CARNEGIE MELLON UNIVERSITY

HYPER MARKOV NON-PARAMETRIC PROCESSES FOR MIXTURE MODELING AND MODEL SELECTION

A DISSERTATION SUBMITTED TO THE GRADUATE SCHOOL IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY
In
STATISTICS

by

DANIEL HEINZ

Department of Statistics
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

June, 2010

Abstract

Markov distributions describe multivariate data with conditional independence structures. Dawid and Lauritzen (1993) extended this idea to hyper Markov laws for prior distributions. A hyper Markov law is a distribution over Markov distributions whose marginals satisfy the same conditional independence constraints. These laws have been used for Gaussian mixtures (Escobar, 1994; Escobar and West, 1995) and contingency tables (Liu and Massam, 2006; Dobra and Massam, 2009).

In this paper, we develop a family of non-parametric hyper Markov laws that we call *hyper Dirichlet processes*, combining the ideas of hyper Markov laws and non-parametric processes. Hyper Dirichlet processes are joint laws with Dirichlet process laws for particular marginals. We also c describe a more general class of Dirichlet processes that are not hyper Markov, but still contain useful properties for describing graphical data. The *graphical Dirichlet processes* are simple Dirichlet processes with a hyper Markov base measure. This class allows an extremely straight-forward application of existing Dirichlet knowledge and technology to graphical settings.

Given the wide-spread use of Dirichlet processes, there are many applications of this framework waiting to be explored. One broad class of applications, known as Dirichlet process mixtures, has been used for constructing mixture densities such that the underlying number of components may be determined by the data (Lo, 1984; Escobar, 1994; Escobar and West, 1995). I consider the use of the new graphical Dirichlet process in this setting, which imparts a conditional independence structure inside each component. In other words, given the component or cluster membership, the data exhibit the desired independence structure.

We discuss two applications. Expanding on the work of Escobar and West (1995), we estimate a non-parametric mixture of Markov Gaussians using a Gibbs sampler. Secondly, we employ the Mode-Oriented Stochastic Search of Dobra and Massam (2009) for determining a suitable conditional independence model, focusing on contingency tables. In general, the mixing induced by a Dirichlet process does not drastically increase the complexity beyond that of a simpler Bayesian hierarchical models sans mixture components. We provide a specific representation for decomposable graphs with useful algorithms for local updates.

Acknowledgements

I would very much like to express my gratitude toward my thesis committee. I am especially grateful to Stephen Fienberg, who chaired the committee, for his many insights and research leads. I thank Hélène Massam for many helpful comments are for carefully proofing this dissertation. I also appreciate the useful remarks and extensions provided by Alessandro Rinaldo, Chad Schafer, and Cosma Shalizi.

I am extremely grateful to my parents for both genetic and environmental factors. Thank you for encouraging my interest in mathematics, and for teaching me the value of doing things well.

Finally, my most humble gratitude to my wife, Michele, for her patience, love, and encouragement. Thank you especially for your understanding during the long days and nights of this writing, and for helping me see this work to completion.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Graphs and Markov Random Fields

Markov distributions are multivariate measures that satisfy a specified set of conditional independence relations, often represented by an undirected graph. A measure is Markov with respect to an undirected graph if, whenever two variables do not have an edge between them, they are conditionally independent given the remaining variables. Markov distributions, or *Markov random fields*, have been used for a wide variety of problems, including demography (Sebastiani, 2003), flood prediction (Allcroft and Glasbey, 2003), and telecommunications (Zachary and Ziedins, 1999).

In telecommunications, Zachary and Ziedins (1999) analyzed loss networks using Markov distributions. Their model described a communications system with multiple resources, each having a limited capacity. The system handles different types of calls, each requiring

a specific subset of the resources. If the system's resources are sufficient, then the call is put through. If the system's resources are insufficient, then the call is lost. The stationary distribution for the amount of unused resources is interesting because it directly impacts the percentage of lost calls. Zachary and Ziedins suggested a graphical model with the various resources represented by nodes. In their model, two resources were considered neighbors if at least one type of call required both of them. They then approximated the stationary distribution with a Markov distribution on this graph.

This example shows the practical importance of Markov models. Zachary and Ziedins (1999) state that the normalizing constant for the stationary distribution is too complex for any networks of realistic size. This is essentially due to the necessity of summing over all possible combinations of quantities of the various call types which do not exceed network capacity. When employing a Markov random field, this sum can be broken into more manageable summations over smaller subsets.

1.2 Markovity for Prior Distributions

Dawid and Lauritzen (1993) extended the notion of Markov distributions for variables to *hyper Markov* distributions for parameters. In Bayesian statistics, one considers a random distribution, which therefore has its own distribution called the prior. A prior law over Markov measures is *hyper Markov* if the random marginal measures also satisfy the conditional independence structure. This is equivalent to requiring that the distribution of each variable is conditionally independent of the joint distribution of the other variables given

the joint distribution of its neighbors. This leads to some useful properties as we discuss in Chapter 2.

For parametric models, we can specify hyper Markov laws by placing certain constraints on the prior laws for the parameters. For example, we consider the hyper Normal prior law presented by Dawid and Lauritzen (1993). Suppose we observe data from a location family with an unknown mean. We may decide to model the mean using a Normal prior law. The hyper Markov version of this law equates to setting certain elements of the precision matrix to be zero.

1.3 Non-Parametric Hyper Markov Priors

Graphical models are especially important in high-dimensional problems for which dimension reduction greatly reduces computational burden. Bayesian non-parametric approaches are useful in these same situations when there is more uncertainty about the underlying structure of a distribution. Since both of these approaches may be desirable in similar situations, it is the goal of this thesis to explore how these concepts may be united. We shall call a model non-parametric if it can closely approximate any arbitrary distribution.

Specifically, we focus on the class of Dirichlet processes first described by Ferguson (1973). The Dirichlet process is one of the best-known and most widely applied non-parametric priors. One possible detraction of a Dirichlet process is that it generates an almost-surely discrete distribution; however Antoniak (1974) introduced the concept of a *mixture* of Dirichlet processes which uses this feature advantageously and Lo (1984)

showed how these mixtures could be used for density estimation. The result is a Bayesian method for specifying mixture distributions in which the number of mixing components can be determined by the data and easily accommodates further observations. An example application is provided by Escobar and West (1995) who develop a Gibbs sampling scheme for a Dirichlet Mixture of univariate Gaussians.

Following the work of Dawid and Lauritzen (1993) we can specify hyper Dirichlet process as the joint law of certain marginal Dirichlet processes. The major theoretical contribution are the necessary and sufficient conditions for a hyper Dirichlet process to be a Dirichlet process. This allows straightforward application of existing theory and technology surrounding Dirichlet processes. We also present a useful class of Dirichlet processes, called *graphical Dirichlet processes* that we can obtain by relaxing some of the aforementioned conditions. These processes are not hyper Markov, but still induce useful graphical properties on the distribution of observable data.

Graphical Dirichlet processes are convenient because they are a special case of the well-known Dirichlet process. This allows straightforward extension of Dirichlet processes to incorporate conditional independence constraints. As an example, we explore the use of graphical Dirichlet processes for density estimation. The result is a mixture model of an unknown number of components. Within each component, the desired conditional independence constraints are satisfied. Finally, we present a method for comparing various graphical Dirichlet process models.

In the second chapter I explain notation and definitions as well as review some well-known and lesser-known properties of graphical models. The third chapter covers the topic of non-parametric priors. I discuss therein various representations, properties, and generalizations of the Dirichlet Process and other non-parametric processes. I also provide a formal proof for a property of the Dirichlet distribution that will be useful for proving independence properties of hyper and graphical Dirichlet processes. Together, chapters two and three represent the majority of the necessary background to understand the theory behind and applications of the hyper and graphical Dirichlet process. In some cases, additional background is covered in specific chapters for emphasis or clarity. The fourth chapter introduces the hyper Dirichlet process and fundamental theory, including how we can specify a hyper Dirichlet process and its relationship to the classic Dirichlet process. In chapter five, I introduce a C++ class I developed to maintain a useful representation of a graph through local updates. This representation is a fundamental component for the hyper Dirichlet process applications I present in chapters six and seven. Chapter six extends the notion of a mixture of Dirichlet processes, by exploring the application of hyper Dirichlet processes in this framework and the resulting independence relationships. The seventh chapter provides results of stochastic searches for both simulated and real contingency table data. Finally, the eighth chapter provides some insights about possible extensions and improvements to the framework I have developed.

Chapter 2

Graph Notation and Theory

As a pre-requisite for understanding the material in this chapter and beyond, we establish notation and introduce some preliminary results in graph theory.

2.1 Definitions and Notation

Throughout this paper we consider a multivariate random variable $X = (X_v)$ where v ranges over some index set, $\mathbf{V}$ of dimension $|\mathbf{V}|$. We denote the range of X by $\mathcal{X} = (\times_{v\in\mathbf{V}}\mathcal{X}_v)$ which has some associated σ-field, $\mathcal{F} = (\times_{v\in\mathbf{V}})$. Marginal values of these values on some subset $\mathbf{A}$ of $\mathbf{V}$ are denoted by subscripts:

$$\begin{aligned} X_{\mathbf{A}} &= (X_{\mathbf{A}} : v \in \mathbf{A}) \\ \mathcal{X}_{\mathbf{A}} &= \times_{v\in\mathbf{A}}\mathcal{X}_v \\ \mathcal{F}_{\mathbf{A}} &= \times_{v\in\mathbf{A}}\mathcal{F}_v \end{aligned}$$

If α is a measure over some space $(\mathcal{X}, \mathcal{F})$, then $\overline{\alpha} = \alpha/\alpha(\mathcal{X})$ is the probability measure proportional to α. If $\mathbf{A} \subseteq \mathbf{V}$, then $\alpha_{\mathbf{A}}$ is the marginal of α over $\mathcal{X}_{\mathbf{A}}$. Thus, $\alpha_{\mathbf{A}}(U) = \alpha(U \times \mathcal{X}_{\mathbf{V}\setminus\mathbf{A}}),\ \forall\ U \in \mathcal{F}_{\mathbf{A}}$. If $\mathbf{B}$ is another (possibly improper) subset of $\mathbf{V}$, then $\alpha_{\mathbf{B}|\mathbf{A}}$ is the collection of marginal distributions of $X_{\mathbf{B}}$ given $X_{\mathbf{A}} = x_{\mathbf{A}}$ induced by α.

If α and β are both measures on some space $(\mathcal{X}, \mathcal{F})$, then we define their sum, $\alpha + \beta$, by

$$[\alpha + \beta](U) = \alpha(U) + \beta(U), \quad \forall\ U \in \mathcal{F}. \tag{2.1}$$

For $x \in \mathcal{X}$, we define δ_x as a point mass concentrated at x.

$$\delta_x(U) = \begin{cases} 1, & x \in U \\ 0, & x \notin U \end{cases}, \qquad \forall U \in \mathcal{F}. \tag{2.2}$$

A graph, $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, is defined by a vertex set (or node set) $\mathbf{V}$ and an edge set $\mathbf{E}$. There is an edge from one vertex, v_1, to another vertex, v_2, if $(v_1, v_2) \in \mathbf{E}$. Unless indicated otherwise, we restrict our attention to *undirected* graphs, which means that $(v_i, v_j) \in \mathbf{E}$ if and only if $(v_j, v_i) \in \mathbf{E}$. We shall therefore take the statement $(v_i, v_j) \in \mathbf{E}$ to imply $(v_j, v_i) \in \mathbf{E}$. In this case, we shall say that v_i and v_j are neighbors and write $v_i \sim v_j$. By convention, we assume that $(v, v) \in \mathbf{E}$ for all $v \in \mathbf{V}$. We call such edges *loops*. For our purposes, there is no practical difference if loops are excluded from $\mathbf{E}$, though some minor changes are required for certain definitions. If $\mathbf{A} \subseteq \mathbf{V}$, then $\mathcal{G}_{\mathbf{A}}$ is a *subgraph* of $\mathcal{G}$ over $\mathbf{A}$, which has vertex set $\mathbf{A}$ and edge set $\mathbf{E}_{\mathbf{A}} = (\mathbf{A} \times \mathbf{A}) \cap \mathbf{E}$. In other words, the subgraph $\mathcal{G}_{\mathbf{A}}$ is obtained from $\mathcal{G}$ by removing all vertices in $\mathbf{V} \setminus \mathbf{A}$ and edges which have at least one endpoint outside of $\mathbf{A}$. We say that $\mathbf{A}$ induces the subgraph $\mathcal{G}_{\mathbf{A}}$.

A path from v_i to v_j is a sequence of nodes starting with v_i and ending with v_j such that nodes that are adjacent in the sequence are neighbors in the graph. We further stipulate that with the possible exception of $v_i = v_j$, all nodes in the path are unique. If there is a path from v_i to v_j, then we say that they are *connected*. A *connected component* is a set of pair-wise connected nodes that are not connected to any nodes outside the component. If only one such connected component exists, we call $\mathcal{G}$ a *connected graph*. A path that starts and ends with the same node is called a *cycle*. A *triangulated* or *chordal* graph is a graph with the property that every cycle of length four or longer contains a *chord*, that is, two non-consecutive nodes which are neighbors. If the graph contains no cycles at all, then it is a *tree*. An immediate consequence of this definition is that all paths in a tree are unique. (This may be seen by contradiction. Choose v_i and v_j such that two different paths exist from v_i to v_j, say p_1 and p_2. Then the sequence formed by concatenating p_1 forwards and p_2 in reverse is a cycle.)

A set $\mathbf{S}$ is said to *separate* v_i and v_j if every path between them contains an element of $\mathbf{S}$. By extension, $\mathbf{S}$ separates $\mathbf{A}$ and $\mathbf{B}$ if it separates v_a and v_b whenever $v_a \in \mathbf{A}$ and $v_b \in \mathbf{B}$.

The graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a *complete* graph if $\mathbf{E} = \mathbf{V} \times \mathbf{V}$. Similarly, a subset $\mathbf{A} \subseteq \mathbf{V}$ is called complete if it induces a complete subgraph. We reserve the term *clique* only for complete subsets that are maximal with respect to inclusion (whereas some refer to any complete subset as a clique and use the term maximal clique if no proper superset is also complete.) A *decomposition* of $\mathcal{G}$ is a pair of sets $\mathbf{A}$,$\mathbf{B} \subseteq \mathbf{V}$ such that $\mathbf{A} \cup \mathbf{B} = \mathbf{V}$, $\mathbf{A} \cap \mathbf{B}$ is complete and $\mathbf{A} \cap \mathbf{B}$ separates $\mathbf{A}$ and $\mathbf{B}$. The decomposition is *proper* if both $\mathbf{A}$ and $\mathbf{B}$

are proper subsets of $\mathbf{V}$. Finally, we say that $\mathcal{G}$ is *decomposable* if it is complete or there exists a proper decomposition $(\mathbf{A}, \mathbf{B})$ such that $\mathcal{G}_{\mathbf{A}}$ and $\mathcal{G}_{\mathbf{B}}$ are decomposable. Thus, a decomposable graph may be continuously decomposed into its cliques. Perhaps the most well-known result on graph theory is that a graph is decomposable if and only if it is triangulated (see Lemma 2.1.3).

