative clustering is often needed for downstream analysis. Given that our MCMC algorithms output approximate samples from the posterior distribution of clusterings, we may wish for a method that takes advantage of the information contained in the posterior samples when constructing a representative clustering.

In Bayesian statistics, a principled approach for estimating an unknown parameter based on observed data is to choose a parameter which minimizes the posterior expectation of some appropriately chosen loss function [34]. More formally, suppose that θ is an unknown parameter we wish to estimate, and we are given a loss function $\ell(\theta, \tilde{\theta})$. Then a *Bayes estimator* $\tilde{\theta}$ of θ based on the loss function $\ell(\theta, \tilde{\theta})$

minimizes the value

$$\mathbb{E}_{\theta\sim\pi}[\ell(\theta,\tilde{\theta})|\mathbf{X}] = \int \ell(\theta,\tilde{\theta})\pi(\theta|\mathbf{X})\mathrm{d}\theta$$

where $\pi(\theta|\mathbf{X})$ is the posterior distribution of θ based on observations $\mathbf{X}$.

The choice of taking a *maximum a posteriori* (MAP) estimate of θ can be seen as taking a Bayes estimator of θ, as it can be shown that the MAP estimate minimizes the posterior expectation of the 0-1 loss $\ell(\theta,\tilde{\theta}) = \mathbf{1}_{\tilde{\theta}\neq\theta}(\theta,\tilde{\theta})$ when θ takes on discrete values, and is the limit of the sequence of Bayes estimators of the losses $\ell_\varepsilon(\theta,\tilde{\theta}) = \mathbf{1}_{|\tilde{\theta}-\theta|>\varepsilon}(\theta,\tilde{\theta})$ as $\varepsilon\to 0$ in the continuous case [34].

In general, we constructed Bayes estimates of the data-generating clustering using the following three loss functions:

1. the 0-1 loss function $\mathbf{1}_{\tilde{\mathbf{Z}}_n\neq\mathbf{Z}}(\mathbf{Z},\tilde{\mathbf{Z}}_n)$ (which corresponds to MAP estimates),
2. Binder's loss function $\ell_B(\mathbf{Z},\tilde{\mathbf{Z}}_n)$ for clusterings (whose Bayes estimates will be denoted as Binder estimates), and
3. the variation of information loss function $\ell_{VI}(\mathbf{Z},\tilde{\mathbf{Z}}_n)$ for clusterings (whose Bayes estimates will be denoted as VI estimates).

We briefly describe Binder's and variation of information loss functions, though we refer readers to Wade and Ghahramani [35] for complete descriptions and comparisons of both losses.

Binder's loss function $\ell_B(\mathbf{Z},\tilde{\mathbf{Z}})$ for clusterings, first introduced in [36], adds a penalty whenever two observations are contained in the same cluster in one of $\tilde{\mathbf{Z}}$ or $\mathbf{Z}$, but in different clusters in the other. When all errors have the same penalty, $\ell_B(\mathbf{Z},\tilde{\mathbf{Z}})$ can be represented as a quadratic function of the counts. This loss function accounts for basic symmetries when representing clusterings as a vector of cluster assignments $\mathbf{Z}$. In particular, $\ell_B(\mathbf{Z},\tilde{\mathbf{Z}})$ satisfies the following invariance properties:

1. permutation invariance of the order of the observations, and
2. permutation invariance of the cluster labels.

The variation of information loss function $\ell_{VI}(\mathbf{Z},\tilde{\mathbf{Z}})$, first introduced in [37], takes an information theoretic point of view when comparing clusterings, where

the loss can be written as

$$\ell_{VI}(\mathbf{Z},\tilde{\mathbf{Z}}) = H(\mathbf{Z}) + H(\tilde{\mathbf{Z}}) - 2I(\mathbf{Z},\tilde{\mathbf{Z}})$$

where $H(\mathbf{Z})$ is the entropy of the clustering $\mathbf{Z}$, and $I(\mathbf{Z},\tilde{\mathbf{Z}})$ is the mutual information of clusterings $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$. Intuitively, the variation of information penalizes uncertainty H captured in either of the clusterings $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$, but this penalty is reduced by the information I that is shared between clusterings $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$.

For the MAP estimate, we simply took the clustering from the MCMC output that maximizes the joint log likelihood. We used the **R** package mcclust.ext [38] (which extends from the package mcclust [39]) to compute the Binder and VI estimates, where the functions used were `minbinder.ext` and `minVI`, respectively. For both estimates, we used the argument `method = 'draws'` so that the Binder and VI estimates were taken from the MCMC samples of the posterior, to match our method of computing the MAP estimate.

For each of the above three loss functions, the collapsed Gibbs and Metropolis–Hastings algorithms were run until the last 50% iterations depict convergence, those 50% of samples were taken from the posterior, and the Bayes estimate of the data-generating clustering was computed using the respective loss function. Finally, the ARI was used to determine the similarity between the Bayes estimate and the data-generating clustering.

4.2.4 Alternative clustering methods

Where applicable, results from the NTL mixture model will also be compared to results from the following popular clustering methods:

1. The Dirichlet process mixture model, where convergence is assessed by observing the trace plots of the joint log likelihood, number of clusters, and posterior value of the α parameter. Three chains with different initializations were run, convergence of the last 50% of iterations was checked using the convergence diagnostics [1], and the last 50% of samples were chosen for final computations.

[1] See figures A.1 and A.2 for convergence diagnostics.

2. k-means clustering, where k is chosen via the elbow method [16] using the plot of the distortion scores (sum of squared distances of each observation to the center of its assigned cluster) for each k. [2]

Dirichlet process mixture models were fit using a collapsed Gibbs sampler of the form

$$P(Z_i = s|\mathbf{Z}_{-i}, \mathbf{X}) \propto \begin{cases} P(X_i|Z_i = s, \mathbf{Z}_{-i}, \mathbf{X}_{-i}) n_{-i}(s) & \text{if } s \text{ is an existing cluster} \\ P(X_i|Z_i = s, \mathbf{Z}_{-i}, \mathbf{X}_{-i}) \alpha & \text{if } s \text{ is a new cluster.} \end{cases}$$

We sampled the posterior value of α using a Metropolis–Hastings step, where the prior on α was uniform over the interval $(0, n)$ and the proposal distribution was a Gaussian distribution centered around the previous value of α with variance 1, and truncated from 0 to n. A uniform distribution was chosen for the prior to reflect agnostic belief regarding the value of α (no value of α is preferable over another). The upper limit of the uniform distribution was chosen to be n to allow the value of α to grow with the size n of the dataset (since larger datasets may have more clusters, which the α parameter controls), while maintaining integrability. [3]

4.3 Synthetic data

We considered the following parameterization of the NTL mixture model prior:

$$a(j, T_j) = 1, b(j, T_j) = 1 \text{ for all } j \geq 2,$$

$$\Delta_j := T_j - T_{j-1} \overset{\text{iid}}{\sim} \text{Geom}(\phi), \phi \sim \text{Beta}(1, 1).$$

Under this parameterization, the distributions of stick breaking weights $(\psi_s)_{s=2}^{K_n}$ as well as relative frequency of the arrival of new clusters are *iid* uniform over the unit interval $[0, 1]$, which induces clusters whose expected sizes are asymptotically bounded conditionally on ϕ, as per proposition 1.

[2] See Figure A.3 for the distortion plots.

[3] We chose not to use the improper prior of a uniform distribution over the positive real line on the philosophical belief that a prior distribution should be a probability distribution over its support, and thus must have finite integral over its domain.

We also considered the following likelihoods:

$$X_i|Z_i = s \sim \mathcal{N}(\mu_s, \sigma^2 I),\ \mu_s \sim \mathcal{N}(\underline{0}, I) \text{ for all } s$$

with dimension equal to 2, $\sigma^2 = 0.1$, and

$$X_i|Z_i = s \sim \text{Multinomial}(m, p_s),\ p_s \sim \text{Dirichlet}(\underline{1}) \text{ for all } s$$

with dimension equal to 10 and $m = 10$ counts per observation.

For all parameterizations, we generated $n = 100$ observations from the prior. All parameterizations were given the same data-generating clustering, which will allow analyses on how changes in the likelihood may change the performance of the method.

4.3.1 Characteristics of the data-generating clustering

In this subsection, we examine the characteristics of the data-generating clustering of the simulated data.

Figure 4.4 depicts the co-occurrence matrix of the data-generating clustering, which was used to generate both the multinomial and multivariate Gaussian data. We see that the matrix depicts a block diagonal structure, where clusters contain observations which are close to each other in time. Given the *iid* interarrival times, cluster sizes are quite small in comparison to the total number of observations.

Figure 4.5 depicts the assignment of observations to clusters over time. We see that there are 39 clusters out of 100 observations.

Figure 4.6 depicts a histogram of the sizes of the clusters in the data-generating clustering. The histogram resembles a geometric distribution with mean approximately equal to 2.56. Proposition 1 implies that when the interarrival time distribution is parameterized to be a geometric distribution with some fixed probability ϕ, the expected number of observations assigned to a given cluster is $\frac{1}{\phi}$. Figure 4.5 shows that there are 39 clusters out of 100 observations, therefore ϕ is approximately 0.4. Proposition 1 then predicts that the expected number of observations assigned to a cluster is asymptotically $\frac{1}{\phi} \approx \frac{1}{0.4} = 2.5$, which approximately matches what we see as the mean cluster size in Figure 4.6.

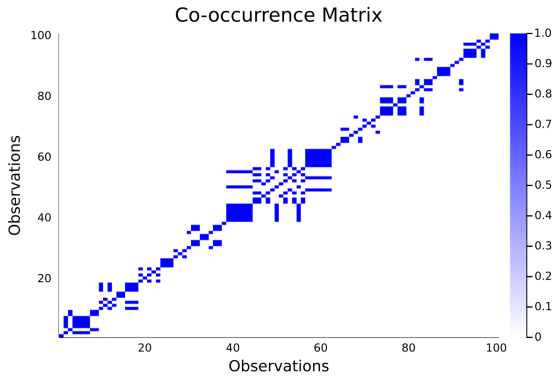


Figure 4.4: The co-occurrence matrix of the data-generating clustering of both the simulated multinomial and multivariate Gaussian data.

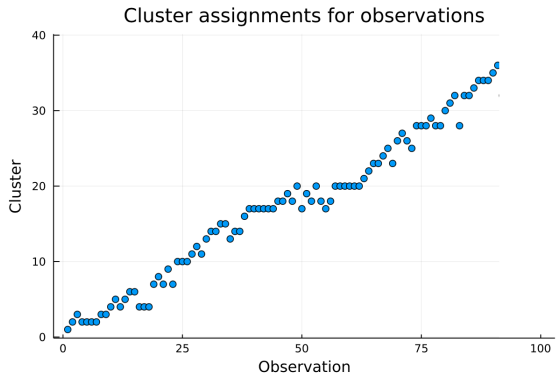


Figure 4.5: Assignments of the observations to clusters over time of the data-generating clustering.

4.3.2 The Metropolis–Hastings sampler is more time efficient than the collapsed Gibbs sampler and the DPMM sampler

Table 4.1 depicts the number of iterations per second performed by the Metropolis–Hastings and collapsed Gibbs samplers for the NTL mixture model, as well as the collapsed Gibbs sampler for the DPMM, on multivariate Gaussian and multinomial data. As predicted by the time complexity analysis of the samplers, the Metropolis–Hastings sampler is one order of magnitude faster than the collapsed Gibbs sampler, owing to its $O(nK_n)$ time complexity per sweep as opposed to the collapsed Gibbs sampler $O(nK_n^2)$ complexity.