For simplicity, we focus almost exclusively on decomposable graphs, though we discuss more general applications from time to time. A decomposable graph admits a *perfect ordering* of its cliques.

Definition 2.1.1 (Perfect Ordering)**.** *Suppose a graph $\mathcal{G}$ has n cliques. Let the cliques have an arbitrary ordering $\mathbf{C}_1, \ldots, \mathbf{C}_k$. Define $\mathbf{H}_i = \cup_{j=1}^{i} \mathbf{C}_j$. For $i \geq 2$ define $\mathbf{S}_i = \mathbf{C}_i \cap \mathbf{H}_{i-1}$ and $\mathbf{R}_i = \mathbf{C}_i \setminus \mathbf{H}_{i-1}$. The ordering of the cliques is a perfect ordering if for each $2 \leq i \leq n$, there exists $j < i$ such that $\mathbf{S}_i \subset \mathbf{C}_j$.*

The sets $\mathbf{H}_k$ are called the histories. The separators, $\mathbf{S}_k$, separate $\mathbf{C}_k$ from the previous history. The sets $\mathbf{R}_k$ are called the residuals, which represent the new nodes being added to the history. In a perfect ordering, each separator is complete. In general, we will allow some or all separators to be empty, which occurs if and only if $\mathcal{G}$ is disconnected. Note that the perfect ordering is not unique. In fact, each clique can be taken to be $\mathbf{C}_1$ in some perfect ordering.

Closely related to the concept of a perfect ordering is the definition of a simplicial set or node. A node or set of nodes is *simplicial* if its boundary is complete. The boundary of a node is the set of all neigbors of v excluding v itself. That is $bd(v) = \{v_j : v_j \sim v, v_j \neq v\}$.

The boundary of a set is $bd(\mathbf{A}) = \cup_{v \in \mathbf{A}} bd(v) \setminus \mathbf{A}$. A perfect ordering of cliques is one such that for $i \geq 2$, $\mathbf{C}_i$ is simplicial in $\mathcal{G}_{\mathbf{H}_{i-1}}$. The related idea of a *perfect ordering of nodes* is an ordering $v_1, \ldots, v_{|\mathbf{V}|}$ such that for $i \geq 2$, v_i is simplicial in $\mathcal{G}_{\cup_{j<i} v_j}$.

A consequence of a perfect ordering is that decomposable graphs can be represented by a *junction tree*.

Definition 2.1.2 (Junction Tree)**.** *Let $\mathcal{G}$ be a decomposable graph with clique set $\mathscr{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$. A junction tree representation of $\mathcal{G}$ is a tree, $\mathcal{T}$, with node set $\mathscr{C}$ such that for any $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathscr{C}$, if $\mathbf{C}$ is on the unique path from $\mathbf{A}$ to $\mathbf{B}$, then $\mathbf{A} \cap \mathbf{B} \subseteq \mathbf{C}$.*

Note that we may take the subset property (also known as the *junction tree property*) to be proper because otherwise $\mathbf{C} \subset \mathbf{A}$ and is not a clique. If $\mathbf{C}_1, \ldots, \mathbf{C}_k$ is a perfect ordering, then the junction tree may be constructed using the edge set $\mathbf{E} = \{(\mathbf{C}_i, \mathbf{C}_j) : i = 2, \ldots, k\}$, where $j < i$, and $\mathbf{S}_i \subset \mathbf{C}_j$.

When dealing with a disconnected graph, some authors choose to represent it as a *junction forest*, which is a collection of junction trees for each connected component. While the extended analogy is cute, we point out that it is unnecessary. We can simply connect the individual trees with arbitrary edges. By definition, $\mathbf{A} \cap \mathbf{B}$ is empty when the sets are in different connected components so it is trivially the case that these additional edges will not break the junction tree property. As we see in Chapter 5, a convenient method will be to include an additional $\mathbf{C}_0 = \emptyset$ and add an edge between it and each connected component.

2.1.1 Fundamental Theorems about Decomposable Graphs

In Chapter 7, we will utilize a stochastic search for finding a decomposable graphical model. We will need an algorithm for proposing moves between graphical models that is capable of exploring the entire space of decomposable graphs. Toward that end, we present three lemmae which have been proven previously about this class of graphs. The first lemma guarantees that the set of decomposable graphs can be traversed by toggling one edge at a time. That is, it is enough to propose moves between graphs that differ by exactly one edge. The second (Frydenberg and Lauritzen, 1989) and third (Giudici and Green, 1999) results provide the method for testing whether or not a proposed move would result in a decomposable graph. For completeness, we first present equivalent definitions of decomposability that we will use to prove the trio of lemmae and various other results in this paper.

Lemma 2.1.3. *The following are equivalent:*

(i) $\mathcal{G}$ *is decomposable.*

(ii) $\mathcal{G}$ *is triangulated.*

(iii) The nodes of $\mathcal{G}$ *have a perfect ordering.*

(iv) The cliques of $\mathcal{G}$ *have a perfect ordering.*

Proof. (i $\Rightarrow$ *ii).* Suppose $\mathcal{G}$ is decomposable. We prove that it is triangulated by induction on $|\mathbf{V}|$. For $|\mathbf{V}| \leq 3$, all graphs are triangulated. Assume that $|\mathbf{V}| > 3$, and that any smaller decomposable graph is triangulated. Let $v_0, v_1, \ldots, v_n = v_0$ be a chordless cycle

with $n \geq 4$. Since $\mathcal{G}$ is decomposable, there exists a proper decomposition $(\mathbf{A}, \mathbf{B})$ such that $\mathcal{G}_{\mathbf{A}}$ and $\mathcal{G}_{\mathbf{B}}$ are each decomposable. By the inductive hypothesis, either $\mathcal{A}$ or $\mathcal{B}$ contains the cycle, then it must have a chord and we are done. Consequently, assume that $\{v_0, \ldots, v_n\}$ is not a subset of $\mathbf{A}$ or $\mathbf{B}$. Clearly, the choice of v_0 is arbitrary, so choose a vertex in $\mathbf{A} \setminus \mathbf{B}$. Define $i = \min\{k : V_k \in \mathbf{B} \setminus \mathbf{A}\}$ and $j = \max\{k : V_k \in \mathbf{B} \setminus \mathbf{A}\}$. We have $v_{i-1} \sim v_i$ with $v_{i-1} \in \mathbf{A}$ and $v_i \in \mathbf{B}$. By definition of a decomposition, either v_{i-1} or v_i is in $\mathbf{A} \cap \mathbf{B}$. By construction, $v_i \notin \mathbf{A}$, so $v_{i-1} \in \mathbf{A} \cap \mathbf{B}$. A similar argument shows $v_{j+1} \in \mathbf{A} \cap \mathbf{B}$. By our choice of $v_0 = v_n \in \mathbf{A} \setminus \mathbf{B}$, we have that $i - 1 \neq 0$ and $j + 1 \neq n$. Therefore, $0 < i - 1 < i \leq j < j + 1 < n$. By definition, $\mathbf{A} \cap \mathbf{B}$ is complete so $v_{i-1} \sim v_{j+1}$ is a chord.

(ii ⇒ iii). Suppose $\mathcal{G}$ is triangulated. We prove its nodes have a perfect ordering by induction on $|\mathbf{V}|$. For $|\mathbf{V}| = 1$, there is nothing to show. Suppose $|\mathbf{V}| > 1$ and that the implication holds for $|\mathbf{V}| - 1$. By induction, it is enough to find one vertex which is simplicial in $\mathcal{G}$. If $\mathcal{G}$ is disconnected, then a perfect ordering exists for each connected subgraph. Concatenating these yields a perfect ordering for $\mathcal{G}$. Henceforth, assume $\mathcal{G}$ is connected. To establish a contradiction, we suppose that no vertices are simplicial in $\mathcal{G}$ and show that this induces a chordless cycle. Let $\{v_1, \ldots, v_n\}$ be the longest path in $\mathcal{G}$ along distinct vertices. By choosing the longest path, we guarantee that $bd(v_1)$ is contained by this path, whence $\mathcal{G}_{\mathbf{V} \setminus \{v_1\}}$ is connected. Choose u simplicial in $\mathcal{G}_{\mathbf{V} \setminus \{v_1\}}$. By supposition, u is not simplicial in $\mathcal{G}$, so there exists $t \in bd(u)$ such that $t \not\sim v_1$. Since $\{u\} \cup bd(u) \setminus \{v\}$ is complete, the same reasoning implies that there exists $s \notin bd(u)$ such that $v_1 \sim s$. Let $t = a_0, \ldots, a_n = s$ be the shortest path from t to s in $\mathcal{G}_{\mathbf{V} \setminus \{v_1\}}$, which exists since $\mathcal{G}_{\mathbf{V} \setminus \{v_1\}}$

is connected. Choose $i = \min\{j : a_j \sim v_1 \text{ in } \mathcal{G}\}$. Recall that $v_1 \not\sim t = a_0$, so $j \geq 1$. We see that $a_0, \ldots, a_j, v_1, u, t = a_0$ is a chordless cycle of length $3 + j$.

(iii ⇒ iv). Suppose that the nodes of $\mathcal{G}$ admit a perfect ordering. We prove that the cliques admit a perfect ordering by induction on $|\mathbf{V}|$. There is nothing to show for $|\mathbf{V}| = 1$. Suppose $|\mathbf{V}| > 1$ and the implication holds for $|\mathbf{V}| - 1$. Let k be the number of cliques in $\mathcal{G}$. It is enough to find the last clique in the perfect ordering. That is, we will find a clique, $\mathbf{C}_k$ such that $\mathbf{S}_k \subset \mathbf{C}_j$ for some j, where $\mathbf{S}_k = \mathbf{C}_k \cap (\cup_{i<k} \mathbf{C}_i)$. Choose u simplicial in $\mathcal{G}$. It is simple to verify that $\mathbf{C}_i = \{u\} \cup bd(u)$ is a clique and u is not an element of any other clique. If $bd(u)$ is a proper subset of another clique, say $\mathbf{C}_j$, then let $\mathbf{C}_k = \mathbf{C}_i$. Clearly, $\mathbf{S}_k = bd(u) \subset \mathbf{C}_j$. If $bd(u)$ is not a proper subset of another clique besides $\mathbf{C}_i$, then consider the graph $\mathcal{G}^* = \mathcal{G}_{\mathbf{V} \setminus \{u\}}$. Let $\mathbf{C}_1^* = \mathbf{C}_1 \setminus \{u\}, \ldots, \mathbf{C}_k^* \setminus \{u\}$ be a perfect ordering in $\mathcal{G}^*$ with $\mathbf{S}_k^* \subset \mathbf{C}_j^*$. Since u is in exactly one clique in $\mathcal{G}$, we see $\mathbf{S}_k = \mathbf{S}_k^* \subset \mathbf{C}_j^* \subseteq \mathbf{C}_j$.

(iv ⇒ i). Let $\mathbf{C}_1, \ldots, \mathbf{C}_k$ be a perfect clique ordering of $\mathcal{G}$. We show that $\mathcal{G}$ is decomposable by induction on k. For $k = 1$, $\mathcal{G}$ is complete and thus decomposable by definition. Assume $k > 1$ and the implication holds for $k - 1$. We have a proper decomposition, $(\mathbf{C}_k, \mathbf{H}_{k-1})$, where $\mathcal{G}_{\mathbf{C}_k}$ is decomposable by virtue of being complete, and $\mathcal{G}_{\mathbf{H}_{k-1}}$ is decomposable by the inductive hypothesis. □

Lemma 2.1.4 is originally Lemma 5 of Frydenberg and Lauritzen (1989). Here we present a slightly simpler proof. This is the first of the three lemmae that we will need for the stochastic search.

Lemma 2.1.4. *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and $\mathcal{G}^* = (\mathbf{V}, \mathbf{E}^*)$ be decomposable graphs with $\mathbf{E} \subset \mathbf{E}^*$. There exists an increasing sequence of decomposable graphs $\mathcal{G} = \mathcal{G}_0 \subset \cdots \subset \mathcal{G}_n = \mathcal{G}^*$ differing by exactly one edge.*

Proof. By induction, it suffices to find an edge $e \in \mathbf{E}^* \setminus \mathbf{E}$ such that $(\mathbf{V}, \mathbf{E} \cup \{e\})$ is decomposable. If $\mathcal{G}$ and $\mathcal{G}^*$ differ by exactly one edge, then e is that edge. Henceforth, assume $\mathcal{G}$ and $\mathcal{G}^*$ differ by more than one edge. For $|\mathbf{V}| \leq 3$, the lemma is trivial. Suppose $|\mathbf{V}| > 3$ and assume it holds for $|\mathbf{V}| - 1$. By Lemma 2.1.1, there is a node u which is simplicial in $\mathcal{G}$. Choose an edge e such that $(\mathbf{V} \setminus \{u\}, \mathbf{E}_{\mathbf{V}\setminus\{u\}} \cup \{e\})$ is triangulated. Let $\mathcal{G}_1 = (\mathbf{V}, \mathbf{E} \cup \{e\})$. Any chordless cycles of length 4 or more must include u, but such a one can not exist because $bd(u)$ is still complete in $\mathcal{G}_1$. □

With Lemma 2.1.4, we need only consider moves between decomposable graphs obtained by toggling a single edge. The next two lemmae provide criteria for deciding it a proposed toggle will lead to a decomposable graph. The first is useful in case we propose to remove an existing edge.

Lemma 2.1.5. *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is decomposable, and $\mathcal{G}^- = (\mathbf{V}, \mathbf{E} \setminus \{(a, b)\})$. $\mathcal{G}^-$ is decomposable if and only if $\{a, b\}$ is contained in exactly one clique.*

Proof. Clearly, $\mathcal{G}^-$ is triangulated unless there exists a 4-cycle (a, v, b, u, a) in $\mathcal{G}$ such that $a \sim b$ is the only chord. This cycle exists if and only if there exists some $v, u \in bd(a) \cap bd(b)$ such that $v \not\sim u$. Such u, v exist if and only if $\{a, b, u\}$ and $\{a, b, v\}$ are contained in two distinct cliques. □

The next lemma is a counterpart to Lemma 2.1.5. It provides a method for testing if a graph will still be decomposable when a given edge is added. We present two proofs, the first of which is a new proof that is very simple yet not so useful from an implementation point-of-view. This will demonstrate the usefulness of the junction tree representation for decomposable graphs. The second proof makes use of junction trees and provides a method which more readily lends itself to implementation.

Lemma 2.1.6. *Suppose* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *is a decomposable graph in which* $a \not\sim b$*, and* $\mathcal{G}^+ = (\mathbf{V}, \mathbf{E} \cup \{(a, b)\})$*.* $\mathcal{G}^+$ *is decomposable if and only if* $\mathbf{S} = bd(a) \cap bd(b)$ *separates* a *and* b *in* $\mathcal{G}$*.*

Proof. If $\mathcal{G}^+$ contains a cycle which is not already in $\mathcal{G}$, then the edge (a, b) closes a path $a = v_1, \ldots, v_n = b$ from a to b in $\mathcal{G}$. Thus, we consider paths in $\mathcal{G}$ of the form $a = v_1, \ldots, v_n = b$ with $n \geq 4$. We show sufficiency first. Assume $\mathbf{S} = bd(a) \cap bd(b)$ separates a and b. By definition, at least one $v_i \in \mathbf{S}$. For $i > 1$, (a, v_i) is a chord; for $i < n$, (b, v_i) is a chord. To show necessity, assume $\mathbf{S} = bd(a) \cap bd(b)$ does not separate a and b. In this case, we choose the shortest path from a to b which does not intersect $\mathbf{S}$. By choosing the smallest path, we ensure that $v_i \sim v_j$ if and only if $|i - j| = 1$. Thus, the corresponding cycle in $\mathcal{G}^+$ contains no chords. Furthermore, $b \not\sim v_2$, because v_2 is chosen in $bd(a) \setminus bd(b)$. We conclude that $n \geq 4$ and $\mathcal{G}^+$ is not decomposable. □

As noted, Lemmae 2.1.4-2.1.6 enable the use of stochastic searches on the space of decomposable graphs. Lemma 2.1.4 states that the search can cover the entire space simply by toggling one edge at a time. Thus, we can simply propose a pair of cliques,

(a, b). If $a \sim b$, we consider deleting the edge and use Lemma 2.1.5 to decide if the move is legal. On the other hand, if $a \not\sim b$, we consider adding the edge and use Lemma 2.1.6 to decide if that move is legal. The difficulty is implementing these decisions. To apply, Lemma 2.1.5 we need to examine cliques which contain a, stopping if we find two which also contain b or we exhaust the clique set. To apply, Lemma 2.1.6 we need to find $bd(a)$ and $bd(b)$, their intersection, and then somehow determine if that intersection separates a and b. The solution is to represent $\mathcal{G}$ as a junction tree.

By Lemma 2.1.1 every decomposable graph has a perfect clique ordering. A junction tree can be considered a graphical representation of this ordering. Construct the tree by beginning with $\mathbf{C}_1$. For $i > 1$, add the vertex $\mathbf{C}_i$ and an edge $(\mathbf{C}_i, \mathbf{C}_j)$, where $j < i$ is such that $\mathbf{C}_j \supset \mathbf{S}_i$. Note that for $i > 1$, $bd(\mathbf{C}_i)$ contains exactly one clique $\mathbf{C}_j$ such that $j < i$. This property ensures that the junction graph is indeed a tree. To see that this tree satisfies the junction property, choose any $\mathbf{C}_i, \mathbf{C}_j \in \mathscr{C}$, and let $\mathbf{C}_j = \mathbf{C}_{r_1}, \ldots, \mathbf{C}_{r_\ell} = \mathbf{C}_i$ be the unique path between them. Without loss of generality, assume $i > j$. By construction, $r_1 < r_2 < \cdots < r_\ell$ and for $n < m, (\mathbf{C}_{r_n} \cap \mathbf{C}_{r_m}) \subseteq (\mathbf{C}_{r_m} \cap \mathbf{C}_{r_{m-1}})$. Therefore,

$$(\mathbf{C}_j \cap \mathbf{C}_{r_{\ell-1}}) \subseteq (\mathbf{C}_j \cap \mathbf{C}_{r_{\ell-2}}) \cdots \subseteq (\mathbf{C}_j \cap \mathbf{C}_i) \tag{2.3}$$

Thus, the tree satisfies the junction property.

There are many clique orderings which yield the same junction tree, but the junction tree representation is not unique. Indeed, the junction tree indicates that every clique is first in *some* perfect ordering. Let $\mathbf{C}_1$ be an arbitrary clique, and for $i > 1$ choose $\mathbf{C}_i$ such that in the junction tree, $\mathbf{C}_i$ is connected to some $\mathbf{C}_j$ with $j < i$. For each, i we see

that $\{\mathbf{C}_1, \ldots, \mathbf{C}_i\}$ is connected in the junction tree. By virtue of the junction property, $(\mathbf{C}_i \cap \mathbf{C}_m) \subseteq \mathbf{C}_j$ for any $m < i$. Therefore, we can construct a perfect clique ordering beginning with any clique.

Using junction trees provides an easier method for testing whether proposed graphs are decomposable. Using the following lemma, if we find one clique $\mathbf{C}$ that contains $\{a, b\}$, then we only need to check its neighbors in the junction tree.

Lemma 2.1.7. *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a decomposable graph containing a clique $\mathbf{C} \supset \{a, b\}$, and $\mathcal{G}^- = (\mathbf{V}, \mathbf{E} \setminus \{(a, b)\})$. $\mathcal{G}^-$ is decomposable if and only if $\{a, b\}$ is not contained in a clique $\mathbf{C}^*$ where $\mathbf{C}^* \sim \mathbf{C}$ in the junction tree for $\mathcal{G}$.*

Proof. By Lemma 2.1.5, $\mathcal{G}^-$ is decomposable if and only if $\{a, b\}$ is not included in any other clique of $\mathcal{G}$. Suppose $\mathbf{C}^{**}$ is a clique other than $\mathbf{C}$ that contains $\{a, b\}$. If such a clique exists, then let $\mathbf{C}^*$ be the clique which neighbors $\mathbf{C}$ in the unique junction tree path from $\mathbf{C}^{**}$ to $\mathbf{C}$. By the junction property, $\{a, b\} \subseteq \mathbf{C}^*$. Therefore $\mathbf{C}^{**}$ exists if and only if $\mathbf{C}^*$ exists. □

We will now find a decision process for adding a proposed edge which utilizes the junction tree representation. The decision involves the notion of *the shortest path* between a and b in the junction tree. Specifically, we require the shortest path in the junction tree between any cliques $\mathbf{A}$ and $\mathbf{B}$ with $a \in \mathbf{A}$ and $b \in \mathbf{B}$. We show now that the shortest path is unique and that the endpoints separate a and b.

Let $\mathbf{B}$ be an arbitrary clique which contains b. Let $\mathbf{A} = \mathbf{C}_1, \ldots, \mathbf{C}_n = \mathbf{B}$ be "a" shortest path to $\mathbf{B}$ starting from any $\mathbf{A} \ni a$. We will show that this path is unique and therefore well-defined. Let $\mathbf{A}^* = \mathbf{C}_1^*, \ldots, \mathbf{C}_m^* = \mathbf{B}$, $m \geq n$ be any other path. Clearly, these paths intersect (if only because $\mathbf{B} = \mathbf{C}_n = \mathbf{C}_m^*$). Set $i = \min\{i : \mathbf{C}_i = \mathbf{C}_j^* \text{ for some } j\}$ and choose j such that $\mathbf{C}_j^* = \mathbf{C}_i$. $\mathbf{C}_1, \ldots, \mathbf{C}_i = \mathbf{C}_j^*, \ldots, \mathbf{C}_1^*$ is the path from $\mathbf{A}$ to $\mathbf{A}^*$ in the junction tree. By the junction property, $a \in \mathbf{C}_i$. Therefore, $\mathbf{C}_i, \ldots, \mathbf{C}_n = \mathbf{B}$ is a path from a set containing a to $\mathbf{B}$, so $i = 1$ by our choice of the shortest path. We conclude that either $m > n$ (the second path is not as short as the first) or $\mathbf{A}^* = \mathbf{A}$ (the paths are identical). Once $\mathbf{A}$ is proven to exist, we can apply the argument again to find the shortest path from a $\mathbf{B} \ni b$ to $\mathbf{A}$. We see too that if $\mathbf{A}^*$ and $\mathbf{B}^*$ are any other cliques that contain a to b then the path between them in the junction tree includes $\mathbf{A}$ and $\mathbf{B}$. This is important for the following reason. If $v \in \{a\} \cup bd(a)$ and $u \in \{b\} \cup bd(b)$, then there are cliques $\mathbf{A}^*$ and $\mathbf{B}^*$ such that $\{a, v\} \subseteq \mathbf{A}^*$ and $\{b, u\} \subseteq \mathbf{B}^*$. Invoking the junction property, we see that $\mathbf{A}$ and $\mathbf{B}$ each separates $\{a\} \cup bd(a)$ and $\{b\} \cup bd(b)$. This leads to the restatement of Lemma 2.1.6 for the junction tree representation.

Lemma 2.1.8. *Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a decomposable graph such that $a \sim b$. Let $\mathbf{A} = \mathbf{C}_1, \ldots, \mathbf{C}_n = \mathbf{B}$ be the shortest path between a clique containing a to a clique containing b in a junction tree representation of $\mathcal{G}$. The graph $\mathcal{G}^+ = (\mathbf{V}, \mathbf{E} \cup \{a, b\})$ is decomposable if and only if $\mathbf{A} \cap \mathbf{B} = \mathbf{C}_i \cap \mathbf{C}_{i+1}$ for some i.*

Proof. We have already seen that $\mathbf{A}$ and $\mathbf{B}$ each separate $bd(a)$ from $bd(b)$. Hence, if $bd(a) \cap bd(b)$ separates $\mathbf{A}$ and $\mathbf{B}$, then it also separates $bd(a)$ from $bd(b)$. Clearly, this

condition is also necessary since $\mathbf{A} \subseteq bd(a)$ and $\mathbf{B} \subseteq bd(b)$. Finally, the junction property implies that $\mathbf{A} \cap \mathbf{B} = bd(a) \cap bd(b)$. These facts combined show that the condition of Lemma 2.1.6 is equivalent to the condition that $\mathbf{A} \cap \mathbf{B}$ separates $\mathbf{A}$ and $\mathbf{B}$. This holds if and only if $\mathbf{A} \cap \mathbf{B}$ is a separator on the unique path from $\mathbf{A}$ to $\mathbf{B}$. □

In Chapter 5 we discuss details for implementing Lemma 2.1.7 and Lemma 2.1.8, including finding cliques which contain a given node, testing adjacency, computing shortest paths, and local updates to the junction tree when edges are added or dropped.

2.2 Markov Measures for Graphical Models

An undirected graph depicts the conditional independence structure for some variable X. Distributions that satisfy these constraints are called *Markov probability measures.*

Definition 2.2.1 (Markov Probability Measure)**.** *If θ is a probability measure on $(\mathcal{X}, \mathcal{F})$, we say it is* Markov *on a decomposable graph, $\mathcal{G}$, if for any proper decomposition $(\mathbf{A}, \mathbf{B})$,*

$$X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \mid X_{\mathbf{A}\cap\mathbf{B}}[\theta].$$

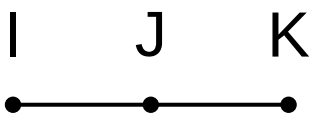

Figure 2.1: A graph depicting conditional independence of X_I and X_K given X_J.

For example, let $\mathcal{G}$ be the graph depicted in Figure 2.1. A measure θ is Markov on $\mathcal{G}$, if and only if $X_I \perp\!\!\!\perp X_K \mid X_J[\theta]$. Implicit in the definition is the fact that it is only sensible to refer to a measure as Markov in relation to a specific graph. For example, if the measure θ is not Markov on the graph in Figure 2.1, is still Markov on the complete graph with $\mathbf{V} = \{I, J, K\}$. Since a complete graph admits no proper decomposition, all measures over $\mathcal{X}_\mathbf{V}$ are trivially Markov. Alternatively, a measure such that the components $\{X_v\}$ are mutually independent is Markov on any graph (with the appropriate vertex set.)

We denote the set of all distributions that are Markov on $\mathcal{G}$ by $\mathscr{M}(\mathcal{G})$. As written, the conditional independence property in Definition 2.2.1 applies only when $\mathbf{A}$ and $\mathbf{B}$ are a decomposition of $\mathcal{G}$. Dawid and Lauritzen (1993) provide an equivalent expression in terms of more general sets. They show that a probability measure is Markov if and only if it satisfies the *global Markov property*:

Definition 2.2.2 (Global Markov Property)**.** *A measure θ on $(\mathcal{X}, \mathcal{F})$ satisfies the* global Markov property *if for $X \sim \theta$,*

$$X_\mathbf{A} \perp\!\!\!\perp X_\mathbf{B} \mid X_\mathbf{S}[\theta] \text{ whenever } \mathbf{S} \text{ separates } \mathbf{A} \text{ and } \mathbf{B}. \tag{2.4}$$

This property is global in the sense that $\mathbf{A}$ and $\mathbf{B}$ may be any subsets of $\mathbf{V}$. Likewise, $\mathbf{S}$ may be any subset that separates $\mathbf{A}$ and $\mathbf{B}$; it need not be their intersection.

2.2.1 Markov Combinations

Let us consider a graph consisting of exactly two cliques, $\mathbf{A}$ and $\mathbf{B}$ with separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. This graph admits only one proper decomposition, namely $(\mathbf{A}, \mathbf{B})$. Thus, a measure θ is

Markov on this graph if and only if $X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \mid X_{\mathbf{S}}[\theta]$. Dawid and Lauritzen (1993) show that such a measure is uniquely identified by its marginals $\theta_{\mathbf{A}}$ and $\theta_{\mathbf{B}}$. That is to say that if there exist two appropriate marginal measures for the two cliques, then there is only one joint measure that is Markov and has those marginals. We will now formally express the notion of "appropriate" and the result of Dawid and Lauritzen.