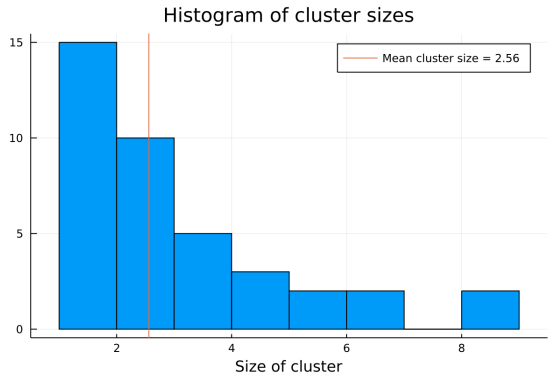
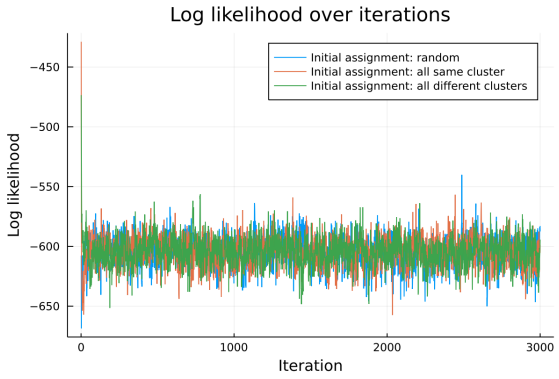


Figure 4.6: Histogram of the sizes of the clusters, with the mean size of the clusters indicated on the figure.

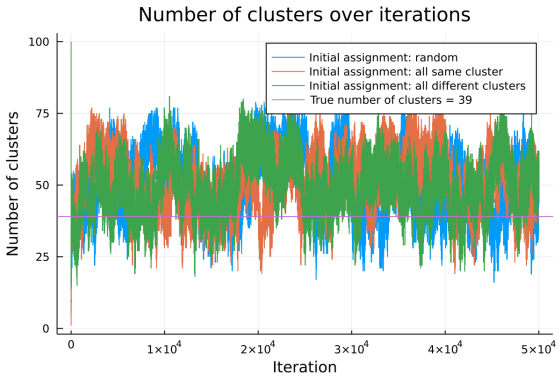
Data	Number of iterations per second		
	Metropolis–Hastings	Collapsed Gibbs	DPMM
Multivariate Gaussian	303.03	8.87	87.41
Multinomial	833.33	13.04	200.80

Table 4.1: Number of iterations per second for the Metropolis–Hastings and collapsed Gibbs sampler, as well as the collapsed Gibbs sampler for the DPMM, on multivariate Gaussian and multinomial data.

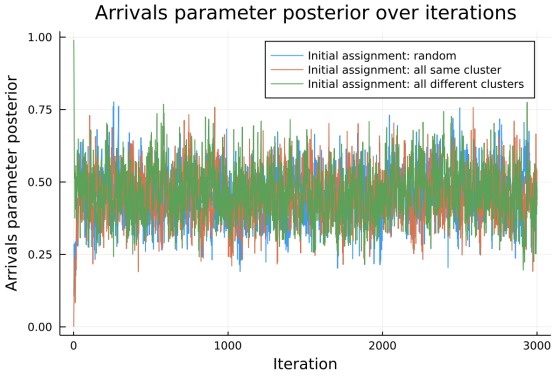
The Metropolis–Hastings sampler is also faster than the collapsed Gibbs sampler for the DPMM. Although both have time complexity $O(nK_n)$ per sweep, the reason for the Metropolis–Hastings sampler's better time efficiency in practice may be due to the fact that the K_n term in the time complexity stems from the need to compute the product in the expression for case 2 of $P(Z_i = s, \mathbf{Z}_{-i})$, when computing the acceptance ratio 3.4. However, in practice, it is likely that the number of terms in this product is significantly less than K_n, since more terms in the product reduces the probability that the proposal will be accepted. On the other hand, for the collapsed Gibbs sampler for the DPMM, the proposal density for each cluster must be computed, so exactly $K_n^{-i} + 1$ computations are required when sampling from $P(Z_i = s | \mathbf{Z}_{-i}, \mathbf{X})$ for the DPMM.



(a) Log likelihood over iterations



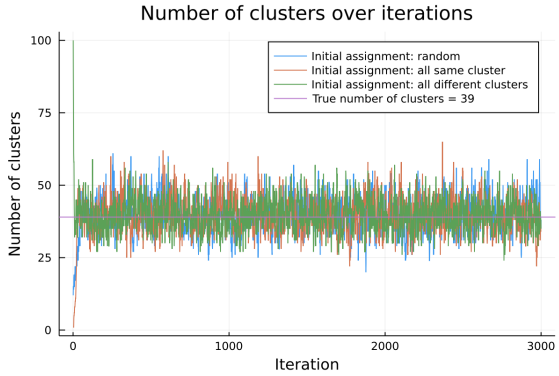
(b) Number of clusters over iterations



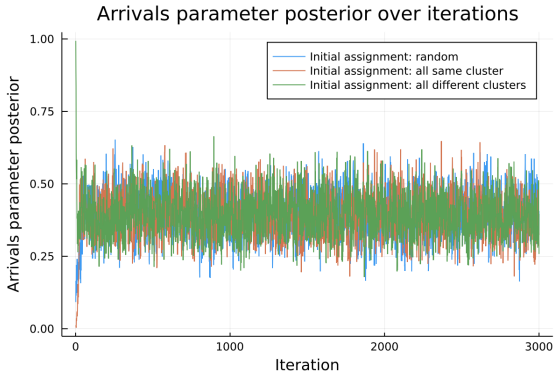
(c) Arrivals parameter posterior over iterations

Figure 4.7: Convergence diagnostic plots for collapsed Gibbs sampler on multivariate Gaussian data.

(a) Log likelihood over iterations



(b) Number of clusters over iterations



(c) Arrivals parameter posterior over iterations

Figure 4.8: Convergence diagnostic plots for collapsed Gibbs sampler on multinomial data.

4.3.3 NTL mixture model samplers converge quickly

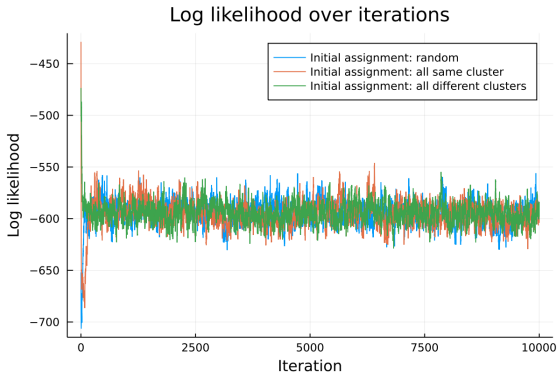
Figures 4.7 and 4.8 depict the convergence diagnostic plots for the collapsed Gibbs sampler on multivariate Gaussian and multinomial data, respectively. Both figures show quick convergence of the collapsed Gibbs sampler — the chains corresponding to the three different initial cluster assignments appear to converge to the same mode after approximately 100 iterations for both data types.

Figures 4.9 and 4.10 depict the convergence diagnostic plots for the Metropolis–Hastings samplers for the NTL mixture model. Similarly to the collapsed Gibbs sampler, these figures demonstrate that chains from the Metropolis–Hastings sampler converge quickly, with the chains from the three distinct initial cluster assignments showing convergence after approximately 500 iterations. This observation, along with the sampler's relatively efficient time complexity of $O(nK_n)$ for a single sweep, lends credence to using the Metropolis–Hastings sampler over the collapsed Gibbs sampler.

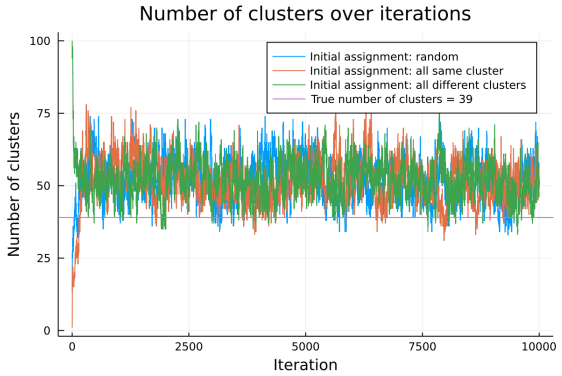
4.3.4 Co-occurrence matrices outputted by NTL mixture model capture block diagonal characteristics of data-generating clustering

Figure 4.11 depicts the co-occurrence matrices outputted by the collapsed Gibbs and Metropolis–Hastings sampler for the NTL mixture model, as well as the co-occurrence matrix given by the DPMM, all on multivariate Gaussian data. Figures 4.11a and 4.11b demonstrate that the collapsed Gibbs and Metropolis–Hastings samplers are able to recover some of the rich block diagonal structure of the co-occurrence matrix of the data-generating clustering shown in Figure 4.4. The collapsed Gibbs sampler seems to be more efficient in exploring clusterings of lesser likelihood than the Metropolis–Hastings sampler, with Figure 4.11a showing distant observations being paired together at a slightly larger frequency than what is shown in Figure 4.11b.

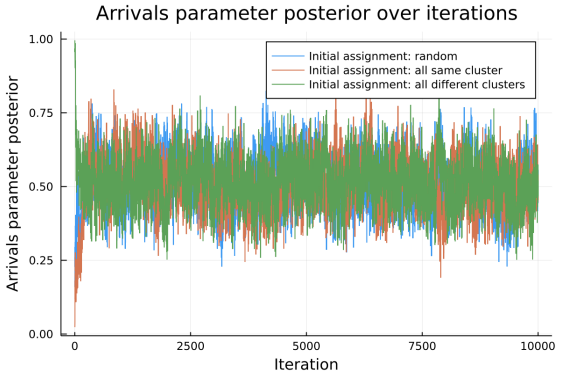
On the other hand, the DPMM fails to recover the block diagonal structure in its co-occurrence matrix, as shown in Figure 4.11c. This is to be expected, since the DPMM assumes exchangeability of the observations, and it seems that the relatively large within-cluster variance of each cluster's data distribution is not



(a) Log likelihood over iterations

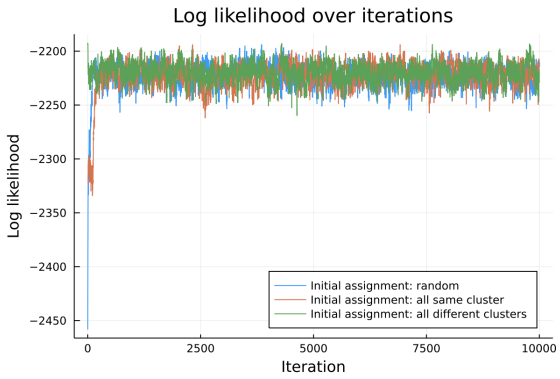


(b) Number of clusters over iterations

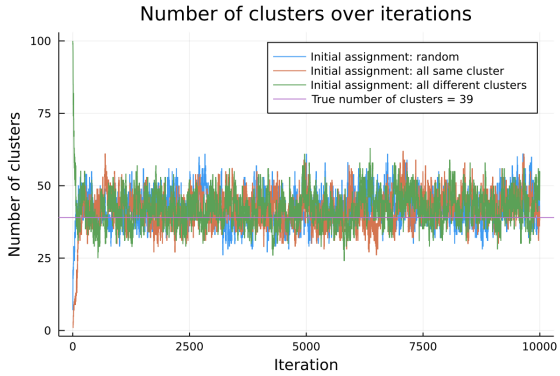


(c) Arrivals parameter posterior over iterations

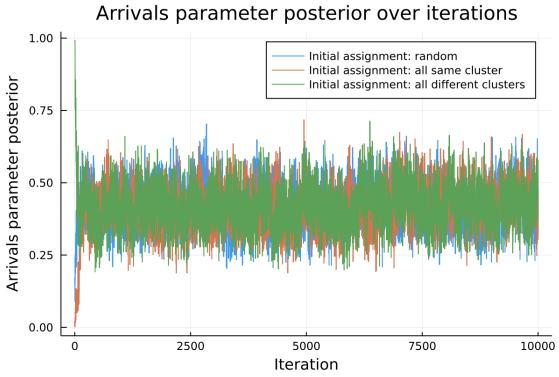
Figure 4.9: Convergence diagnostic plots for Metropolis–Hastings sampler on multivariate Gaussian data.