Definition 2.2.3 (Consistency of Probability Measures)**.** *Let Q be a measure on $(\mathcal{X}_{\mathbf{A}}, \mathcal{F}_{\mathbf{A}})$ and R be a measure on $(\mathcal{X}_{\mathbf{B}}, \mathcal{F}_{\mathbf{B}})$. We say that Q and R are* consistent *probability measures if $Q_{\mathbf{A}\cap\mathbf{B}} = R_{\mathbf{A}\cap\mathbf{B}}$.*

Proposition 2.2.4 (Markov Combination)**.** *Let Q be a measure on $(\mathcal{X}_{\mathbf{A}}, \mathcal{F}_{\mathbf{A}})$ and R be measure on $(\mathcal{X}_{\mathbf{B}}, \mathcal{F}_{\mathbf{B}})$ such that Q and R are consistent. There exists an almost everywhere unique distribution θ such that $\theta_{\mathbf{A}} = Q$, $\theta_{\mathbf{B}} = R$, and $X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \mid X_{\mathbf{A}\cap\mathbf{B}}[\theta]$.*

We call the distribution θ satisfying Proposition 2.2.4 the *Markov Combination* of Q and R and write $\theta = Q \star R$. Clearly, if $\mathcal{G}$ is a decomposable graph with two cliques $\mathbf{A}$ and $\mathbf{B}$ then θ is Markov on $\mathcal{G}$. More generally, let $\mathcal{G}$ be a decomposable graph with cliques given by $\mathbf{C}_1, \ldots, \mathbf{C}_k$. A unique Markov distribution can be constructed from a given set of pairwise consistent marginal distributions by iteratively forming Markov combinations. Let $\theta_{\mathbf{C}_i}$ be a measure on $(\mathcal{X}_{\mathbf{C}_i}, \mathcal{F}_{\mathbf{C}_i})$ with the consistency constraint that for all i, j, $\theta_{\mathbf{C}_i}$ and $\theta_{\mathbf{C}_j}$ induce the same marginal for $(\mathcal{X}_{\mathbf{C}_i\cap\mathbf{C}_j}, \mathcal{F}_{\mathbf{C}_i\cap\mathbf{C}_j})$. Define the i^{th} history, $\mathbf{H}_i = \cup_{j<=i}\mathbf{C}_j$. Of course, $\theta_{\mathbf{H}_1} = \theta_{\mathbf{C}_1}$. For $i \geq 2$, set $\theta_{\mathbf{H}_i} = \theta_{\mathbf{C}_i} \star \theta_{\mathbf{H}_{i-1}}$. The last combination, $\theta_{\mathbf{H}_k}$ is the unique probability measure which is Markov with respect to $\mathcal{G}$ and has the given clique marginals $\{P_{\mathbf{C}_i}\}$. We will denote this relationship by $\theta_{\mathbf{H}_k} = \star(\theta_{\mathbf{C}_1}, \ldots, \theta_{\mathbf{C}_k})$.

2.3 Hyper Markov Prior Laws

Dawid and Lauritzen (1993) extend the notion of Markovity from random variables to random parameters. To reduce confusion, we will refer to a distribution over distributions as a *law*. By a law over $\mathbf{A}$ or $\mathcal{G}_\mathbf{A}$ we shall mean a prior law for $\theta_\mathbf{A}$, the marginal distribution of $X_\mathbf{A}$. Consider a graphical model, $\theta \in \mathscr{M}(\mathcal{G})$ and $X|\theta \sim \theta$. In a Bayesian frame set, θ is a random distribution, so we specify a prior law $\mathscr{L}$ over $\mathscr{M}(\mathcal{G})$. When the prior law also satisfies the conditional independence structure of $\mathcal{G}$, we say that it is *(weak) hyper Markov*. This is expressed formally in Definition 2.3.1. We shall see in Section 2.3.1 that under this constraint, $\mathscr{L}$ is defined by its clique marginals. In Section 2.3.2, we discuss the stronger version of this property, which to leads several desirable results.

2.3.1 The Weak Hyper Markov Property

Definition 2.3.1 (Hyper Markov Law)**.** *A prior law $\mathscr{L}$ is (weak)* hyper Markov *with respect to a graph $\mathcal{G}$ if it gives probability one to $\mathscr{M}(\mathcal{G})$ and for any decomposition $(\mathbf{A}, \mathbf{B})$ of $\mathcal{G}$:*

$$\theta_\mathbf{A} \perp\!\!\!\perp \theta_\mathbf{B} \mid \theta_{\mathbf{A}\cap\mathbf{B}}. \tag{2.5}$$

The analogy of this definition to Markov distributions is clear. The analogy can be extended, in that many of the properties of Markov distributions are also true for hyper Markov laws. For example, hyper Markovity may be expressed as an equivalent "global" property. A law $\mathscr{L}$ is hyper Markov with respect to $\mathcal{G}$ if and only if it satisfies the *global hyper Markov property*.

Definition 2.3.2 (Global Hyper Markov Property)**.** *A law $\mathscr{L}$ on $(\mathcal{X}, \mathcal{F})$ satisfies the* global hyper Markov property *if for $\theta \sim \mathscr{L}$,*

$$\theta_{\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{B}} \mid \theta_{\mathbf{S}}[\mathscr{L}] \text{ whenever } \mathbf{S} \text{ separates } \mathbf{A} \text{ and } \mathbf{B}. \tag{2.6}$$

As the goal of this section is to consider prior laws for $\theta \in \mathscr{M}(\mathcal{G})$, it is desirable to construct an operation which combines marginal laws on the cliques of $\mathcal{G}$ in the same vein as the Markov combination of Proposition 2.2.4. Of course, we first need a consistency criterion.

Definition 2.3.3 (Consistency of Hyper Markov Laws)**.** *Let $\mathscr{Q}$ be a law for $\theta_{\mathbf{A}}$ and $\mathscr{R}$ be a law for $\theta_{\mathbf{B}}$. We say that $\mathscr{Q}$ and $\mathscr{R}$ are* hyper consistent *laws if $\mathscr{Q}_{\mathbf{A}\cap\mathbf{B}} = \mathscr{R}_{\mathbf{A}\cap\mathbf{B}}$.*

As with Markov measures, hyper Markov laws are determined by their clique marginals. Suppose each clique is endowed with a prior law for some random distribution.

Definition 2.3.4 (Hyper Markov Combination)**.** *Let $\mathscr{Q}$ be a prior law for $\theta_{\mathbf{A}}$ and $\mathscr{R}$ be a prior law for $\theta_{\mathbf{B}}$ such that $\mathscr{Q}$ and $\mathscr{R}$ are hyperconsistent. The* hyper Markov combination *of $\mathscr{Q}$ and $\mathscr{R}$, denoted $\mathscr{Q} \odot \mathscr{R}$, is the unique law, $\mathscr{L}$, such that:*

1. $\theta \in \mathscr{M}(\mathcal{G})$ *a.s.*$[\mathscr{L}]$,
2. $\mathscr{L}_{\mathbf{A}} = \mathscr{Q}$,
3. $\mathscr{L}_{\mathbf{B}} = \mathscr{R}$,
4. $\theta_{\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{B}} \mid \theta_{\mathbf{A}\cap\mathbf{B}}[\mathscr{L}]$.

This law exists and is almost-everywhere unique by Lemma 3.3 of Dawid and Lauritzen (1993).

If $\mathcal{G}$ is a decomposable graph and its cliques are $\mathbf{A}$ and $\mathbf{B}$, then $\mathscr{L}$ in Definition 2.3.4 is hyper Markov with respect to $\mathcal{G}$. As desired, this idea may be used iteratively for constructing hyper Markov laws for general decomposable graphs. Let $\mathcal{G}$ be a decomposable graph with cliques $\mathbf{C}_1, \ldots, \mathbf{C}_k$, each with a marginal law $\mathscr{L}_{\mathbf{C}_i}$ for $\theta_{\mathbf{C}_i}$ such that the marginal laws are pair-wise hyperconsistent. Take $\mathscr{L}_{\mathbf{H}_1} = \mathscr{L}_{\mathbf{C}_1}$. For $i \geq 2$, define $\mathscr{L}_{\mathbf{H}_i} = \mathscr{L}_{\mathbf{C}_i} \odot \mathscr{L}_{\mathbf{H}_{i-1}}$. $\mathscr{L} = \mathscr{L}_{\mathbf{H}_k}$ is the unique law which is hyper Markov with respect to $\mathcal{G}$ and has the given clique marginal laws $\{\mathscr{L}_{\mathbf{C}_i}\}$. We will denote this relationship by $\mathscr{L} = \odot(\mathscr{L}_{\mathbf{C}_1}, \ldots, \mathscr{L}_{\mathbf{C}_k})$.

2.3.2 The Strong Hyper Markov Property

Once again, we consider a graphical model $X \sim \theta$, where $\theta \in \mathscr{M}(\mathcal{G})$ is a random distribution with prior law $\mathscr{L}$. Since θ is Markov, the data X are guaranteed to exhibit the desired independence graph, *but only conditionally given* θ. That is, the *marginal* distribution of X is not guaranteed to be Markov. Thus, marginal calculations lose the dimension-reducing advantage of graphical models. It turns out that even if $\mathscr{L}$ is hyper Markov, the marginal distribution of X may not by Markov; a stronger property is needed. This *strong hyper Markov* property will ensure that the distribution of X is Markov, which will be a useful property when we discuss graph selection (see Section 2.4.2.)

Definition 2.3.5 (Strong Hyper Markov Law)**.** *A law $\mathscr{L}$ on $\mathscr{M}(\mathcal{G})$ is called* strong hyper Markov *over $\mathcal{G}$ if for any decomposition $(\mathbf{A}, \mathbf{B})$ of $\mathcal{G}$*

$$\theta_{\mathbf{B}|\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{A}}[\mathscr{L}]. \tag{2.7}$$

Note that this property implies that $\mathscr{L}$ is weak hyper Markov, so the discussion of weak hyper Markov priors applies. For example, a hyper Markov law on a graph $\mathcal{G}$ is determined by its clique marginal laws. For a given set of marginals, there is one hyper Markov law. Whether that law is strong or weak must therefore depend on properties of the marginals. In fact, Dawid and Lauritzen (1993) show the following:

Proposition 2.3.6. *Let $\mathscr{L}$ be hyper Markov over $\mathcal{G}$. Then $\mathscr{L}$ is strong hyper Markov if and only if, for all cliques $\mathbf{C}$ of the graph $\mathcal{G}$ and all subsets $\mathbf{A}$ of $\mathbf{C}$ we have*

$$\theta_{\mathbf{C}\setminus\mathbf{A}|\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{A}} \ [\mathscr{L}] \tag{2.8}$$

Proof. This is Proposition 3.16 of Dawid and Lauritzen (1993). □

Dawid and Lauritzen (1993) prove the next proposition about the joint law of θ and $X \sim \theta$.

Proposition 2.3.7. *If $\mathscr{L}(\theta)$ is hyper Markov over $\mathcal{G}$, and $X \sim \theta$, then the joint distribution of (X, θ) satisfies, for any decomposition $(\mathbf{A}, \mathbf{B})$ of $\mathcal{G}$,*

$$(X_{\mathbf{A}}, \theta_{\mathbf{A}}) \perp\!\!\!\perp (X_{\mathbf{B}}, \theta_{\mathbf{B}}) \mid (X_{\mathbf{A}\cap\mathbf{B}}, \theta_{\mathbf{A}\cap\mathbf{B}}). \tag{2.9}$$

Additionally, if $\mathscr{L}(\theta)$ is strong *hyper Markov, then*

$$(X_{\mathbf{A}}, \theta_{\mathbf{A}}) \perp\!\!\!\perp (X_{\mathbf{B}}, \theta_{\mathbf{B}|\mathbf{A}}) \mid X_{\mathbf{A}\cap\mathbf{B}}. \tag{2.10}$$

This proposition has three important corollaries for our work. The first corollary states that the family of (strong) hyper Markov laws is conjugate to the family of Markov distributions for a particular graph. We show the result for a single observation X, which implies inductively to a sample of size $n > 1$ since we may introduce one observation at a time.

Corollary 2.3.8 (Conjugacy of Hyper Markov Laws)**.** *If the prior law of θ is hyper Markov, so is the posterior obtained by conditioning on a complete observation $X = x$. If the prior law is strong hyper Markov, then so is the posterior*

Proof. From Equation 2.9, it follows immediately that

$$\theta_{\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{B}} \mid (\theta_{\mathbf{A}\cap\mathbf{B}}, X_{\mathbf{A}\cap\mathbf{B}}, X_{\mathbf{A}}, X_{\mathbf{B}}). \tag{2.11}$$

.

Since $(\mathbf{A}, \mathbf{B})$ is a decomposition of $\mathcal{G}$, we have $\mathbf{V} = \mathbf{A} \cup \mathbf{B}$. Therefore, X and the triplet $(X_{\mathbf{A}}, X_{\mathbf{B}}, X_{\mathbf{A}\cap\mathbf{B}})$ are functions of each other so conditioning on the triple is equivalent to conditioning on X. Equation 2.11 becomes

$$\theta_{\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{B}} \mid (\theta_{\mathbf{A}\cap\mathbf{B}}, X). \tag{2.12}$$

The proof is similar in the strong hyper Markov case. □

The remaining two corollaries only apply when the prior law is strong hyper Markov. Under this condition, we are guaranteed two results which will be of much importance

throughout this work for discussing applications of the hyper Dirichlet process. The first states that when the prior law is strong hyper Markov, we can determine the posterior by locally updating the prior marginal laws for the cliques. The second states that the marginal distribution of X is Markov.

Corollary 2.3.9 (Local Updates for Strong Hyper Markov Laws)**.** *Let $\mathcal{G}$ be a decomposable graph with clique set $\mathscr{C}$. If the prior law $\mathscr{L}(\theta)$ is strong hyper Markov on $\mathcal{G}$, then the posterior law of θ is the unique strong hyper Markov law $\mathscr{L}^*$ specified by the clique marginal laws $\{\mathscr{L}^*_{\mathbf{C}} : \mathbf{C} \in \mathscr{C}\}$, where $\mathscr{L}^*_{\mathbf{C}}$ is the posterior law of $\theta_{\mathbf{C}}$ based on the prior law $\mathscr{L}_{\mathbf{C}}$ and the clique-specific data, $\mathbf{X}_{\mathbf{C}} = x_{\mathbf{C}}$.*

Proof. If $\mathbf{C}$ is any clique in $\mathcal{G}$, then it forms a proper decomposition of $\mathcal{G}$ with the union of the remaining cliques. Straightforward application of Equation 2.10 reveals $\theta_{\mathbf{C}} \perp\!\!\!\perp X | X_{\mathbf{C}}$. Thus $\theta_{\mathbf{C}}$ depends on the data only through $X_{\mathbf{C}}$. Applying this to each clique, we see that the clique marginals of $\mathscr{L}$, namely $\{\mathscr{L}^*_{\mathbf{C}} : \mathbf{C} \in \mathscr{C}\}$, may be computed locally by considering only the clique-specific data $x_{\mathbf{C}}$. By Corollary 2.3.8, $\mathscr{L}^*$ is strong Hyper Markov, and by Definition 2.3.4 it is unique. □

Corollary 2.3.10 (Data Marginal under Strong Hyper Markov Laws)**.** *If the prior law of θ is strong hyper Markov, then the marginal distribution of X is Markov.*

Proof. This follows directly from Equation 2.10 since $\theta_{\mathbf{A}}$ and $\theta_{\mathbf{B}|\mathbf{A}}$ may be removed while maintaining the conditional independence property. □

To elucidate the discussion of this section we provide two examples of hyper Markov laws: one that is strong and one that is weak (Dawid and Lauritzen, 1993).