(a) Log likelihood over iterations

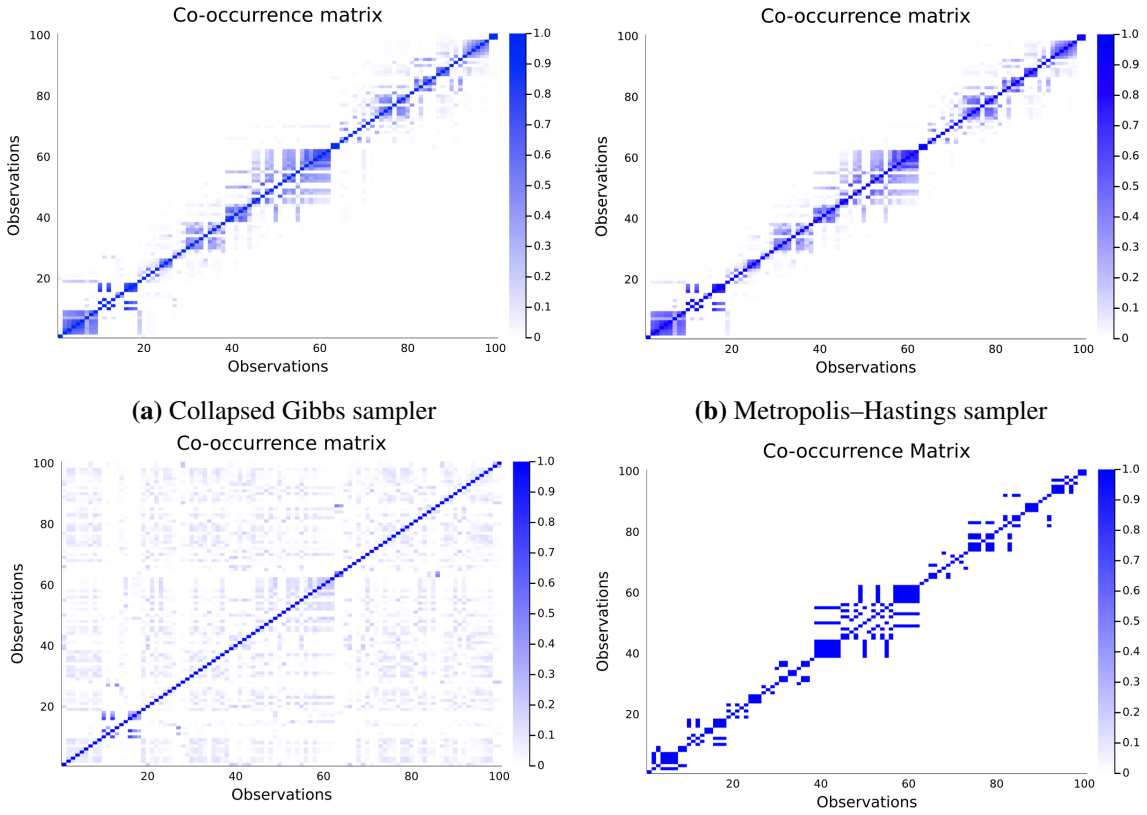


(b) Number of clusters over iterations



(c) Arrivals parameter posterior over iterations

Figure 4.10: Convergence diagnostic plots for the Metropolis–Hastings sampler on multinomial data.



(a) Collapsed Gibbs sampler

(b) Metropolis–Hastings sampler

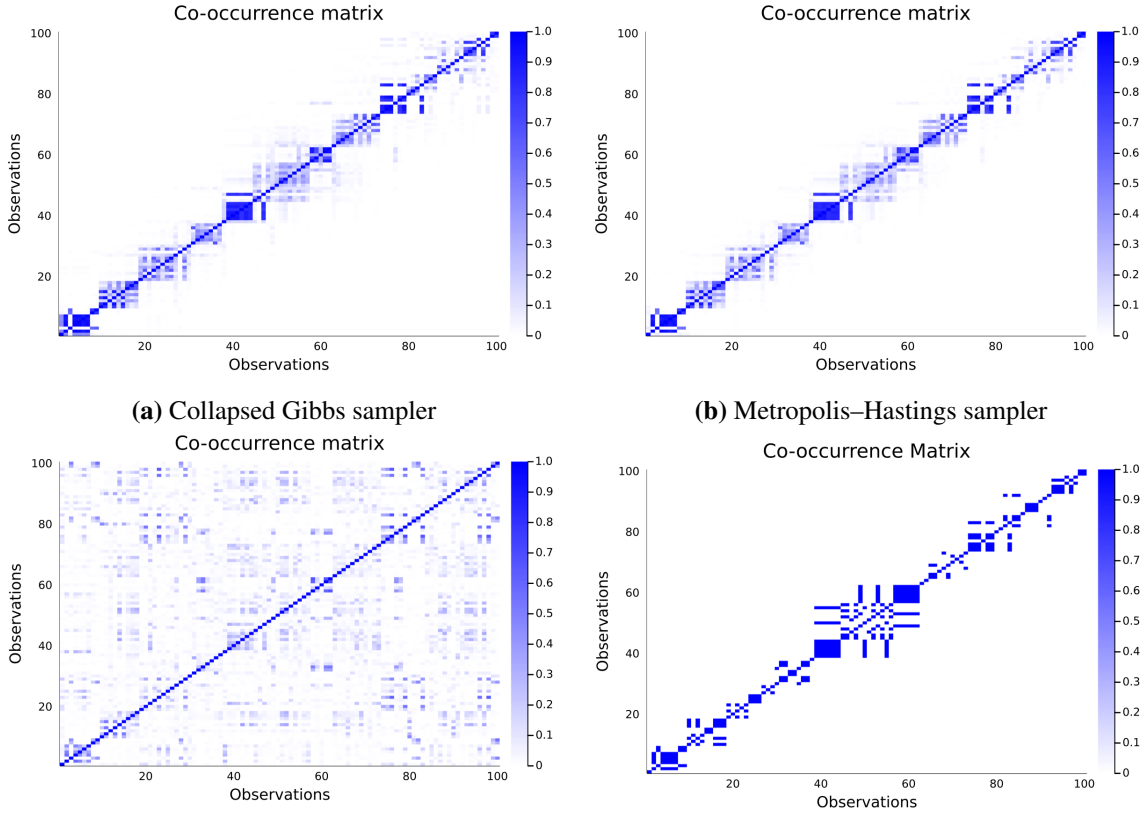
(c) Dirichlet process mixture model

(d) Data-generating clustering

Figure 4.11: Co-occurrence matrices of clusterings of multivariate Gaussian data given by (a) collapsed Gibbs Sampler, (b) Metropolis–Hastings sampler, and (c) DPMM. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

sufficiently strong to overcome the exchangeability property.

Figure 4.12 depicts the co-occurrence matrices of the collapsed Gibbs and Metropolis–Hastings samplers on multinomial data, as well as that of the DPMM. Similarly to the multivariate Gaussian case, Figures 4.12a and 4.12b show that the NTL mixture model samplers can recover some of the block diagonal structure, whereas the DPMM fails to do so as shown in Figure 4.12c.



(a) Collapsed Gibbs sampler

(b) Metropolis–Hastings sampler

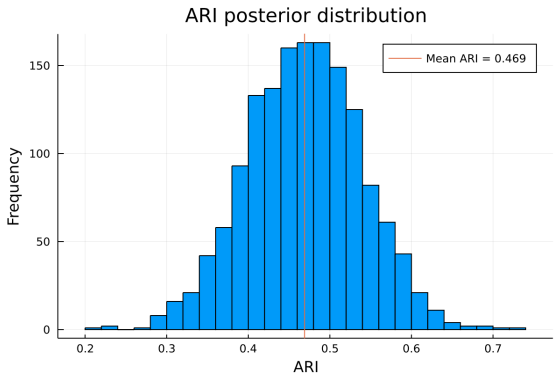
(c) Dirichlet process mixture model

(d) Data-generating clustering

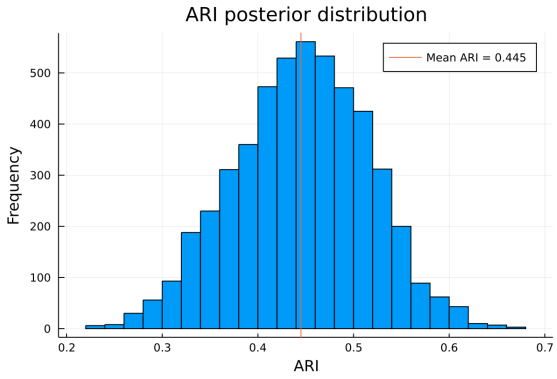
Figure 4.12: Co-occurrence matrices of clusterings of multinomial data given by (a) collapsed Gibbs sampler, (b) Metropolis–Hastings sampler, and (c) DPMM. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

4.3.5 Posterior distribution of ARI of NTL mixture model clusterings indicate greater fit to data than DPMM

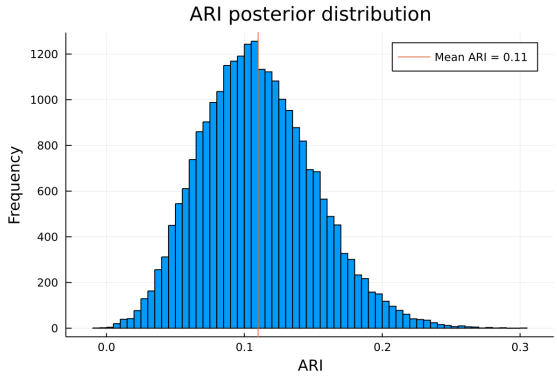
Figures 4.13 and 4.14 depict the posterior distribution of the ARI between the data-generating clustering and the posterior clusterings as given by the collapsed Gibbs sampler, Metropolis–Hastings sampler, and the DPMM, on multivariate Gaussian data and multinomial data respectively. The posterior distributions show that clusterings drawn from the NTL mixture model posterior have greater similarity to the underlying data-generating clustering than that of the DPMM.



(a) Collapsed Gibbs sampler

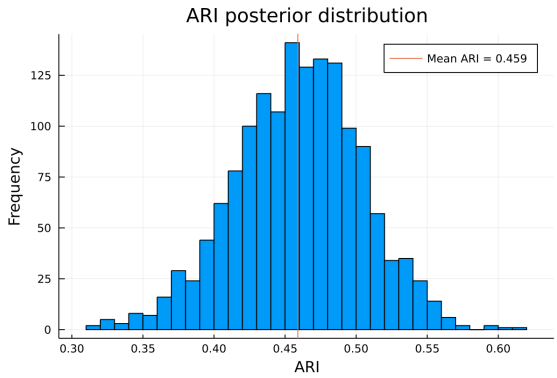


(b) Metropolis–Hastings sampler

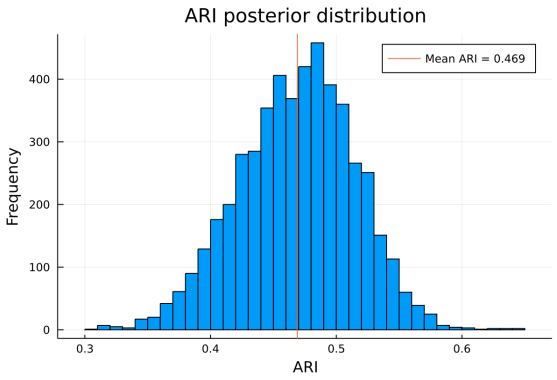


(c) Dirichlet process mixture model

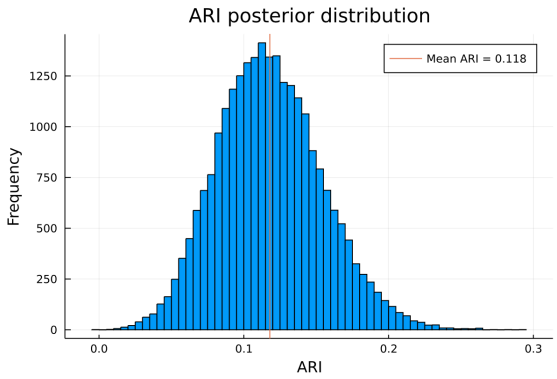
Figure 4.13: Posterior distribution of ARI between data-generating clustering and posterior clusterings from (a) collapsed Gibbs sampler, (b) Metropolis–Hastings sampler, and (c) DPMM, on multivariate Gaussian data.