Example 2.3.1 (Strong Hyper Markov Law)**.** *Suppose $X = (X_I, X_J, X_K)$ is a discrete random variable with $\mathbb{P}(X = (i, j, k)) = \theta_{ijk}$. We construct a law $\mathscr{L}$ which is hyper Markov on the graph $\mathcal{G}$ shown in Figure 2.1. $\theta \in \mathscr{M}(\mathcal{G})$ if it satisfies*

$$\theta_{ijk} = \theta_{ij+}\theta_{+jk}/\theta_{+j+}. \tag{2.13}$$

We specify the marginal laws for $\{\theta_{ij+}\}$ and $\{\theta_{+jk}\}$ as Dirichlet laws with $\{\alpha_{ij}\}$ and $\{\beta_{jk}\}$. Note that these are hyperconsistent if and only if $\alpha_{+j} = \beta_{j+}$ for all j. By Definition 2.3.4 there is a unique hyper Markov law $\mathscr{L}$ with these marginals. In other words, if we constrain $\mathscr{L}(\theta)$ to satisfy the hyper Markov property for $\mathcal{G}$,

$$\theta_{ij+} \perp\!\!\!\perp \theta_{+jk} \mid \theta_{+j+}[\mathscr{L}], \tag{2.14}$$

then the law $\mathscr{L}$ is uniquely determined by $\{\alpha_{ij}\}$ and $\{\beta_{jk}\}$.

We now show that the law $\mathscr{L}$ constructed above is strong hyper Markov. Consider the marginal law for $\{\theta_{ij+}\}$, which we have stated is Dirichlet with parameters $\{\alpha_{ij}\}$. Note that the conditional distributions $\theta_{I|J}$ are equal to $\{\theta_{ij+}/\theta_{+j+}\}$, where $\theta_{+j+} = \sum_i \theta_{ij+}$. From Dirichlet Property 3.1.5, we see that $\{\theta_{ij+}/\sum_i \theta_{ij+}\} \perp\!\!\!\perp \{\sum_i \theta_{ij+}\}$. Therefore we have $\theta_{I|JK} = \theta_{I|J}$ is independent of θ_J and therefore θ_{JK}. By symmetry, $\theta_{K|IJ} \perp\!\!\!\perp \theta_{IJ}$. Since (IJ, JK) is the only proper decomposition of $\mathcal{G}$, we conclude that $\mathscr{L}$ is strong hyper Markov. Technically, we must also show the independence property for improper decompositions since it is not guaranteed; however in this case, it is trivial to show that the $\theta_{\mathbf{A}|\mathbf{V}} \perp\!\!\!\perp \theta_{\mathbf{V}}$ for any complete set $\mathbf{A}$.

The next example of a hyper Markov law is strictly weak.

Example 2.3.2 (Hyper Markov Sampling Law for MLE)**.** *Consider a three-way contingency table for the counts* $\{N_{ijk}\}$ *which are assumed to follow a multinomial distribution with size* n *and probabilities* $\theta = \{\theta_{ijk}\}$*. When* θ *is assumed to be Markov on the graph* $\mathcal{G}$ *in Figure 2.1, then the maximum likelihood estimator satisfies*

$$\hat{\theta}_{ijk} = \frac{N_{ij+}N_{+jk}}{nN_{+j+}}. \tag{2.15}$$

The marginal MLE for θ_{IJ} *satisfies*

$$\hat{\theta}_{ij+} = \sum_k \hat{\theta}_{ijk} = \frac{N_{ij+}}{n}. \tag{2.16}$$

Therefore, $\{\hat{\theta}_{ij+} \perp\!\!\!\perp \hat{\theta}_{+jk} | \hat{\theta}_{+j+}\}$ *and we conclude that the sampling distribution of the MLE is hyper Markov. It is straightforward to show that this law is not strong hyper Markov. The set of marginal distributions* $\theta_{I|J}$ *satisfies*

$$\hat{\theta}_{i|j} = \frac{\hat{\theta}_{ij+}}{\hat{\theta}_{+j+}} = \frac{N_{ij+}}{N_{+j+}}, \tag{2.17}$$

which is not independent of $\hat{\theta}_{+j+} = N_{+j+}/n$ *because* $n\hat{\theta}_{i|j}\hat{\theta}_{+j+} = N_{ij+}$ *is constrained to be an integer. Given that there is a unique hyper Markov combination, we can also conclude that there is no strong hyper Markov sampling law for the MLE.*

2.4 Graph Selection and Models

Suppose we have a set of observations $X^{(n)} = (X^1, \ldots, X^n)$, which are a random sample from a distribution θ. In some applications, we may desire to find the sparsest possible

graph for which θ is Markov. By "sparsest", we mean that the graph does not have any "extra" edges. This is to say

$$X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \mid X_{\mathbf{S}}[\theta], \tag{2.18}$$

if *and only if* $\mathbf{S}$ separates $\mathbf{A}$ and $\mathbf{B}$. In other words, the graph we seek, $\mathcal{G}$, is the decomposable graph such that $\theta \in \mathscr{M}(\mathcal{G})$, and if $\mathcal{G}^*$ is another decomposable graph obtained by deleting one or more edges from $\mathcal{G}$, then $\theta \notin \mathscr{M}(\mathcal{G}^*)$.

For each graph, $\mathcal{G}$, let $\mathscr{L}_{\mathcal{G}}$ be a hyper Markov prior law over $\mathscr{M}(\mathcal{G})$. We shall refer to these $\{\mathscr{L}_{\mathcal{G}}\}$ as *graphical models*. One approach to choosing between the graphs is to choose the corresponding graphical model with the highest marginal likelihood for the data $X^{(n)}$. A more Bayesian approach is to place a prior distribution over the competing graphs, then choose the graphical model with the highest posterior probability. The posterior probability is determined by weighing the prior probability by the marginal likelihood of the data under the corresponding model.

2.4.1 Compatible Priors

The prior laws we choose for graph selection, $\{\mathscr{L}_{\mathcal{G}}\}$, will necessarily vary from graph to graph because the space of Markov distributions depends on the graph. However, it makes sense that these laws be as close as possible in some sense, unless there is some expert reason to do otherwise. In this way, the difference between marginal likelihoods or posterior probabilities for the graphs will - as much as is possible - be attributable to real differences between the graphs and not artificially different priors. We begin by considering the special

case of two competing models for which one graph can be obtained by deleting one or more edges from the other. In this case, we call the former graph a *specialization* of the latter. We then generalize by considering two graphs such that neither is a specialization of the other. Finally, we will be ready to consider comparing multiple graphs, including choosing among all decomposable graphs.

Let $G = (\mathbf{V}, \mathbf{E})$ and $G^* = (\mathbf{V}, \mathbf{E}^*)$ be two competing graphs such that $\mathbf{E}^* \subset \mathbf{E}$. Since $\mathcal{G}^*$ is a *specialization* of $\mathcal{G}$, a law over $\mathscr{M}(\mathcal{G})$ specifies a natural choice for a law over $\mathscr{M}(\mathcal{G}^*)$. If $\mathscr{L}_\mathcal{G}$ is the graphical model for $\mathcal{G}$, then the marginal law $\mathscr{L}_{\mathcal{G}^*}$ for each clique of $\mathcal{G}^*$ should be equal to the marginal law of the set induced by $\mathscr{L}_\mathcal{G}$. As these marginals are induced by the overall measure $\mathscr{L}_\mathbf{G}(\theta)$, they are necessarily hyperconsistent. Thus, our choice is the unique law, $\mathscr{L}_{\mathcal{G}^*}$ which is hyper Markov on $\mathcal{G}^*$ and gives the cliques of $\mathcal{G}^*$ the same marginal laws as $\mathscr{L}_\mathcal{G}$ does. This process works because each clique in $\mathcal{G}^*$ is complete in $\mathcal{G}$, so all the necessary interactions of $\mathscr{L}_{\mathcal{G}^*}$ are extant in $\mathscr{L}_\mathcal{G}$.

Let us now consider a more general scenario of two graphs, in which neither graph is a specialization of the other. Suppose we wish to decide between $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and $\mathcal{G}^* = (\mathbf{V}, \mathbf{E}^*)$. A compatible prior can be specified by considering $\mathcal{G}^+ = (\mathbf{V}, \mathbf{E}^+)$, where $\mathbf{E}^+ = \mathbf{E} \cup \mathbf{E}^*$. If $\mathcal{G}^+$ is decomposable, we can place a prior law on $\mathscr{M}(\mathcal{G}^+)$ and form $\mathscr{L}_\mathcal{G}$ and $\mathscr{L}_{\mathcal{G}^*}$ by marginalization as above. If $\mathcal{G}^+$ is not decomposable, then we can make it so by adding chords to its edges until it is triangulated. Note that *removing* edges is not an option because we need each clique in $\mathcal{G}$ and $\mathcal{G}^*$ to be complete in $\mathcal{G}^+$. Alternatively, we may still be able to choose $\mathscr{L}_{\mathcal{G}^+}$ such that it satisfies the conditional independence

structure even if $\mathcal{G}^+$ is not decomposable. This requires a generalizing the work on hyper Markov priors to accommodate non-decomposable graphs.

It is simple to extend the procedure of specifying compatible priors when we have multiple competing priors, $\{\mathcal{G}^1, \ldots, \mathcal{G}^m\}$. Let $\mathcal{G}^+$ be a "generalized" graph which contains any edge that exists in any of the m competing graphs. If $\mathcal{G}^+$ is decomposable, then we simply specify a hyper Markov law $\mathscr{L}_{\mathcal{G}^+}$ over $\mathscr{M}(\mathcal{G}^+)$. (If $\mathcal{G}^+$ is not decomposable, and this is a problem, we simply add chords to its cycles until it is triangulated before specifying $\mathscr{L}_{\mathcal{G}^+}$.) For example, if we wish to compare all decomposable graphs, then we specify a prior law for the complete model. There are no restrictions on this law because all laws are hyper Markov with respect to the complete graph. The hyper Markov prior for any particular graph is the unique hyper Markov law whose clique marginals are equal to marginal laws for the overall prior of the complete graph.

2.4.2 Strong Hyper Markov Priors for Graph Selection

The graph selection process is more efficient when the competing graphical models incorporate *strong* hyper Markov prior laws. Let $\mathcal{G}$ be a decomposable graph with a perfect sequence of cliques, $\mathbf{C}_1, \ldots, \mathbf{C}_k$ having separators $\mathbf{S}_2, \ldots, \mathbf{S}_k$. Consider the graphical model where $\mathscr{L}(\theta)$ is a hyper Markov prior over $\mathscr{M}(\mathcal{G})$ and $X|\theta \sim \theta$. In this case, Corollary 2.3.10 ensures that the marginal distribution of the data is Markov, which means that we can compute it locally.

$$f(x) = f_{\mathbf{C}_1}(x_{\mathbf{C}_1}) \prod_{i=2}^{k} f_{\mathbf{C}_i|\mathbf{H_{i-1}}}(x_{\mathbf{C}_i}|X_{\mathbf{H}_{i-1}} = x_{\mathbf{H}_{i-1}}) \tag{2.19}$$

Since the cliques are perfectly ordered, we have that $\mathbf{S}_i$ separates $\mathbf{C}_i$ and $\mathbf{H}_{i-1}$. Invoking the Global Markov Property,

$$f(x) = f_{\mathbf{C}_1}(x_{\mathbf{C}_1}) \prod_{i=2}^{k} f_{\mathbf{C}_i|\mathbf{S_i}}(x_{\mathbf{C}_i}|X_{\mathbf{S}_i} = x_{\mathbf{S}_i}). \tag{2.20}$$

By definition, $\mathbf{S}_i \subset \mathbf{C}_i$ which implies $\{X_{\mathbf{S}_i} = x_{\mathbf{S}_i}\} \supset \{X_{\mathbf{C}_i} = x_{\mathbf{c}_i}\}$ for any given value x. Therefore, $f_{\mathbf{C}_i|\mathbf{S}_i}(x_{\mathbf{C}_i}|X_{\mathbf{S}_i} = x_{\mathbf{S}_i}) = f_{\mathbf{C}_i}(x_{\mathbf{C}_i})/f_{\mathbf{S}_i}(x_{\mathbf{S}_i})$. This yields

$$f(x) = f_{\mathbf{C}_1}(x_{\mathbf{C}_1}) \prod_{i=2}^{k} \frac{f_{\mathbf{C}_i}(x_{\mathbf{C}_i})}{f_{\mathbf{S}_i}(x_{\mathbf{C}_i})}. \tag{2.21}$$

We see that the calculation of the data marginal may be computed locally, within each clique and separator.

Dawid and Lauritzen (1993) discuss another advantage of compatible hyper Markov priors for log-likelihood tests. Suppose $\mathcal{G}^* \subset \mathcal{G}$ are decomposable graphical models which differ only by one edge, $e = (a, b)$. By Lemma 2.1.5, $\{a, b\}$ is contained in exactly one clique in $\mathcal{G}$, say $\mathbf{C}$. Choose a perfect ordering of the cliques of $\mathcal{G}$ where $\mathbf{C} = \mathbf{C}_1$. Therefore Equation 2.21 is the likelihood for the model $\mathcal{G}$ where $\mathbf{C} = \mathbf{C}_1$. Certainly, $\mathbf{C}$ is not a clique in $\mathcal{G}^*$. However, note that $\{a, b\}$ is not contained in any separator $\mathbf{S}_i$ so each $\mathbf{S}_i$ remains complete in $\mathcal{G}^*$. Therefore, it is not too hard to see that Equation 2.21 can also be used to calculate $f^*(x)$, the likelihood for $\mathcal{G}^*$. Furthermore, if the prior laws

are compatible, $f_{\mathbf{C}_i}(x_{\mathbf{C}_i}) = f^*_{\mathbf{C}_i}(x_{\mathbf{C}_i})$ and $f_{\mathbf{C}_i}(x_{\mathbf{C}_i}) = f^*_{\mathbf{C}_i}(x_{\mathbf{C}_i})$ for $i > 1$. Thus the likelihood ratio for the two models is $L(\mathcal{G}^* : \mathcal{G}) = f^*_{\mathbf{C}}(x_{\mathbf{C}})/f_{\mathbf{C}}(x_{\mathbf{C}})$. Note that $f^*_{\mathbf{C}}(x_{\mathbf{C}}) = f^*_{\mathbf{C}_a}(x_{\mathbf{C}_a})f^*_{\mathbf{C}_b}(x_{\mathbf{C}_b})/f^*_{\mathbf{C}_0}(x_{\mathbf{C}_0})$, where $\mathbf{C}_a = \mathbf{C}\setminus\{a\}, \mathbf{C}_b = \mathbf{C}\setminus\{b\}$, and $\mathbf{C}_0 = \mathbf{C}\setminus\{a, b\}$. Note that each of these sets is complete in both $\mathcal{G}$ and $\mathcal{G}^*$. Since the prior laws are compatible, the marginal likelihoods are the same for both models. Hence, the likelihood ratio can be written

$$L(\mathcal{G}^* : \mathcal{G}) = \frac{f_{\mathbf{C}_a}(x_{\mathbf{C}_a})f_{\mathbf{C}_b}(x_{\mathbf{C}_b})}{f_{\mathbf{C}_0}(x_{\mathbf{C}_0})f_{\mathbf{C}}(x_{\mathbf{C}})}. \tag{2.22}$$

To reiterate, Equation 2.22 holds when $\mathcal{G}$ and $\mathcal{G}^*$ are both decomposable, such that $\mathcal{G}$ and $\mathcal{G}^*$ differ only by the presence of edge (a, b) in $\mathcal{G}$, and the priors are compatible strong hyper Markov laws.

Lastly, Equation 2.21 can be written

$$f(x) = \frac{\prod_{i=1}^k f_{\mathbf{C}_i}(x_{\mathbf{C}_i})}{\prod_{i=2}^k f_{\mathbf{S}_i}(x_{\mathbf{S}_i})}. \tag{2.23}$$

As we will be focusing on strong hyper Markov priors, we will use the short-hand notation

$$\prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} f_A(x_{\mathbf{A}}) = \frac{\prod_{i=1}^k f_{\mathbf{C}_i}(x_{\mathbf{C}_i})}{\prod_{i=2}^k f_{\mathbf{S}_i}(x_{\mathbf{S}_i})}. \tag{2.24}$$

where $\mathscr{C}$ and $\mathscr{S}$ are the cliques and separators of $\mathcal{G}$, assumed to be perfectly ordered. Additionally, this notation will be convenient for the priors and posteriors of both weak and strong hyper Markov laws.

2.4.3 Parametric Graphical Models

Up to now, the discussion of this chapter focused on Markov distributions and hyper Markov laws in general. If in addition we think that the data come from some parametric family then we restrict our attention to some family $\mathscr{F}$. Let $X \sim \theta \in \mathscr{F}$. A graph $\mathcal{G}$ of the conditional independence structure of X denotes the belief that θ is Markov with respect to $\mathcal{G}$. Thus, it restricts the model to a sub-family, $\mathscr{F}_{\mathcal{G}} = \mathscr{F} \cap \mathscr{M}(\mathcal{G})$. Graph selection is the problem of determining the smallest $\mathscr{F}_{\mathcal{G}}$ that contains θ. The most prevalent examples are graphical Gaussian models. Graph selection for Gaussian models is often called covariance selection. In this setting, the relevant family is the set of p-variate Gaussian distributions. Denote this family $\mathscr{N} = \{N_p(\mu, \Sigma) : \mu \in \mathbb{R}^p, \Sigma \in M_p^+\}$, where M_p^+ is the cone of real-valued, symmetric $p \times p$ matrices that are positive definite. Specifying a graph, $\mathcal{G}$, translates to putting constraints on Σ. Speed and Kiiveri (1986) showed that a sufficient statistic for the covariance matrix is the collection of sub-matrices for each clique. In other words, if there is no edge between nodes i and j, then σ_{ij} is a function of the other covariance elements. For example, if (x_1, x_2, x_3) is such that $x_1 \perp\!\!\!\perp x_3 | x_2$, then σ_{13} is no longer a free parameter, but a function of $\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}$, and σ_{23}. In general, denote the sub-family of Gaussian distributions Markov on $\mathcal{G}$ by $\mathrm{N}_{\mathcal{G}}$. Let $P_{\mathcal{G}}$ be the set of positive definite matrices such that if $K \in P_{\mathcal{G}}$, then $K_{ij} = 0$ for all $(i, j) \notin \mathbf{E}$; let $Q_{\mathcal{G}}$ be the image of $P_{\mathcal{G}}$ under matrix inversion. If $N_p(\mu, \Sigma_{\mathcal{G}})$ is Markov with respect to $\mathcal{G}$, then $\Sigma_{\mathcal{G}} \in Q_{\mathcal{G}}$. The goal of covariance selection is to find the smallest $Q_{\mathcal{G}}$ containing Σ, the population covariance matrix.

Much progress has been made with graph selection for parametric models. Dawid and Lauritzen (1993) proved many results for decomposable graphical models, including multinomial and multivariate Gaussian problems. For example, they present the distribution of the restricted maximum likelihood estimate of Σ for the $N_{\mathcal{G}}$ model with μ known. This distribution is called the hyper Wishart distribution since the marginal law for each clique is Wishart. Several others have run with this idea. Letac and Massam (2007) have extended the hyper inverse Wishart to a richer family of distributions on $Q_{\mathcal{G}}$ and $P_{\mathcal{G}}$. Giudici and Green (1999) implemented a reversible jump Markov chain Monte Carlo algorithm for determining $\mathcal{G}$. Carvalho et al. (2007) provide an algorithm for generating random variables from this family. For decomposable models, the process is simplified by the presence of a perfect ordering. For two cliques, the algorithm begins by generating an inverse Wishart variable on one clique. If the cliques overlap, this determines some of the parameters for the other clique. Therefore, one needs to generate a conditional Wishart variable given those entries. For multiple cliques, one simply repeats this process. With a perfect ordering, the process is simpler because each new clique is conditioned on only one previous clique. Conditioning on multiple cliques can lead to moderate complications in the conditional distribution, because the conditioning set is incomplete. Hence, decomposable models are computationally convenient.

If we wish to impose fewer constraints on the model, then we may consider a nonparametric family for the graphical model. Most work to date has focused on parametric models. Specifically, we refer to the generalized hyper Dirichlet models for analyzing contingency tables (Letac and Massam, 2007; Dobra and Massam, 2009) and graphical Gaussian

models (Giudici and Green, 1999). The major focus of this dissertation is to provide the mechanism for extending the great amount of knowledge on graphical models to allow one to choose non-parametric hyper Markov processes. In particular, we focus on a hyper Markov version of the Dirichlet process, which we dub a *hyper Dirichlet process*.

Chapter 3

Non-Parametric Processes

A parametric model constrains the space of available functions or distributions to lie within a finite-dimensional subspace. For example, a (global) linear regression projects the function $\mathbb{E}(Y|X = x)$ to the space of linear functions. Such constraints are acceptable only if one believes the function is "close" to this space in some sense. In this case, the loss of information may be outweighed by the benefit of a simpler structure. On the other hand, if one lacks a good idea of the structure of a distribution, a parameterized space can lack the flexibility to find a suitable estimate. By contrast, a *non-parametric family* can closely approximate any arbitrary distribution (van Belle et al., 2004). The best-known example of a non-parametric family may be the Dirichlet process laws (Ferguson, 1973). The Dirichlet process is convenient due to several "neutrality" properties, which we discuss further in this chapter. Briefly, if $B \subseteq A$ are measurable sets, then $P(B|A) \perp\!\!\!\perp P(A)$. In other words, the probability of any measurable set provides no information about how that mass

is distributed among its subsets. Likewise, the probability of any measurable set provides no information about how the remaining mass is distributed among any disjoint sets.

3.1 The Dirichlet Distribution

The Dirichlet process is constructed to have Dirichlet distributions for all finite-dimensional distributions. Many of the Dirichlet process' convenient properties stem from this fact. It is therefore instructive to gain an understanding of the Dirichlet distribution and its many beautiful properties.

The Dirichlet distribution is a probability distribution of vectors $\vec{X} = (X_1, \ldots, X_n)$ such that the $X_i's$ are non-negative and sum to unity. Geometrically speaking, we say that $(X_1, \ldots, X_n)$ exists on the $(n-1)$-dimensional simplex. For this reason, the Dirichlet distribution can be used as a prior for a discrete random variable, where $\mathbb{P}(Y = i|\vec{X} = \vec{x}) = x_i$. It is the conjugate prior for the family of multinomial vectors.

Definition 3.1.1 (Dirichlet Distribution)**.** *Let $\alpha_1, \ldots, \alpha_n$ be non-negative real numbers, where at least one α_i is strictly positive. The Dirichlet distribution, Dir$(\alpha_1, \ldots, \alpha_n)$, has density*

$$d\text{Dir}(x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n) = \frac{\Gamma(\alpha_1 + \alpha_2 + \ldots + \alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\ldots\Gamma(\alpha_n)} x_1^{\alpha_1 - 1} x_2^{\alpha_2 - 1} \ldots x_n^{\alpha_n - 1}, \tag{3.1}$$

if each x_i is non-negative and $\sum_{i=1}^{n} x_i = 1$. In all other cases, the density is 0.

3.1.1 Representations of Dirichlet Random Vectors

The Dirichlet distribution has various representations in terms of smaller-dimensional Gamma, Beta, and Dirichlet random variables. The Gamma and Beta representations are well-known. The representation of a Dirichlet random vector in terms of smaller Dirichlet vectors is possibly widely believed, but it is infrequently stated. We first note that the Dirichlet distribution is a generalization of the Beta distribution. If $x \sim \text{Beta}(\alpha_1, \alpha_2)$ then it is easily seen that $(x, 1-x) \sim \text{Dir}(\alpha_1, \alpha_2)$ by comparing the density functions. It is well-known that Beta distribution can be represented in terms of Gamma random variables. If $\psi_1 \sim \text{Gamma}(\alpha_1, \beta)$ and $\psi_2 \sim \text{Gamma}(\alpha_2, \beta)$, then $\psi_1/(\psi_1 + \psi_2) \sim \text{Beta}(\alpha_1, \alpha_2)$. This representation can be generalized for the Dirichlet distribution.

Dirichlet Property 3.1.2 (Gamma Representation)**.** *Let $\alpha_1, \ldots, \alpha_n$ be a sequence of non-negative real numbers, where at least one α_i is strictly positive. For each i, let $\psi_i \sim Gamma(\alpha_i, 1)$, where $Gamma(0,1) = \delta_0$. If $X_i = \psi_i / \left(\sum_{j=1}^n \psi_j\right)$, then*

$$(X_1, \ldots, X_n) \sim \text{Dir}(\alpha_1, \ldots, \alpha_n). \tag{3.2}$$

The Gamma representation implies that for each $i, X_i = 0$ if and only if $\alpha_i = 0$ with probability one.

Gamma random variables are additive, such that if $\psi_1 \sim \text{Gamma}(\alpha_1, \beta)$, and $\psi_2 \sim \text{Gamma}(\alpha_2, \beta)$, then their sum has distribution $\text{Gamma}(\alpha_1 + \alpha_2, \beta)$. As a result, the Dirichlet distribution also has an additivity property. In the Gamma representation, we see that the Dirichlet vector is the vector of proportions of each Gamma random variable to the total. Consider the distribution of the vector $(Y = X_1 + X_2, X_3, \ldots, X_n)$. Mathematically,

the ratio of $\psi_1 + \psi_2$ to the grand total is the sum of their individual ratios. Thus, we can generate $(Y, X_3, \ldots, X_n)$ by generating $\psi_1, \ldots, \psi_n$, and setting $Y = \psi_1 + \psi_2$. Of course, this is equivalent to generating $\psi_1 + \psi_2 \sim \text{Gamma}(\alpha_1 + \alpha_2, 1)$ directly. Hence, we see that $(Y, X_3, \ldots, X_n) \sim \text{Dir}(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_n)$. The next property is a generalization of this idea.

Dirichlet Property 3.1.3 (Additivity)**.** *Suppose* $(X_1, \ldots, X_n) \sim$*Dir*$(\alpha_1, \ldots, \alpha_n)$*. Choose any* $0 = r_0 < r_1 < \ldots < r_k = n$*.*

$$\left(\sum_{i=r_0}^{r_1} X_i, \sum_{i=r_1+1}^{r_2} X_i, \ \ldots, \ \sum_{i=r_{k-1}+1}^{r_k} X_i \right) \sim \text{Dir} \left(\sum_{i=r_0}^{r_1} \alpha_i, \sum_{i=r_1+1}^{r_2} \alpha_i, \ \ldots, \ \sum_{i=r_{k-1}+1}^{r_k} \alpha_i \right) \quad (3.3)$$

The major implication of this property is seen in Section 3.2 where it ensures that the Dirichlet process is consistent in the sense that marginal values do not depend on how you compute them. The additivity of Dirichlet processes also tells us the marginal distribution for each component. For example, $(X_1, X_2 + \ldots + X_n) \sim$Dir$(\alpha_1, \alpha_2 + \ldots + \alpha_n)$, which we know is equivalent to a Beta random variable. Now consider the conditional distribution of X_2 given $X_1 = x_1$. Specifically, $X_2/(1 - X_1) = \gamma_2/(\gamma_2 + \cdots + \gamma_n)$ is another Beta random variable. This leads to a component-wise construction of the Dirichlet distribution via its Beta marginals. (As a technical note, the Beta$(0, 0)$ definition is ill-defined, but our choice of definition is immaterial. To see this, suppose $\alpha_i = \beta_i = 0$ and we have chosen the smallest i with this property. In this case, $\alpha_{i-1} > \beta_{i-1} = 0$ which implies that $\phi_{i-1} = 1$. Therefore $\sum_{j=1}^{i-1} X_j = 1$, so $X_j = 0$ for all $j \geq i$.)

Dirichlet Property 3.1.4 (Beta Representation)**.** *A Dirichlet random vector can be generated one element at a time. Let* $\alpha_1, \ldots, \alpha_n$ *be non-negative real numbers where at least one*

α_i is strictly positive. Let $\beta_i = \sum_{j=i+1}^{n} \alpha_j$. Let $X_1 = \phi_1 \sim Beta(\alpha_1, \beta_1)$. For $1 < i < n$, let $\phi_i \sim Beta(\alpha_i, \beta_i)$, and set $X_i = \phi_i(1 - \sum_{j=1}^{i-1} X_j)$. Let $X_n = 1 - \sum_{i=1}^{n-1} X_i$. Then $(X_1, \ldots, X_n) \sim Dir(\alpha_1, \ldots, \alpha_n)$.