(a) Collapsed Gibbs sampler

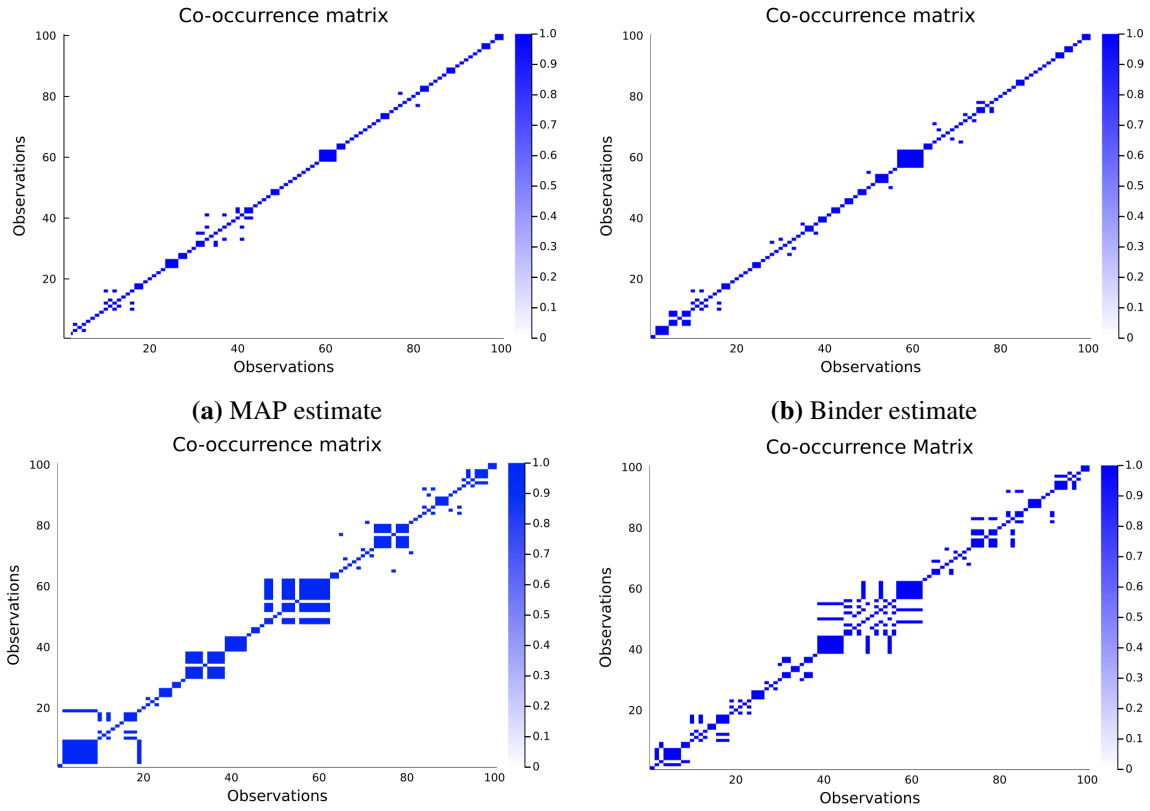


(b) Metropolis–Hastings sampler



(c) Dirichlet process mixture model

Figure 4.14: Posterior distribution of ARI between data-generating clustering and posterior clusterings from (a) collapsed Gibbs sampler, (b) Metropolis–Hastings sampler, and (c) DPMM, on multinomial data.



(a) MAP estimate

(b) Binder estimate

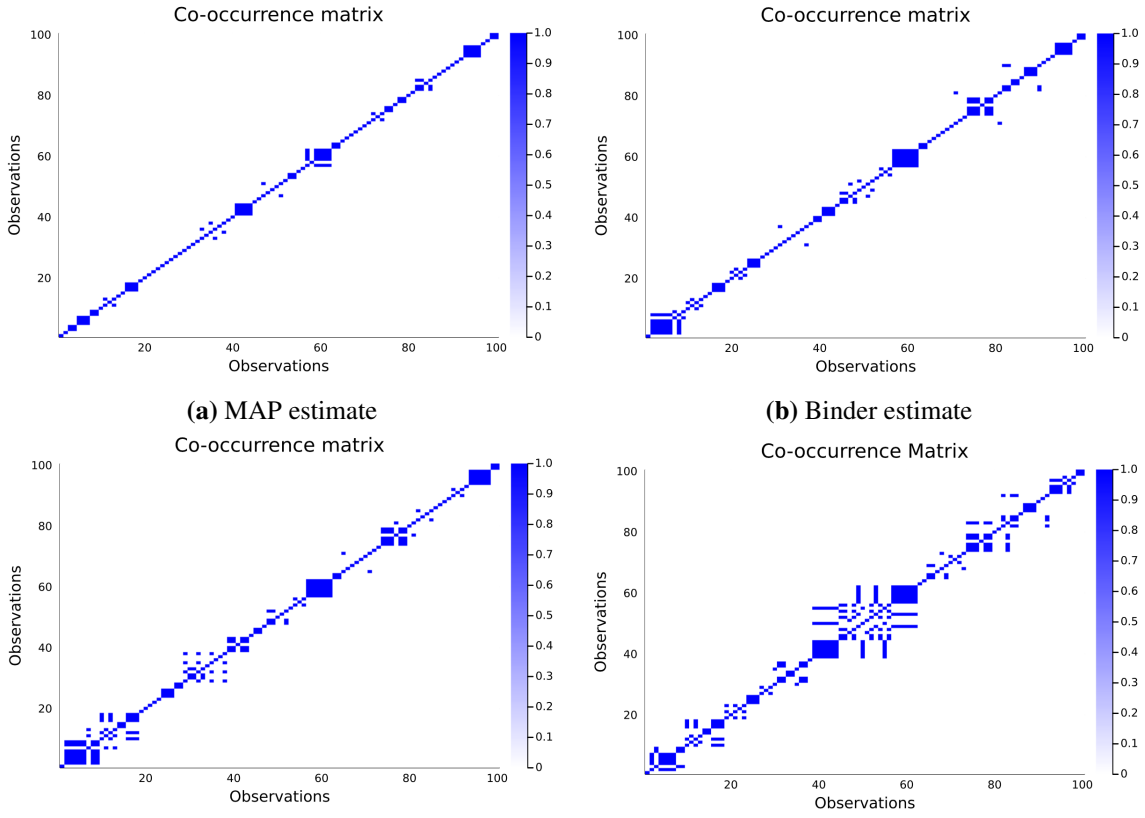
(c) VI estimate

(d) Data-generating clustering

Figure 4.15: Co-occurrence matrices of Bayes estimates of data-generating clustering using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate), on multivariate Gaussian data using the collapsed Gibbs sampler. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

4.3.6 Bayes estimates based on VI and Binder's loss more reliably estimate data-generating clustering than MAP estimates

Figures 4.15 and 4.16 depict the co-occurrence matrices of Bayes estimates of the data-generating clustering based on samples from the NTL mixture model posterior, with the former figure showing results from the collapsed Gibbs sampler, and the latter from the Metropolis–Hastings sampler. Figures 4.17 and 4.18 depict



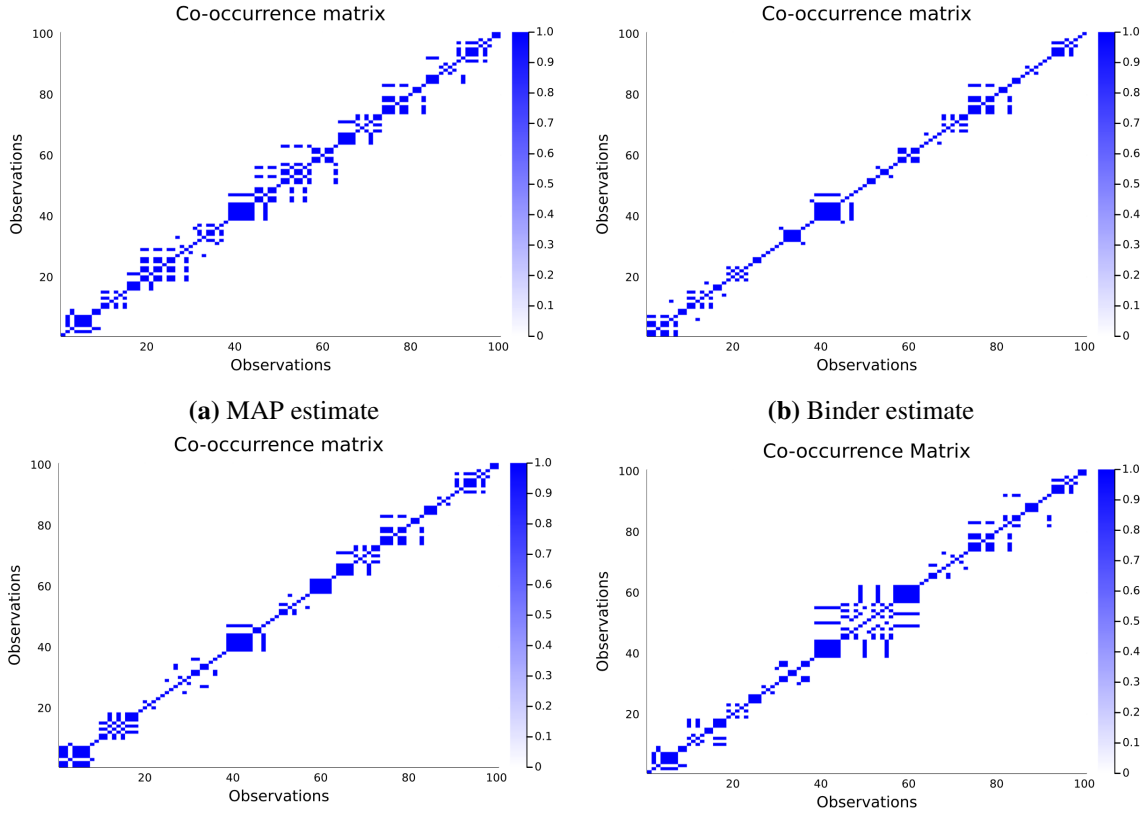
(a) MAP estimate

(b) Binder estimate

(c) VI estimate

(d) Data-generating clustering

Figure 4.16: Co-occurrence matrices of Bayes estimators of data-generating clustering using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate), on multivariate Gaussian data using the Metropolis–Hastings sampler. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.