The first step is obvious, since $X_1 \sim \text{Beta}(\alpha_1, \beta_1)$. The remaining components can be understood by applying the Gamma representation. Note that $\phi_2 = X_2/(1 - X_1)$. Using the Gamma representation, we can write $\phi_2 = \psi_2/(\sum_{i=2}^{n} \psi_i)$. Thus, it is the marginal for a $(n-1)$-dimensional Dirichlet vector, and has distribution $\text{Beta}(\alpha_2, \beta_2)$. Likewise, $\phi_3 = X_3/(1 - X_1 - X_2) \sim \text{Beta}(\alpha_3, \beta_3)$. This algorithm continues until we have determined $X_1, \ldots, X_{n-1}$. Since the support of the Dirichlet distribution is the $(n-1)$-simplex, we must have $X_n = 1 - \sum_{i=1}^{n-1} X_i$.

Alternatively, we can think about a Dirichlet vector as dividing a stick or interval of unit length into randomly-sized pieces. We begin by breaking of a fraction, ϕ_1 and assigning this to X_1. Next, we break off a fraction, ϕ_2, of the remaining stick and assign this to X_2. Thus, $X_2 = \phi_2(1 - \phi_2)$. After i breaks, the length of the remaining piece is $\prod_{j=1}^{i}(1 - \phi_j)$. Note that this quantity is equal to $1 - \sum_{j=1}^{i} X_i$, so this method is a simple restatement of the Beta Representation. This representation will resurface in Section 3.4.1 when we talk about stick-breaking processes.

Rather than break off one piece at a time, a second way to divide an interval is to divide it into two smaller intervals and then divide each of these into pieces. More generally, we consider the stick-breaking process as a two-phase process. In the initial phase, we divide the stick into n smaller pieces. In the second phase we sub-divide each piece. This

two-phase process is equivalent to the simpler stick-breaking scheme assuming a certain correspondence between the parameters. If we consider each division as a Dirichlet random variable, this leads to a new characterization of the Dirichlet distribution in terms of smaller-dimensional Dirichlet random variables.

Dirichlet Property 3.1.5 (Dirichlet Representation)**.** *Suppose* $(w_1 \dots w_k) \sim Dir(a_1 \dots a_k)$ *and for* $1 \leq i \leq k, (v_{i1} \dots v_{in}) \sim Dir(a_{i1} \dots a_{in})$*, where the vectors are mutually independent and* $a_i = \sum_{j=1}^{n} a_{ij}$ *for each* i*. If* $x_{ij} = w_i v_{ij}$*, then* $(x_{11} \dots x_{kn}) \sim Dir(a_{11} \dots a_{kn})$*.*

Proof. This proof is a straight-forward, though messy application of the pdf-method for changes of variables. Fortunately, most of the book-keeping is involved in finding the determinant of the transformation, which is already done by Lemma A.0.1. For $i < k$, define $s_i = \sum_{j=1}^{n} x_{ij}$. Note that some of the variables specified are not free. These are w_k, x_{kn}, and for each i, v_{in}. Therefore, we express the inverse transformation without reference to them.

$$w_i = s_i, \quad \text{for } i < k \tag{3.4}$$

$$v_{ij} = x_{ij}/s_i, \quad \text{for } i < k, j < n \tag{3.5}$$

$$v_{kj} = x_{kj}/(1 - \sum_i s_i) \tag{3.6}$$

Using this notation, we now express the partial derivatives. For w_i $(i < k)$, the partial derivatives are

$$\frac{\partial w_i}{\partial x_{\ell m}} = \begin{cases} 1, & \ell = i \\ 0, & \ell \neq i \end{cases} . \tag{3.7}$$

For v_{kj}, the partial derivatives are

$$\frac{\partial v_{kj}}{\partial x_{\ell m}} = \begin{cases} (1 - \sum_i s_i)^{-1}, & (k,j) = (\ell, m) \\ x_{kj}/(1 - \sum_i s_i)^{-2}, & \ell < k \end{cases} . \tag{3.8}$$

For the other v's, such that $i < k$ and $j < n$, the partial derivatives are

$$\frac{v_{ij}}{\partial x_{\ell m}} = \begin{cases} (s_i - x_{ij}/(s_i)^2, & \ell = i, m = j \\ -x_{ij}/(s_i)^2, & \ell = i, m \neq j \\ 0, & o.w. \end{cases} . \tag{3.9}$$

The Jacobian is an $nk - 1$ square matrix, but it has a rather simple block structure. Define $J_k = (1 - \sum_i s_i)^{-1} I_{n-1}$, where I_{n-1} is the $(n-1)$-identity matrix. For $i < k$, define

$$J_i = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \frac{s_i - x_{i1}}{s_i^2} & \frac{-x_{i1}}{s_i^2} & \cdots & \frac{-x_{i1}}{s_i^2} & \frac{-x_{i1}}{s_i^2} \\ \frac{-x_{i2}}{s_i^2} & \frac{s_i - x_{i2}}{s_i^2} & \cdots & \frac{-x_{i2}}{s_i^2} & \frac{-x_{i2}}{s_i^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-x_{i,n-1}}{s_i^2} & \frac{-x_{i,n-1}}{s_i^2} & \cdots & \frac{s - x_{i,n-1}}{s_i^2} & \frac{-x_{i,n-1}}{s_i^2} \end{pmatrix} . \tag{3.10}$$

Finally, define

$$K = \frac{1}{1 - \sum_{i=1}^{k-1} s_i} \cdot \begin{pmatrix} x_{k1} & \cdots & x_{k1} \\ \vdots & \vdots & \vdots \\ x_{k,n-1} & \cdots & x_{k,n-1} \end{pmatrix}. \tag{3.11}$$

Using this notation, the Jacobian for the inverse transformation is

$$J = \begin{pmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots 0 & \\ 0 & 0 & \ddots & 0 \\ K & K & \cdots & J_k \end{pmatrix}. \tag{3.12}$$

A well-known fact from linear algebra is that this matrix has determinant $|J| = \prod_{i=1}^{k} |J_i|$.

Lemma A.0.1 provides all the necessary determinants:

$$|J_k| = (1 - \sum_{i=1}^{k-1} s_i)^{-(n-1)} \tag{3.13}$$

$$|J_i| = (-1)^{n-1} s_i^{-(n-1)} \tag{3.14}$$

$$||J|| = \prod_{i=1}^{k-1} (1 - s_i)^{-(n-1)} \cdot (1 - \sum_{i=1}^{k-1} s_i)^{-(n-1)} \tag{3.15}$$

We are now ready to calculate the pdf of $(X_{11} \ldots X_{nk})$.

$$
\begin{aligned}
f_X(\vec{x}) &= f_{W,V}(\vec{w}(\vec{x}), \vec{v}(\vec{x}) \cdot ||J|| && (3.16)\\
&= f_W(\vec{w}(\vec{x})) \times f_V(\vec{v}(\vec{x})) \times ||J|| && (3.17)\\
&= \prod_{i=1}^{k-1} s_i^{a_i-1}(1-\sum_{i=1}^{k-1} s_i)^{a_k-1} \\
&\quad \times \prod_{i=1}^{k-1}\prod_{j=1}^{n}\left(\frac{x_{ij}}{s_i}\right)^{a_{ij}-1} \prod_{j=1}^{n}\left(\frac{x_{kj}}{1-\sum_{i=1}^{k-1} s_i}\right)^{a_{kj}-1} \\
&\quad \times \prod_{i=1}^{k-1} s_i^{-(n-1)}\left(1-\sum_{i=1}^{k-1} s_i\right)^{-(n-1)} && (3.18)\\
&= \prod_{i=1}^{k-1} s_i^{a_i-1-(\sum_{j=1}^{n} a_{ij}-n)-n+1}(1-\sum_{i=1}^{k-1} s_i)^{a_k-1-(\sum_{k=1}^{n} a_k j-n)-n+1} \\
&\quad \times \prod_{i=1}^{k}\prod_{j=1}^{n} x_{ij}^{a_{ij}-1} && (3.19)\\
&= \prod_{i=1}^{k-1} s_i^{a_i-\sum_{j=1}^{n} a_{ij}}(1-\sum_{i=1}^{k-1} s_i)^{a_k \sum_{k=1}^{n} a_k j} \cdot \prod_{i=1}^{k}\prod_{j=1}^{n} x_{ij}^{a_{ij}-1} && (3.20)\\
f_X(\vec{x}) &\propto \prod_{i=1}^{k}\prod_{j=1}^{n} x_{ij}^{a_{ij}-1} \qquad \square && (3.21)
\end{aligned}
$$

The penultimate line emphasizes the necessity that $a_i = \sum_{j=1}^{n} a_{ij}$. It is merely a notational convenience that $\vec{v}_i$ has k components for each i. This constraint can be dropped without breaking the backbone of the proof.

3.1.2 Neutrality and the Generalized Dirichlet Distribution

Implicit in the Beta representation of the Dirichlet distribution is that $X_2/(1-X_1)$ is independent of X_1. In fact, X_1 is independent of the entire vector $(X_2/(1-X_1), \ldots, X_n/(1-X_1)$. In the language of Connor and Mossiman (1969), we say that X_1 is *neutral*. In

other words, the value of X_1 does not affect how the remaining $(1 - X_1)$ is divided among the remaining components. Continuing, (X_1, X_2) is independent of $(X_3/(1 - X_1 - X_2), \ldots, X_n/(1 - X_1 - X_2)$ and so on. In general, at each step $i + 1$, the proportions $(X_1, \ldots, X_i)$ are independent of the relative values of the other $n - i$ proportions. This property is expressed by the following definition.

Definition 3.1.6 (Completely Neutral Random Vector)**.** *A random vector* $(X_1, \ldots, X_n)$ *is* completely neutral *if for all* $i < n$*,*

$$(X_1, \ldots, X_i) \perp\!\!\!\perp \left(\frac{X_{i+1}}{1 - \sum_{j=1}^{i} X_j}, \ldots, \frac{X_n}{1 - \sum_{j=1}^{i} X_j} \right) \tag{3.22}$$

A Dirichlet random vector is completely neutral regardless of the ordering of its components. This is easy to see because the Dirichlet density is symmetric among the α_i's. For general random vectors this is not the case. In fact, Mossiman (1962) shows that, if we exclude cases for which some of the X_i's are degenerate, then the Dirichlet distribution is the *only* distribution where this is the case. The Dirichlet representation (Dirichlet Property 3.1.5) shows that Dirichlet random vectors have an even deeper level of neutrality: for any subset of X_i's, their total value is independent of how that total is divided among that subset. Mathematically, for any $\mathbf{A} \subseteq \{1, \ldots, n\}$, $T_{\mathbf{A}} \perp\!\!\!\perp \{X_i/T_{\mathbf{A}} : i \in \mathbf{A}\}$, where $T_{\mathbf{A}} = \sum_{i \in \mathbf{A}} X_i$. Heuristically, this property may be implied by complete neutrality for all permutations, as the Dirichlet distribution is the only probability measure which satisfies either one.

Using the idea of neutrality, Connor and Mossiman (1969) develop a *generalized Dirichlet distribution* for random vectors that are completely neutral only for one ordering of components.

Definition 3.1.7 (Generalized Dirichlet Distribution)**.** *Let $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n$ positive real numbers. The generalized Dirichlet distribution, GD$(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n)$, has density*

$$d\mathrm{GD}(\vec{x}; \vec{\alpha}, \vec{\beta}) = \left[\prod_{i=1}^{n-1} B(\alpha_i, \beta_i)\right]^{-1} x_n^{\beta_{n-1}-1} \prod_{i=1}^{n-1} \left[x_i^{\alpha_i - 1} \left(\sum_{j=i}^{n} x_j\right)^{\beta_{i-1}-(\alpha_i+\beta_i)}\right], \qquad (3.23)$$

where B is the beta function, if each x_i is non-negative and $\sum_{i=1}^{n} x_i = 1$. In all other cases, the density is 0*.*

Recall that we can construct a Dirichlet random vector by successive Beta random variables, where $Z_i = X_i/(1 - \sum_{j<i} X_j) \sim \mathrm{Beta}(\alpha_i, \sum_{j>i} \alpha_j)$. The generalized Dirichlet has a similar construction in which $Z_i \sim \mathrm{Beta}(\alpha_i, \beta_i)$. It is easily seen that if $\beta_{i-1} = \alpha_i + \beta_i$ then we recover the Dirichlet distribution. Tying it back to the Beta representation, this constraint is equivalent to $\beta_{n-1} = \alpha_n$, and for $i < n$, $\alpha_i = \sum_{j>i} \alpha_j$.

3.2 The Dirichlet Process

The Dirichlet process is a much-publicized non-parametric process formally introduced by Ferguson (1973). It is a prior law over probability distributions whose finite-dimensional marginals have a Dirichlet distribution. Dirichlet processes have been used for modeling

Gaussian mixtures when the number of components is unknown (Escobar and West, 1995; Rasmussen, 1999), hidden Markov models with infinite state spaces (Beal et al., 2001), and evolutionary clustering in which both data and clusters come and go as time progresses (Xu et al., 2008).

Definition 3.2.1 (Dirichlet Process (Ferguson, 1973))**.** *Let α be a measure over some measurable space $(\Theta, \mathcal{A})$, and let P be a random probability measure over the same space. We say that P is a Dirichlet process with base measure α, and write $P \sim DP_\alpha$, if*

$$(P(A_1), P(A_2), \ldots, P(A_k)) \sim \mathrm{Dir}(\alpha(A_1), \alpha(A_2), \ldots, \alpha(A_k)), \tag{3.24}$$

whenever $(A_i)_{i=1}^n$ is a finite partition of Θ.

Remark 3.2.1. *An alternative definition of the Dirichlet process comes from substituting α with νH, where $\nu = \alpha(\Theta)$ and $H = \overline{\alpha}$. In this case we say that $P \sim DP(\nu H)$ is a Dirichlet process with base distribution (or measure) H and precision ν.*

Immediately, several properties emerge from this definition. From Dirichlet Property 3.1.2, it is clear that $\mathbb{P}(A) = 0$ if and only if $\alpha(A) = 0$. In particular, we have the axiomatic $(\mathbb{P}(\Theta), \mathbb{P}(\emptyset)) = (1, 0)$ almost surely.