(a) MAP estimate

(b) Binder estimate

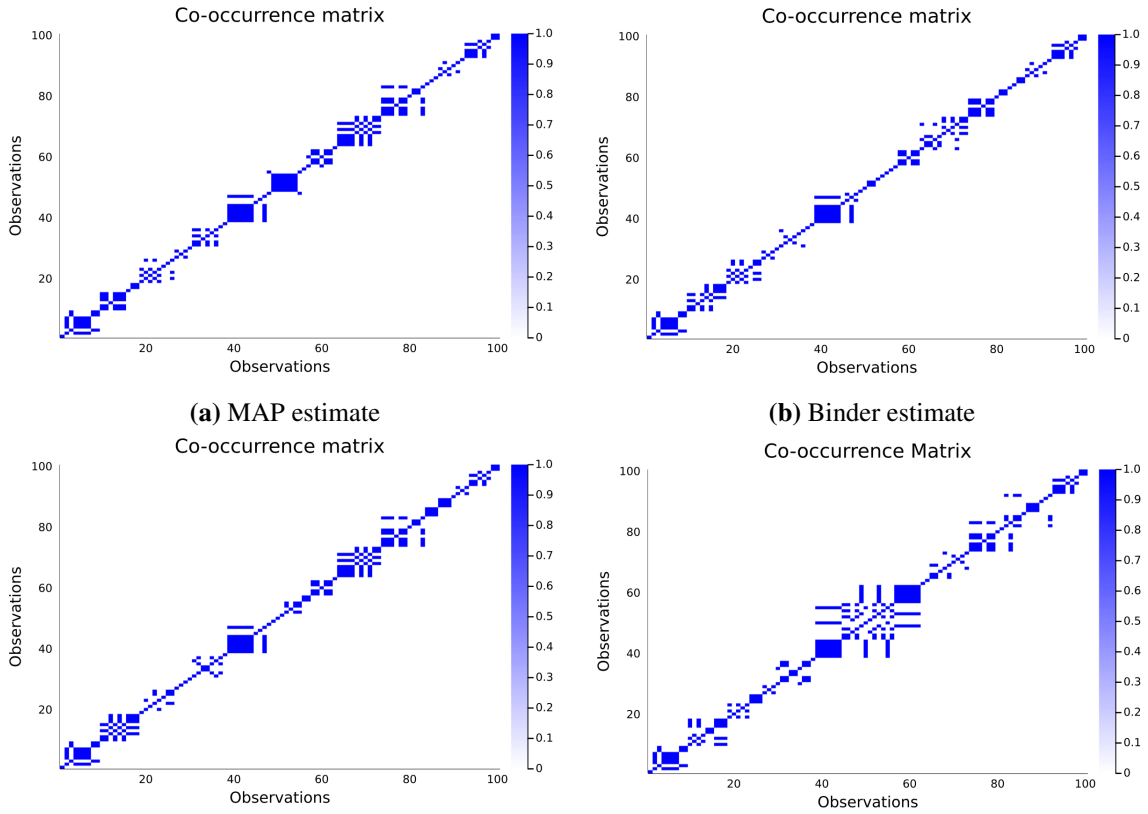
(c) VI estimate

(d) Data-generating clustering

Figure 4.17: Co-occurrence matrices of Bayes estimators of data-generating clustering using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate), on multinomial data using the collapsed Gibbs sampler. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

co-occurrence matrices of the Bayes estimates from the NTL mixture model from both samplers on multinomial data.

Qualitatively, the co-occurrence matrices from Bayes estimates based on the variation of information loss and Binder's loss reliably capture some of the block diagonal structure present in the data-generating clustering for both the multivariate Gaussian and multinomial data. On the other hand, the MAP estimate less reliably recovers a block diagonal structure. For example, it fails to do so in the multivariate



(a) MAP estimate

(b) Binder estimate

(c) VI estimate

(d) Data-generating clustering

Figure 4.18: Co-occurrence matrices of Bayes estimators of data-generating clustering using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate), on multinomial data using the Metropolis–Hastings sampler. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

Data	Sampler	ARI		
		MAP estimate	Binder estimate	VI estimate
Multivariate Gaussian	Collapsed Gibbs	0.211	0.356	0.557
	Metropolis–Hastings	0.324	0.509	0.502
Multinomial	Collapsed Gibbs	0.535	0.496	0.527
	Metropolis–Hastings	0.530	0.492	0.513

Table 4.2: ARI between Bayes estimates and data-generating clusterings for multivariate Gaussian and multinomial data, using output from collapsed Gibbs and Metropolis–Hastings sampler.

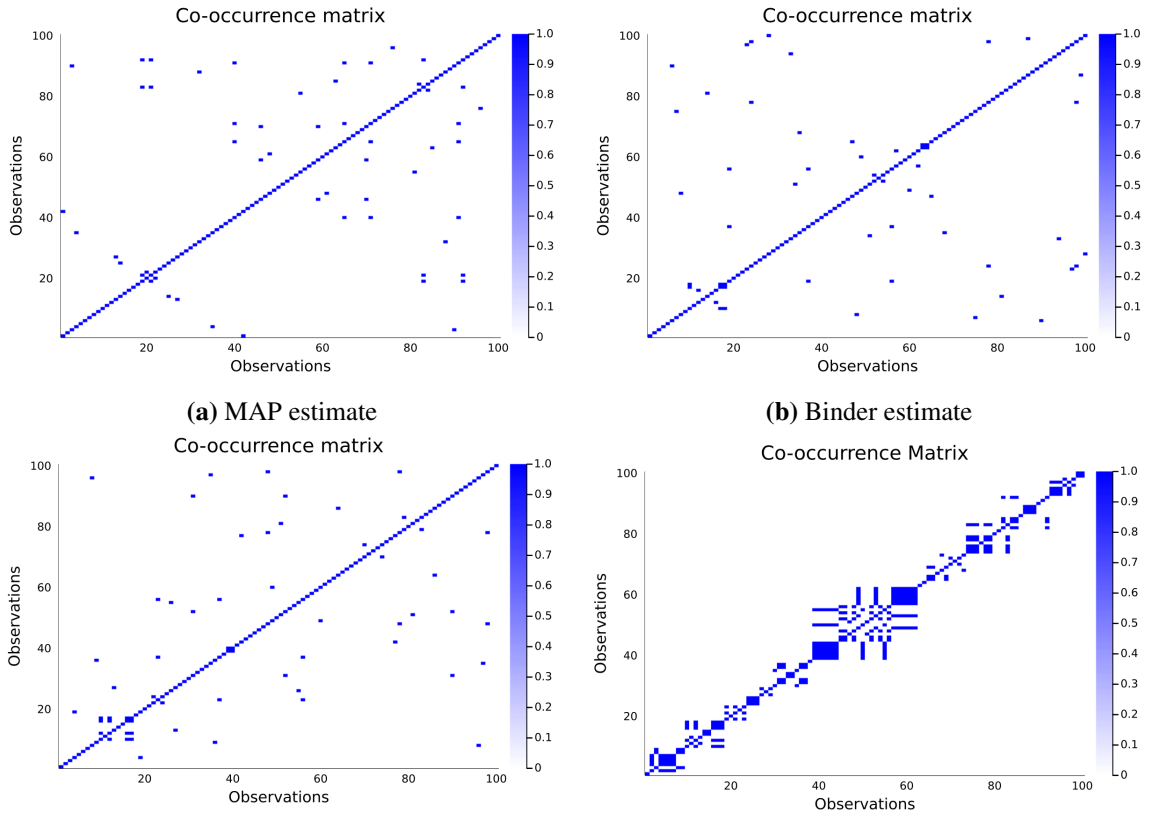
Gaussian case.

These qualitative results are supported by the quantitative measures depicted in Table 4.2 – the ARI of the MAP estimates fluctuate from 0.211 to 0.535 between the multivariate Gaussian and multinomial data, whereas the Binder and VI estimates only range from 0.356 to 0.509 in ARI for the former, and 0.502 to 0.557 for the latter.

Overall, the results suggest that the Variation of Information loss provides the greatest quality Bayes estimates of the data-generating clustering from both qualitative and quantitative points of view. The middling results from MAP estimates may be due to the fact that intuitively, the corresponding 0-1 loss is not appropriate for combinatorial problems such as clustering, since the loss assigns the same penalty when the two input clusterings differ with no regard to the extent to which they differ.

Figures 4.19 and 4.20 depict the MAP, Binder, and VI estimates using output from the DPMM on multivariate Gaussian and multinomial data, respectively. As can be seen in these two figures, point estimates from the DPMM fail to capture the block diagonal structure of the underlying data-generating clustering. The clusterings are also quite sparse in nature, with many singleton clusters present in the point estimates, save for the MAP estimate for multinomial data in Figure 4.20a.

Figure 4.21 depicts clustering of the multivariate Gaussian and multinomial data using k-means clustering. Similarly to the case of the DPMM point estimates, the estimates from k-means clustering do not capture the block diagonal structure of the data-generating clustering due to the assumption of exchangeability.



(a) MAP estimate

(b) Binder estimate

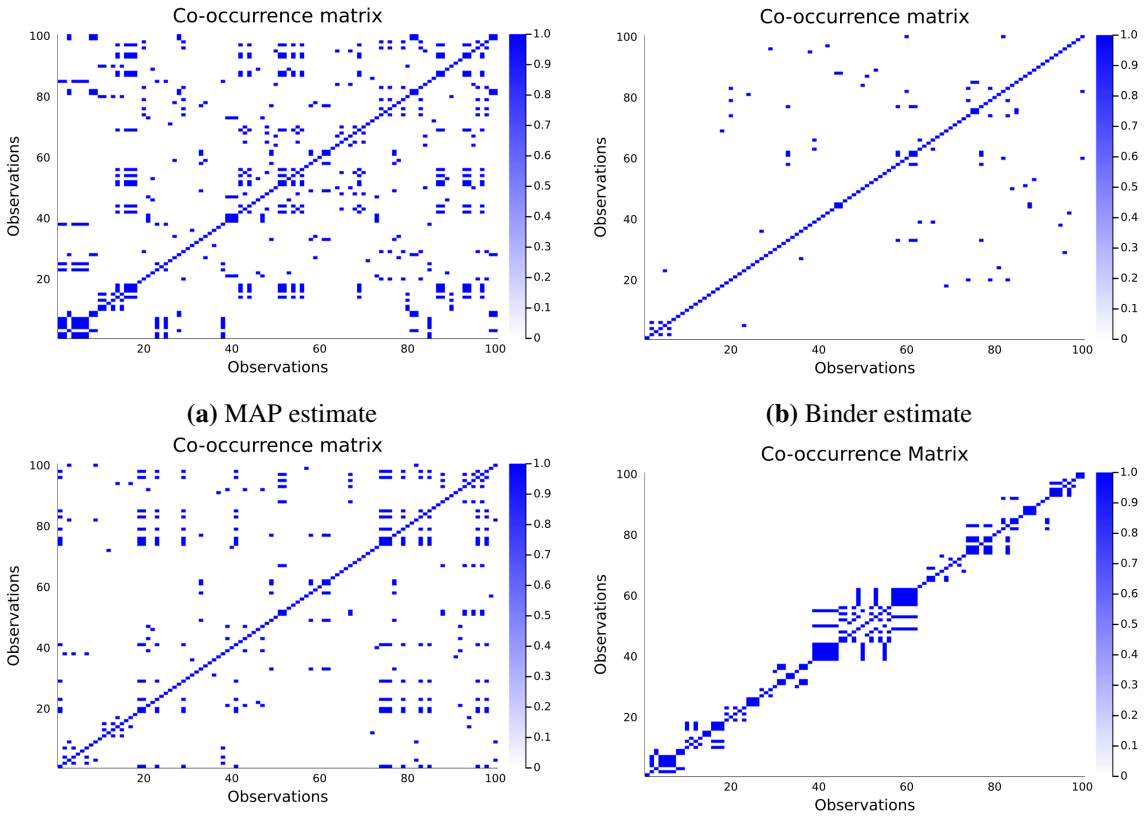
(c) VI estimate

(d) Data-generating clustering

Figure 4.19: Co-occurrence matrices of Bayes estimates of data-generating clustering with multivariate Gaussian data from DPMM, using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate). The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

	Dirichlet Process Mixture Model			
Data	*k*-means ARI	MAP estimate ARI	Binder estimate ARI	VI estimate ARI
Multivariate Gaussian	0.107	0.014	0.072	0.092
Multinomial	0.109	0.159	0.094	0.154

Table 4.3: ARI between point estimates from DPMM and *k*-means clustering and data-generating clusterings for multivariate Gaussian and multinomial data.



(a) MAP estimate

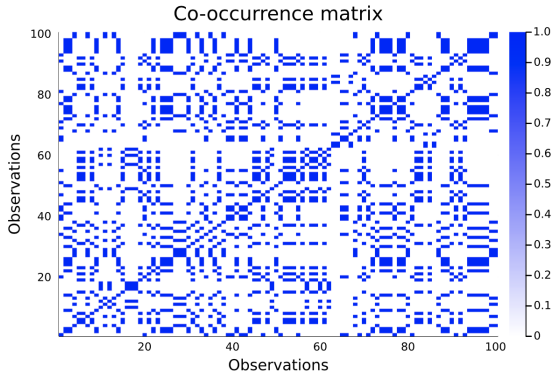
(b) Binder estimate

(c) VI estimate

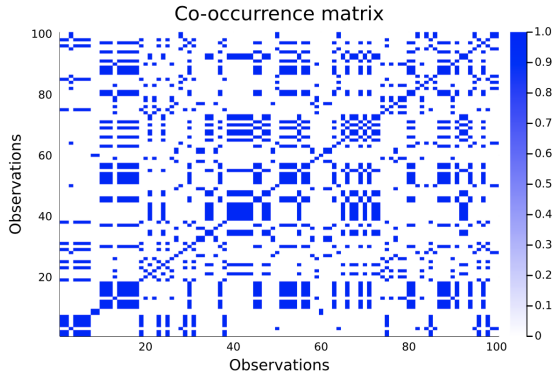
(d) Data-generating clustering

Figure 4.20: Co-occurrence matrices of Bayes estimates of data-generating clustering with multinomial data from the DPMM, using (a) 0-1 loss (MAP estimate), (b) Binder's loss (Binder estimate), and (c) variation of information loss (VI estimate). The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

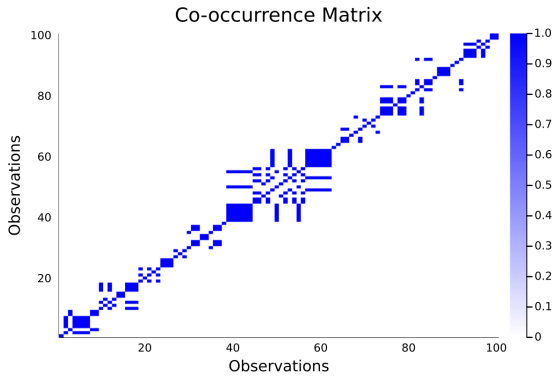
The qualitative lack of fit of the DPMM and k-means point estimates are supported by low ARI between the data-generating clustering and the respective estimates, which are shown in Table 4.3.



(a) Multivariate Gaussian data



(b) Multinomial data



(c) Data-generating clustering

Figure 4.21: Co-occurrence matrix of point estimates of data-generating clustering using k-means clustering, on (a) multivariate Gaussian data, and (b) multinomial data. The co-occurrence matrix of the (d) data-generating clustering is shown for comparison.

4.4 Real data

The parameterization of the NTL mixture model considered in this thesis has a natural interpretation as representing data that evolves over time (as can be seen visually in Figure 4.5), due to its non-exchangeable nature and the fact that under the NTL mixture model, the probability of assigning observations to a particular cluster degrades over time.

One type of data that may be well modelled by this parameterization of the NTL mixture model are tweets, and more specifically, the topic structure of tweets. We would expect that the set of topics that a Twitter user tweets about should evolve over time, where new topics emerge at some rate and older topics eventually disappear. A natural way to model textual data like tweets is by using a multinomial distribution, where each dimension of the multinomial distribution corresponds to a particular word, with the prior on the parameters of the multinomial distribution to be the conjugate Dirichlet distribution. Each cluster, then, is associated with a multinomial distribution with a particular set of parameters. In the language of topic modelling, this corresponds to modelling each cluster as being associated with one topic, a topic being a distribution over words [40]. Tweets assigned to the same cluster then have the same topic, and so we can expect tweets within the same cluster to have similar compositions of words.

We note that in topic modelling, it is more common to assume that each document is a distribution over multiple topics, as opposed to assuming that each document is associated with one topic as we do here. This is a valid assumption for large documents such as journal articles, but it may not be a good modelling choice for small documents such as tweets.

4.4.1 Experimental setup

We applied a parameterization of the NTL mixture model to model the cluster structure of President Joe Biden's tweets from 24 October 2007 to 31 October 2020 [41]. The last 100 tweets from this dataset were taken as the final dataset, and after removing stop words, definite and indefinite articles, prepositions, pronouns, numbers, non-letters, HTML tags, and frequent and infrequent terms, and stemming each word, we found that the dataset contained 588 unique word stems.

Each tweet was then represented as an integer vector with 588 components, with the integer at a component indicating the number of times the corresponding word stem appeared in the tweet. We then fitted the following NTL mixture model using the Metropolis–Hastings algorithm:

$$
\begin{aligned}
\phi &\sim \text{Beta}(1,1), \\
\Delta_j &\overset{\text{iid}}{\sim} \text{Geom}(\phi) \text{ for } j \geq 2, \\
\psi_j &\overset{\text{iid}}{\sim} \text{Beta}(1,1) \text{ for } j \geq 2, \\
P_{j,K_n} &= \psi_j \prod_{\ell=j+1}^{K_n} (1-\psi_\ell) \text{ (where } \psi_1 = 1) \\
Z_n &\sim \begin{cases} \delta_{K_n}(\cdot) & \text{for } n = T_{K_n} \\ \text{Categorical}(P_{j,K_n}) & \text{otherwise} \end{cases} \\
p_j &\overset{\text{iid}}{\sim} \text{Dirichlet}((1/10)\cdot \mathbf{1}), \\
X_n | Z_n &\sim \text{Multinomial}\left(\sum_\ell X_n^\ell, p_{Z_n}\right).
\end{aligned}
\tag{4.1}
$$

The Markov chain was burned in for 40000 iterations [4], with the last 40000 iterations taken for the final calculations. A representative clustering was created by taking a VI estimate based on the last 40000 iterations from the Metropolis–Hastings algorithm. The qualitative performance of the final VI estimate was then assessed.

We also used the following alternative clustering methods to cluster the same Twitter dataset.

1. A DPMM fitted using a collapsed Gibbs sampler, where the chain was burned-in for 5000 iterations, and the last 5000 iterations were used to calculate a VI estimate of the underlying clustering. [5]

2. k-means clustering, where k was chosen via the elbow method. [6] For this method, word vectors were normalized so that the components of each vector

[4] See Figure A.4 for the Metropolis–Hastings convergence diagnostics.
[5] See Figure A.5 for the DPMM convergence diagnostics.
[6] See Figure A.6 for the k-means distortion plot.

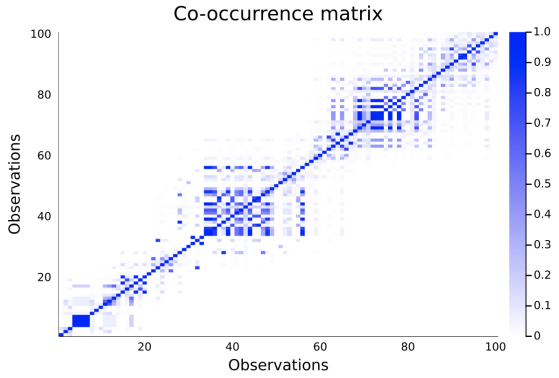


Figure 4.22: Co-occurrence matrix of clusterings sampled from the NTL mixture model for President Joe Biden's Twitter dataset.

sum to 1, to account for the fact that the tweets differ in length.

Point estimates from each of the above two alternative methods were then qualitatively compared to the VI estimate from the NTL mixture model. [7]

4.4.2 Results

Figure 4.22 depicts the co-occurrence matrix of the clusterings sampled from the NTL mixture model posterior for Joe Biden's Twitter dataset. The matrix indicates that the model is quite confident of a block diagonal structure, with two large blocks in the middle of the dataset corresponding to clusters of tweets of moderately large size. Towards the ends of the dataset, the model is less confident in the presence of clusters, with many tweets towards the end being in singleton clusters.

Figure 4.23 depicts a VI estimate of the underlying clustering based on the last 5000 iterations of the Markov chain outputted by the Metropolis–Hastings algorithm. The comments made regarding the empirical co-occurrence matrix in Figure 4.22 also apply for the VI estimate of the underlying clustering — there are two large clusters within the middle of the dataset, with many smaller clusters towards the end.

[7] Complete clusterings given by the NTL mixture model, DPMM, and k-means clusterings can be found at the following link: https://github.com/realseanla/ntl-mixture-model.

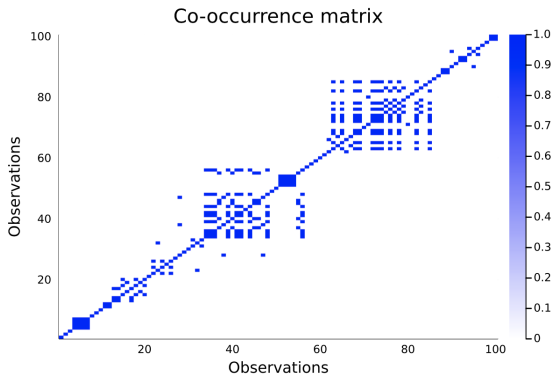


Figure 4.23: Co-occurrence matrix of the VI estimate of the clustering for President Joe Biden's Twitter dataset from the NTL mixture model.

Cluster	Tweet
1	Two years ago, a white supremacist entered Pittsburgh's Tree of Life Synagogue and perpetrated the deadliest anti-Semitic attack in American history. May the memories of those we lost be a blessing — and may we never stop fighting the scourges of anti-Semitism and gun violence.
1	6 days. Return your ballot now: https://t.co/eoxT07d7QB
34	Christen — tell your grandmother I'm incredibly grateful to have her support, and thank you for helping her cast her ballot.
34	The future of our country is on the ballot — and you get to decide what it looks like. Vote: https://t.co/eoxT07d7QB
34	Enough of the lies. Vote him out: https://t.co/eoxT07uIl9 https://t.co/iTdiPVy8FA
63	This is your chance to be a part of history. Vote: https://t.co/eoxT07uIl9 https://t.co/4bH8iawiSE
63	The future of this country is in your hands. Make a plan to vote now. https://t.co/uoiVh9ZqzI
63	The urgency of this election couldn't be greater — and the stakes couldn't be higher. Don't wait: go to https://t.co/eoxT07d7QB and vote early today.
80	I want to extend my prayers and condolences to the Chaldean Assyrian Community this 10th anniversary of Our Lady of Deliverance church massacre in Baghdad. The right to worship is fundamental, and as Americans we should be proud that people from around the world find a home here.
80	You're absolutely right, Brayden — no one should ever underestimate the American people. https://t.co/aCsdlWmF62

Table 4.4: Tweets from various clusters in the VI estimate of the clustering from the NTL mixture model of President Joe Biden's tweets.

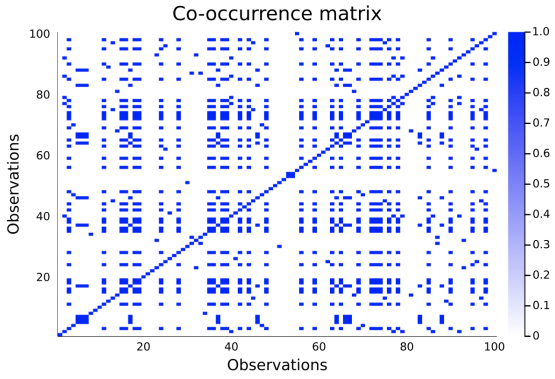
Table 4.4 depicts tweets from various clusters in the VI estimate. Qualitatively, many of the clusters within the VI estimate of the underlying clustering of President Joe Biden's tweets seem to capture the same topic.

The first three tweets from the two largest clusters, clusters 34 and 63, pertain to the president urging American citizens to vote in the 2020 US election. Although it is promising that the tweets within these two clusters seem to pertain to the same topics, these two clusters could also be meaningfully merged into the same cluster, so the fact that these two clusters are considered to be different is an indication of the limitation of this model to recognize the same cluster across large stretches of time.

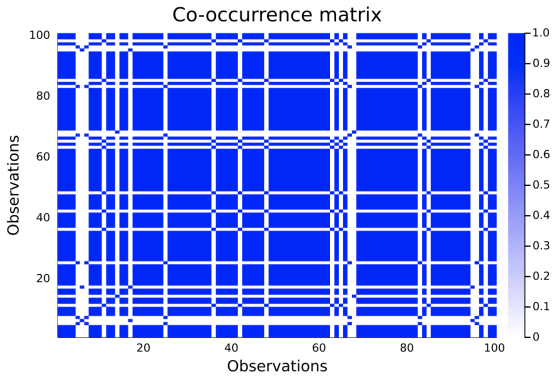
For smaller clusters, the model seems to have difficulty grouping tweets together with the same topic structure. For example, cluster 1 contains two tweets which discuss different topics – the first tweet discusses an incident of racial violence, and the second tweet discusses voting. Cluster 80 also seems to contain unrelated tweets, with the two tweets in this cluster discussing racial violence for the former, and a response to a supporter of Joe Biden for the latter.

Overall, the lack of clarity in the topic structure for smaller tweets may be an indication that the structure of this Twitter dataset differs in an important way from what is assumed in the parameterization of the NTL mixture model that is applied. One possible improvement to the model is to modify the scale parameter of the Dirichlet prior to be a quantity smaller than the given scale of $(1/10) \cdot \mathbf{1}$. The scale factor controls the cost of grouping together tweets with different sets of words together, with smaller values of the scale increasing the penalty for grouping dissimilar tweets. A more extensive modification could be to assume that each cluster is assigned a distribution over multiple topics, instead of the current assumption of a single topic being assigned to each cluster. This may give the model more leeway in grouping together tweets of similar topics, though choosing hyperparameters for probabilistic topic models often requires extensive tuning [42].

Figure 4.24 depict the co-occurrence matrices of clusterings given by a VI estimate of the DPMM and k-means clustering. The co-occurrence matrices do not appear to capture any temporal structure of the tweets, as the methods assume exchangeability of the data.



(a) DPMM VI estimate



(b) k-means estimate; $k = 5$

Figure 4.24: Co-occurrence matrices of clusterings given by (a) VI estimate from DPMM, and (b) k-means clustering with $k = 5$.

Cluster	Tweet
1	Two years ago, a white supremacist entered Pittsburgh's Tree of Life Synagogue and perpetrated the deadliest anti-Semitic attack in American history. May the memories of those we lost be a blessing — and may we never stop fighting the scourges of anti-Semitism and gun violence.
4	We are the United States of America. We can beat this virus. There is no challenge we cannot meet, no enemy we cannot face, no threat we cannot conquer when we stand together.
4	There's no challenge we can't overcome when we stand united. With just four days to go, tune in as we get out the vote in Iowa. https://t.co/0O9S2J9Tw6

Table 4.5: Tweets from various clusters in the VI estimate of the underlying clustering from the DPMM of President Joe Biden's tweets.

Table 4.5 depicts clusters from various clusters from the VI estimate for the DPMM. Despite not capturing any temporal patterns, some tweets are grouped into more meaningful clusters in the VI estimate of the DPMM in comparison to the NTL mixture model. Cluster 1 contains the first Tweet in the dataset, and it is the only Tweet in that cluster from the DPMM estimate. In contrast, the NTL mixture model has this tweet grouped together with a tweet that discusses a different topic. However, not every cluster in the VI estimate from the DPMM seems to describe a coherent topic structure. Cluster 4 contains two tweets which describe two different topics, one discussing COVID-19, and the other discussing voting.

Cluster	Tweet
4	Let's put dogs back in the White House. https://t.co/7pBihksfXT
4	The issues we're facing are far bigger than any political party. It's why I'll be a president for all Americans — Democrats, Republicans, and Independents alike — because I believe we must work together if we're going to get anything done.
4	.@BarackObama and I have seen the office of the presidency up close, we know what the job entails, and there's too much at stake to give Donald Trump another four years. Vote: https://t.co/eoxT07uII9 https://t.co/7imuWqlZSN

Table 4.6: Tweets from various clusters in the VI estimate of the underlying clustering from the k-means clustering of President Joe Biden's tweets.

k-means clustering also appears to have trouble grouping together tweets with meaningful topic structure. Three tweets in cluster 4 of the clustering given by *k*-means clustering, described in Table 4.6, appear to discuss two different topics – the first tweet discusses dogs, whereas the latter two tweets discuss the American politics and the US election.

Chapter 5

Conclusion

We have introduced the Neutral-to-the-Left mixture model, a family of infinite mixture models that generalize the Dirichlet process mixture model. The NTL mixture model is parameterized by the arrival time distribution of new clusters, and the distribution of stick breaking weights. We consider one parameterization of the NTL mixture model with characteristics that differ from that of the Dirichlet process mixture model. We describe two Metropolis-within-Gibbs algorithms for sampling the posterior distribution of clusterings given data, and validate the correctness of the samplers by constructing accurate estimates of the true posterior probabilities of all clusterings for small data sets. We evaluate the efficacy of the Metropolis-within-Gibbs algorithms on data simulated from the NTL mixture model prior, and find that it more adequately recovers the data-generating clustering in comparison to clusterings given by Dirichlet process mixture models and k-means clustering. Finally, we apply a parameterization of the NTL mixture model to cluster tweets.

5.1 Future work

There are various directions for future research in the parameterizations and derivations of the NTL mixture model.

In similar vein to the multitude of applications of the Dirichlet process, one possible direction for future is to adapt the NTL mixture model to more compli-

cated clustering-style models. During the work that led to the creation of this thesis, preliminary research was performed regarding the following adaptations of the NTL mixture model:

1. The *Neutral-to-the-Left Infinite Hidden Markov Model*, where the NTL stick breaking structure models the arrivals and evolution of hidden states within a hidden Markov model with infinitely many states. This early research was inspired by the application of the hierarchical Dirichlet process to modelling hidden Markov models [19].

2. The *Neutral-to-the-Left Multiple Changepoint Model*, where the arrival distribution of new changepoints in a time series is an explicit parameter of the model.

In future work, we would like to further explore the utility of these models, and other possible derivations of the NTL mixture model.

Bibliography

[1] Benjamin Bloem-Reddy and Peter Orbanz. Preferential Attachment and Vertex Arrival Times. arXiv:1710.02159 [math.PR], 2017. → pages 3, 11, 14

[2] Benjamin Bloem-Reddy, Adam Foster, Emile Mathieu, and Yee Whye Teh. Sampling and Inference for Beta Neutral-to-the-Left Models of Sparse Networks. In *Uncertainty in Artificial Intelligence*, volume 34. AUAI Press, 2018. → pages 3, 11, 12, 15, 18

[3] Brenda Betancourt, Giacomo Zanella, Jeffrey W Miller, Hanna Wallach, Abbas Zaidi, and Rebecca C. Steorts. Flexible Models for Microclustering with Application to Entity Resolution. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. → page 3

[4] Brenda Betancourt, Giacomo Zanella, and Rebecca C. Steorts. Random partition models for microclustering tasks. *Journal of the American Statistical Association*, 0(0):1–13, 2020. → pages 3, 16, 17

[5] J.E. Griffin and M.F.J. Steel. Stick-breaking autoregressive processes. *Journal of Econometrics*, 162(2):383–396, 2011. ISSN 0304-4076. → page 3

[6] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*. Springer, 2010. → pages 5, 8

[7] Christopher K. I Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. The MIT Press, 2019. ISBN 9780262256834. → page 5

[8] Thomas L. Griffiths and Zoubin Ghahramani. The Indian Buffet Process: An Introduction and Review. *Journal of Machine Learning Research*, 12 (32):1185–1224, 2011. → page 5

[9] Yee Whye Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010. → pages 6, 7

[10] Bernt Oksendal. *Stochastic Differential Equations (3rd Ed.): An Introduction with Applications*. Springer-Verlag, Berlin, Heidelberg, 1992. ISBN 3387533354. → page 6

[11] Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994. ISSN 1017-0405. → page 6

[12] Jeffrey W. Miller. An elementary derivation of the Chinese restaurant process from Sethuraman's stick-breaking process. *Statistics & Probability Letters*, 146:112–117, 2019. ISSN 0167-7152. → page 7

[13] Jim Pitman and Jean Picard. *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII - 2002*, volume 1875 of *Lecture notes in mathematics*. Springer, Berlin/Heidelberg, 2006. ISBN 354030990X. → page 7

[14] David Blackwell and James B MacQueen. Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, 1(2):353–355, 1973. ISSN 0090-5364. → page 7

[15] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004. ISSN 0885-6125. → page 8

[16] Paolo Giordani. *An Introduction to Clustering with R*. Behaviormetrics: Quantitative Approaches to Human Behavior, 1. Springer Singapore : Imprint: Springer, 1st ed. 2020. edition, 2020. ISBN 981-13-0552-8. → pages 8, 41

[17] Kevin Murphy. Conjugate Bayesian analysis of the Gaussian distribution. The University of British Columbia, 2007. → pages 8, 28

[18] Subhashis Ghosal and Aad van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017. → page 9

[19] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. → pages 9, 74

[20] Stanley I. M. Ko, Terence T. L. Chong, and Pulak Ghosh. Dirichlet Process Hidden Markov Multiple Change-point Model. *Bayesian Analysis*, 10(2): 275 – 296, 2015. → page 10

[21] Peter Orbanz and Daniel M. Roy. Bayesian Models of Graphs, Arrays and Other Exchangeable Random Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):437–461, 2015. → pages 10, 11

[22] Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-exchangeable graphs and sparsity. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. → page 11

[23] Pierpaolo De Blasi, Stefano Favaro, Antonio Lijoi, Ramses H. Mena, Igor Prunster, and Matteo Ruggiero. Are Gibbs-Type Priors the Most Natural Generalization of the Dirichlet Process? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):212–229, Feb 2015. ISSN 2160-9292. → page 18

[24] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Texts in statistical science. Chapman & Hall/CRC, 2nd ed. edition, 2004. ISBN 158488388X. → page 24

[25] Gareth O Roberts and Jeffrey S Rosenthal. General state space Markov chains and MCMC algorithms. *Probability surveys*, 1, 2004. ISSN 1549-5787. → page 27

[26] Charles Elkan. Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 289–296. ACM, 2006. ISBN 9781595933836. → page 29

[27] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. → page 31

[28] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. → page 31

[29] Creagh Briercliffe. Poisson Process Infinite Relational Model: a Bayesian nonparametric model for transactional data. MSc thesis, The University of British Columbia, 2016. → page 32

[30] Saptarshi Chakraborty, Suman K. Bhattacharya, and Kshitij Khare. Estimating accuracy of the MCMC variance estimator: a central limit theorem for batch means estimators. arXiv:1911.00915 [stat.CO], 2019. → page 32

[31] David I Hastie, Silvia Liverani, and Sylvia Richardson. Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations. *Statistics and computing*, 25 (5):1023–1037, 2015. ISSN 0960-3174. → page 35

[32] Gordon J. Ross and Dean Markwick. *dirichletprocess: Build Dirichlet Process Objects for Bayesian Modelling*, 2020. R package version 0.4.0. → page 35

[33] William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. → page 38

[34] Christian P Robert. *Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer New York, New York, 2007. ISBN 9780387715988. → pages 38, 39

[35] Sara Wade and Zoubin Ghahramani. Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). *Bayesian Analysis*, 13(2), Jun 2018. ISSN 1936-0975. → page 39

[36] David A. Binder. Bayesian cluster analysis. *Biometrika*, 65(1):31–38, 1978. ISSN 0006-3444. → page 39

[37] Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. ISSN 0047-259X. → page 39

[38] Sara Wade. *mcclust.ext: Point estimation and credible balls for Bayesian cluster analysis*, 2015. R package version 1.0. → page 40

[39] Arno Fritsch. *mcclust: Process an MCMC Sample of Clusterings*, 2012. R package version 1.0. → page 40

[40] David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models: A focus on graphical model design and applications to document and image analysis. *IEEE signal processing magazine*, 27(6):55–65, 2010. ISSN 1053-5888. → page 62

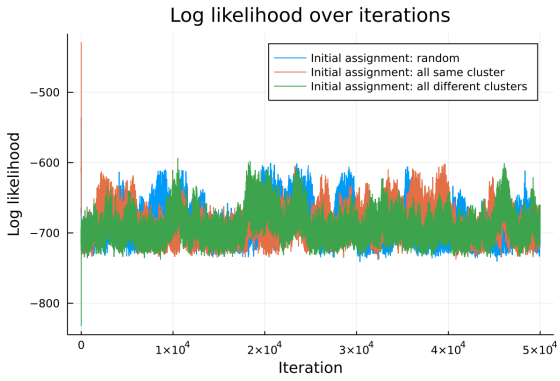
[41] Rohan Vopani. Tweets of Joe Biden's official Twitter handle @JoeBiden. Kaggle, 2020. → page 62

[42] Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why Priors Matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams,

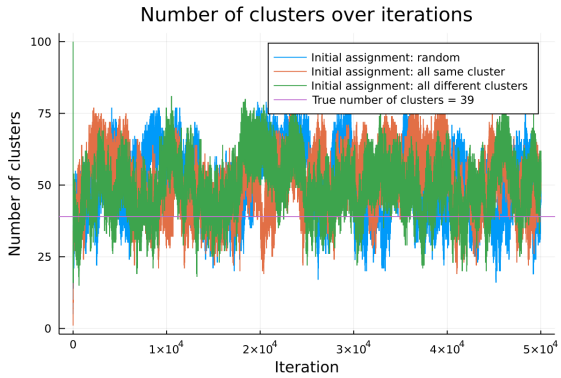
and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. → page 67

Appendix A

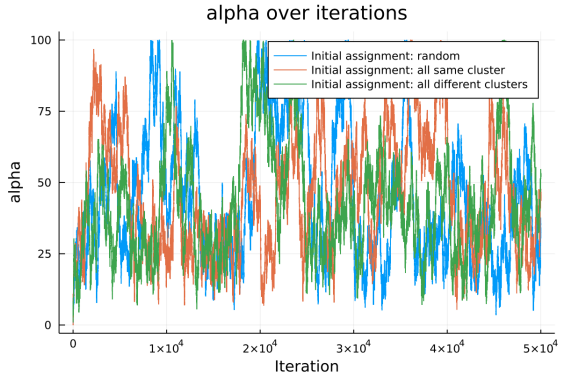
Supporting Materials



(a) Log likelihood over iterations

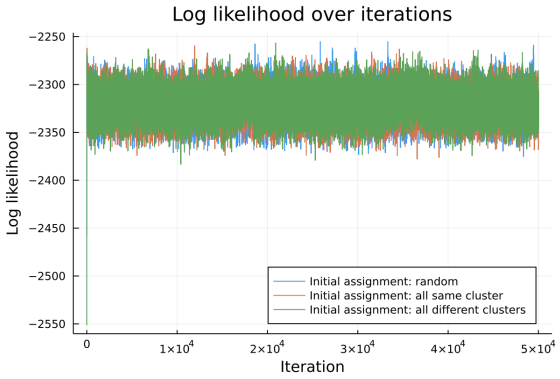


(b) Number of clusters over iterations

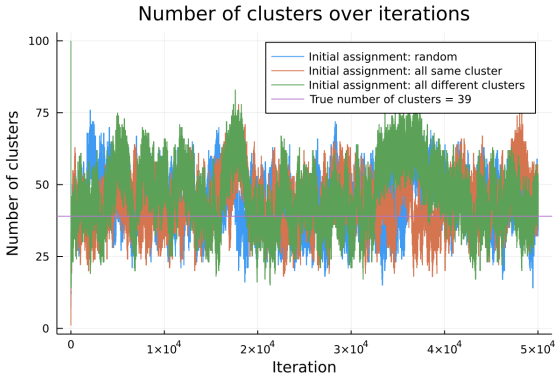


(c) α parameter posterior over iterations

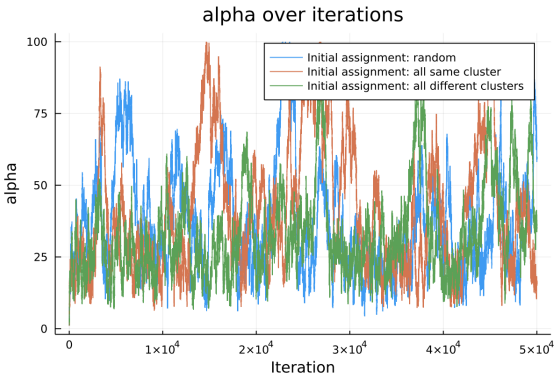
Figure A.1: Convergence diagnostic plots for the DPMM on multivariate Gaussian data.



(a) Log likelihood over iterations

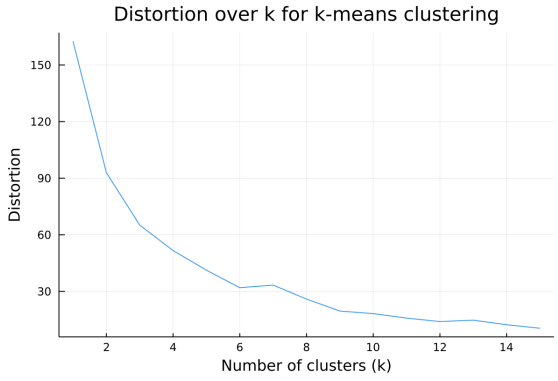


(b) Number of clusters over iterations

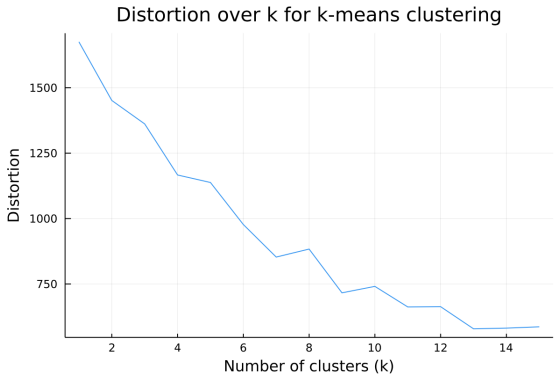


(c) α parameter posterior over iterations

Figure A.2: Convergence diagnostic plots for the DPMM on multinomial data.

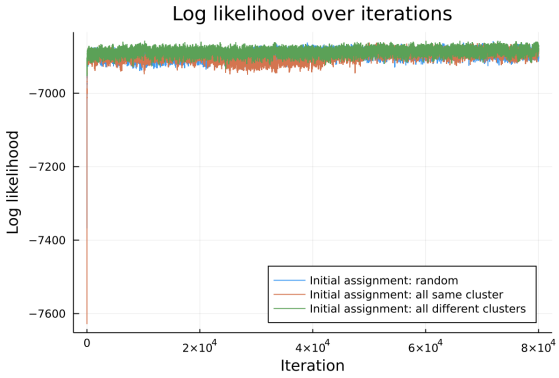


(a) Multivariate Gaussian data

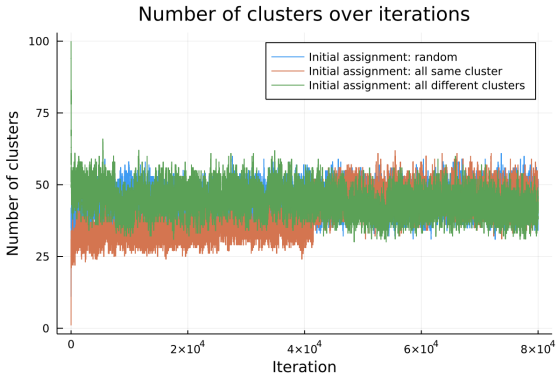


(b) Multinomial data

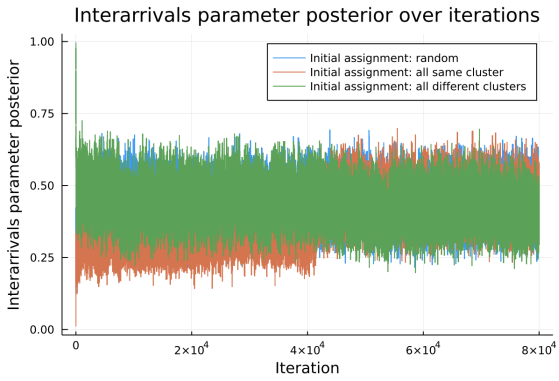
Figure A.3: Distortion plots for *k*-means clustering on (a) multivariate Gaussian data, and (b) multinomial data.



(a) Log likelihood over iterations

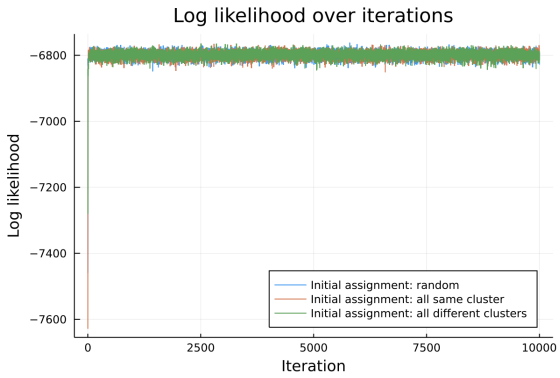


(b) Number of clusters over iterations

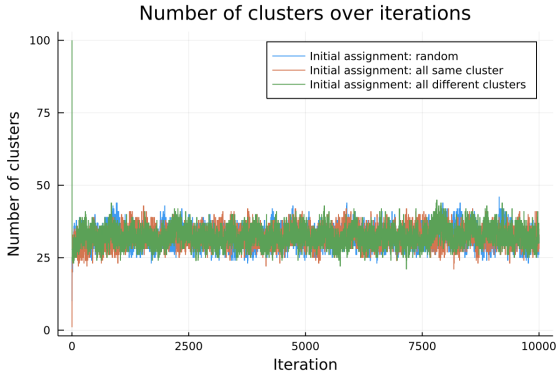


(c) Interarrivals parameter posterior over iterations

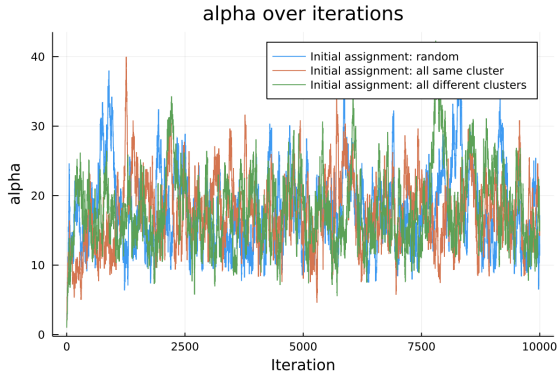
Figure A.4: Convergence diagnostic plots for the Metropolis-Hastings sampler on President Joe Biden's Twitter dataset.



(a) Log likelihood over iterations



(b) Number of clusters over iterations



(c) α parameter posterior over iterations

Figure A.5: Convergence diagnostics for DPMM on President Joe Biden's Twitter dataset.

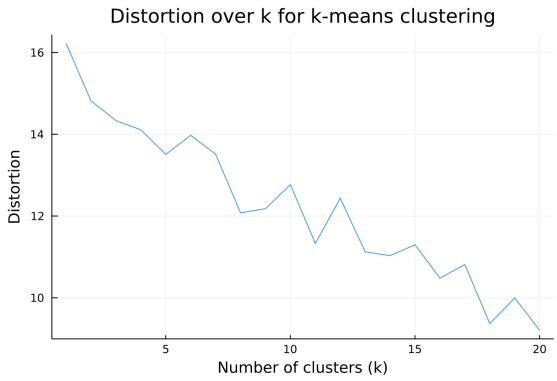


Figure A.6: Distortion plot for *k*-means clustering on President Joe Biden's Twitter dataset.