The Dirichlet process satisfies an important consistency property that is necessary for any reasonable distribution. Suppose $A_1, \ldots, A_k$ is a partition of Θ. For each, i, let $A_{i1}, \ldots, A_{in_i}$ be a partition of A_i. The collection $\{A_{ij}\}$ is a *refinement* of $\{A_i\}$. Since these sets are disjoint and have union A_i, we require that $\mathbb{P}(A_i) = \sum_{j=1}^{n_i} \mathbb{P}(A_{ij})$ almost surely. This is exactly Property 3.1.3.

The Dirichlet process also satisfies for any $A^* \subseteq A \subseteq X$, $\mathbb{P}(A^*|A) \perp\!\!\!\perp P(A)$. This is easily seen from the representation of a Dirichlet distribution by smaller-dimensional Dirichlet distributions and the fact that $P(A^*|A) = P(A^*)/P(A)$.

3.2.1 Stick-Breaking Representation of a Dirichlet Process

As Ferguson (1973) proved, the Kolmogorov extension theorem guarantees the existence of Dirichlet processes. Unfortunately, mere proof of existence does not imply a usable working definition. In fact, for twenty-one years, it was easier to generate a sample from an unknown random measure $P \sim DP_\alpha$ than to generate P itself! In 1994, Sethuraman finally published the first useful constructive definition of a Dirichlet process. We have already seen that a Dirichlet distribution can be constructed component-wise using Beta random variables. Sethuraman showed that this can be extended *ad infinitum* to include Dirichlet processes. The result is a stick-breaking process.

Theorem 3.2.2 (Stick-Breaking Definition of the Dirichlet Process)**.** *Let $(\phi_1, \phi_2, \ldots)$ be a sequence of independent random variables with distribution Beta$(1, \nu)$. Independent of this sequence, let $(Z_1, Z_2, \ldots)$ be a sequence of independent random variables with distribution H. Define $p_1 = \phi_1$, and for $i > 1$, define $p_i = \phi_i \prod_{j<i}(1 - \phi_j)$. The random measure $P = \sum_{i \in \mathbb{N}} p_i \delta_{Z_i}$ is a Dirichlet process with precision ν and base measure H.*

The proof is too long to reconstruct here, but we can get an intuitive sense of this definition. Recall from Section 3.1, that we can construct a Dirichlet vector from a sequence of independent Beta random variables. We can loosely imagine that the sequence

$(p_1, p_2, \ldots)$ is an infinite-dimensional extension of the Dirichlet distribution. Choose any finite partition, $(B_1, B_2, \ldots, B_n)$. We define $A_i = \{j : Z_j \in B_i\}$. Clearly, $(A_1, \ldots, A_n)$ is a partition of the natural numbers, with $P(B_i) = \sum_{j \in A_i} p_j$. If Property 2 of the Dirichlet Distribution applies to infinite-dimensional vectors, then we see that $(P(B_1), \ldots, P(B_n))$ is a Dirichlet vector.

The stick-breaking definition is important because it provides a constructive definition of a Dirichlet process. It also simplifies the proofs for some of the properties of the Dirichlet process. For example, it is trivial to see that the Dirichlet process gives probability one to the space of discrete distributions. The representation is especially important in the current work since we will use it to validate our method of constructing hyper Markov Dirichlet processes.

3.2.2 Independence Properties of the Dirichlet Process

It is well-known that a Dirichlet process on the real line is tail-free as defined by Freedman (1962) and Fabius (1964). A process is tail-free with respect to (s, ∞) if for all $t_0 = s < t_1 < \cdots < t_k$, $(F(t_1), \ldots, F(t_k)$ is completely neutral. Doksum (1974) shows a more general independence property that is applicable to Dirichlet processes on other spaces.

Let $\{\Pi_m : m = 0, 1, \ldots\}$ denote a sequence of nested, measurable partitions with $\pi_0\{\theta\}$. Denote Π_m by $\{A_{m,1}, \ldots, A_{m,l_m}\}$ and for $s < m$ define $l(i)$ such that $A_{m,i} \subseteq A_{s,l(i)}$.

Definition 3.2.3 (F-neutral)**.** *A stochastic process, P, is* F-neutral *with respect to the sequence $\{\Pi_m\}$ of nested, measurable partitions if there exist non-negative random variables*

$\{Z_{m,i} : i = 1, \ldots, k_m; m = 1, 2, \ldots\}$ *such that for each* $m \geq 1$ *the families* $\{Z_1, i : i = 1, \ldots, k_1\}, \ldots, \{Z_{m+1,i} : i = 1, \ldots, k_{m+1}\}$ *are independent and*

$$(P(A_{m,1}), \ldots, P(A_{m,k_m})) \stackrel{d}{=} (\prod_{s=1}^{m} Z_{s,l(1)}, \ldots, \prod_{s=1}^{m} Z_{s,l(k_m)}). \tag{3.25}$$

Definition 3.2.4 (Neutral)**.** *A stochastic process,* P*, is* neutral *with respect to the sequence* $\{\Pi_m\}$ *of nested, measurable partitions if for each* $m \geq 1$ *there exist non-negative independent random variables,* $V_{m,1}, \ldots, V_{m,k,}$ *with* $V_{m,k_m} = 1$ *and*

$$(P(A_{m,1}), P(A_{m,2}), \ldots, P(A_m, k_m)) \stackrel{d}{=} (V_{m,1}, V_{m,2}(1 - V_{m,1}), \ldots, V_{m,k_m} \prod_{j=1}^{k_m - 1} (1 - V_{m,j})). \tag{3.26}$$

The concept of F-neutrality refers to independence properties among partitions. It describes how the process is refined as the partitions become more detailed. Essentially, it means that when the value of the process is known at one level of refinement Π_s, the relative probabilities for the refined partition Π_{s+1} are independent. For example, the conditional distributions $(P(A_{s+1,i}|A_{s,l(i)}) : i = 1, \ldots, k_{s+1}))$ are independent of $(P(A_{s,1}), \ldots, P(A_{s,k_s}))$. Moreover, this independence property holds simultaneously at all partition levels: the families

$$\{P(A_{1,i})\}, \{P(A_{2,i}|A_{2,l(i)})\}, \ldots, \{P(A_{m+1,i}|A_{m,l(i)})\} \tag{3.27}$$

are jointly independent.

The concept of neutrality refers to independence properties within a partition. Basically, for all $m \geq 1$,

$$(P(A_{m,1}), P(A_{m,2}), \ldots, P(A_{m,k_m})), \tag{3.28}$$

is completely neutral.

The Dirichlet process is F-neutral and neutral for every sequence of measurable partitions. It is neutral because for any partition $\Pi_m, (P(A_{m,1}), \ldots, P(A_{m,k_m}))$ has a Dirichlet distribution by definition, which Connor and Mossiman (1969) have shown is completely neutral. To show F-neutrality, we first consider the case $m = 2$. In this case, $\{P(A_{1,i})\} \sim \text{Dir}(\{\alpha(A_{1,i})\})$, and for all $i \leq k_1$,

$$\{P(A_{2,j}|A_{2,l(j)}) : l(j) = i\} = \left\{ \frac{P(A_{2,j})}{P(A_{1,i})} : l(j) = i \right\} \sim \text{Dir}(\{\alpha(A_{2,j}) : l(j) = i\}). \quad (3.29)$$

Applying Property 3.1.5 of the Dirichlet distribution, we see

$$\{P(A_{2,j})\} = \left\{ \frac{P(A_{2,j})}{P(A_{1,l(j)})} \cdot P(A_{1,l(j)}) \right\} \sim \text{Dir}(\{\alpha(A_{2,j})\}). \quad (3.30)$$

It is clear that for any m, $\{P(A_{m,j}|A_{m-1,l(j)}\}$ is independent of $\{P(A_{s,i}) : s < m-1, i = 1, \ldots, k_s\}$; induction on m proves that the Dirichlet process is F-neutral.

The Dirichlet process is both F-neutral and neutral to all sequences of partitions. Furthermore, Doksum (1974) state that the Dirichlet process is essentially the only stochastic process with either type of neutrality with respect to all sequences. Define the class C_1 consisting of processes which are (i) degenerate at a given probability distribution, (ii) concentrated at a random point, or (iii) concentrated on two non-random points. Dirichlet processes are the only stochastic processes that are not in C_1 that are F-neutral to all sequences of measurable partitions. They are also the only stochastic processes that are not in C_1 that are neutral to all sequences of partitions. Because a Dirichlet process is F-neutral to any partition, the posterior distribution of $P(A)$ for any $A \in \mathcal{F}$ depends only

on the number of observations which fall in A and not on where they fall inside or outside of A. Furthermore, Dirichlet processes are the only stochastic processes not in C_1 for which this is true.

3.2.3 Sampling from Dirichlet Processes

We turn our attention to samples from Dirichlet processes. We employ a hierarchical model, $P \sim DP_\alpha$ and given P, $\theta_1, \ldots, \theta_n$ are independent draws from P. Antoniak (1974) uses the more formal definition:

Definition 3.2.5 (Sample from a Dirichlet Process)**.** *Let P be a Dirichlet process on $(\Theta, \mathcal{A})$. We say that $\theta_1, \ldots, \theta_n$ is a sample of size n from P if for any $m \in \mathbb{N}$, and measurable sets $A_1, \ldots, A_m, C_1, \ldots, C_n$,*

$$\mathbb{P}(\theta_1 \in C_1, \ldots, \theta_n \in C_n | P(A_1), \ldots, P(A_m), P(C_1), \ldots, P(C_n)) = \prod_{i=1}^{n} P(C_i) \text{ a.s.} \quad (3.31)$$

We next discuss the marginal distribution of a sample from a Dirichlet process. The next theorem concerns the special case $n = 1$.

Theorem 3.2.6. *Let θ be a sample of size 1 from $P \sim DP_\alpha$. The marginal distribution of θ and the expectation of P are both $\overline{\alpha}$.*

Proof. For any measurable set A, $\mathbb{P}(\theta \in A) = \int_0^1 P(A) dP(A)$, where $P(A) \sim \text{Beta}(\alpha(A), \alpha(A^c))$. Therefore, $\mathbb{P}(\theta \in A)$ is the expectation of a Beta random variable which is well-known to be $\alpha(A)/(\alpha(A) + \alpha(A^c)) = \overline{\alpha}(A)$. The same argument shows $\mathbb{E}(P) = \overline{\alpha}$. □

We note that for general samples of size n, $\theta_1, \ldots, \theta_n$ are no longer independent but they are exchangeable. In particular, for $n = 2$ we have $\mathbb{P}(\theta_2 \in A_2|\theta) = \int_0^1 P(A_2)dP_{\theta_1}(A_2)$, where $P_{\theta_1}(A_2)$ is the posterior distribution of $P(A_2)$ given θ_1 and is given by the next theorem.

Theorem 3.2.7. *Let $\theta_1, \ldots, \theta_n$ be a sample of size n from $P \sim DP_\alpha$. The posterior distribution of P is DP with a base measure of $\alpha + \sum_{i=1}^n \delta_{\theta_i}$.*

Proof. For a completely formal proof see Schervish (1995). We simply point out that the following relationship holds for partitions with arbitrarily small sets. For any set A, let $N(A) = \sum_{i=1}^n \delta_{\theta_i}(A)$ be the number of draws in A. Let $A_1, \ldots, A_k$ be a measurable partition of Θ. $(P(A_1), \ldots, P(A_k)) \sim \text{Dir}(\alpha(A_1), \ldots, \alpha(A_k))$ and $(N(A_1), \ldots, N(A_k))$ is a multinomial draw conditional on the vector of probabilities. It is well-known that the posterior distribution of $(P(A_1), \ldots, P(A_k))$ in this model is $\text{Dir}(\alpha(A_1)+N(A_1), \ldots, \alpha(A_k)+N(A_k))$. We point out that $\alpha(A_j) + N(A_j) = [\alpha + \sum_{i=1}^n \delta_{\theta_i}](A_j)$. □

Returning to a sample of size $n = 2$ from $P \sim DP_\alpha$, we see that $P(\theta_2 \in A_2|\theta_1) = (\alpha(A_2) + \delta_{\theta_1}(A_2))/\alpha(\Theta) + 1)$. In other words, the conditional distribution of θ_2 is the mixture $p\overline{\alpha} + q\delta_{\theta_1}(A_2)$, where $p = \alpha(\Theta)/(1 + \alpha(\Theta))$ and $q = 1/(1 + \alpha(\Theta))$.

We now turn to the problem of practically sampling from a Dirichlet process. Sethuraman (1994) noted that even with his constructive definition, it is impossible to fully sample P because it is a weighted average of an infinite number of atoms. Regardless, we can truncate the stick-breaking process at any finite sequence leaving the last $1 - \sum_{i=1}^n p_i$ mass unknown. This allows us to specify a distribution P^* which is as close to P as

needed. In particular, we *can* sample the random outcomes $\theta_1, \ldots, \theta_n$ from the exact distribution P, even though we cannot generate P exactly. Suppose for now that we knew the stick-breaking weights $p_1, p_2, \ldots$ and atoms $Z_1, Z_2, \ldots$ as defined in Equation 3.2.2. Let $U_1, \ldots, U_n$ be independent uniform random variables on $[0, 1)$. For each $i \leq n$, define $r_i = \min\{m : \sum_{j=1}^{m} p_j < U_i\}$ and set $\theta_i = Z_{r_i}$. We see that $\theta_1, \ldots, \theta_n$ is a sample of size n from P, but that we did not require knowledge of (p_j, Z_j) for $j > \max\{r_i\}$. Therefore, we could also generate the sample by beginning with $U_1, \ldots, U_n$ and carrying out the stick-breaking construction until $\sum_j p_j \geq \max\{U_i\}$.

The stick-breaking construction is useful because we can generate a distribution P^* that agrees with $P \sim DP_\alpha$ except on a set U satisfying $P(U) < \epsilon$. We can do so by continuing the stick-breaking construction until $1 - \sum_j p_j < \epsilon$.

We now describe the method of Blackwell and MacQueen (1973) which generates P^* and $\theta_1, \ldots, \theta_n$ using a Pólya urn. While the stopping criterion is less straightforward, the method is important both because historically and theoretically because it very clearly shows the predictive distribution of θ_{n+1}. (Sethuraman (1994) also proves the predictive distribution using his stick-breaking construction, but the urn scheme is more straightforward in this area.)

Sampling from a Dirichlet Processes via a Pólya Urn Scheme

Pre-dating the stick-breaking process, Blackwell and MacQueen (1973) provided a method of sampling from a Dirichlet process using a Pólya urn scheme. They define the following:

Definition 3.2.8 (Pólya Sequence)**.** *Let α be a finite positive measure on a separable space $(\Theta, \mathcal{A})$. A sequence of random variables, $\{\theta_n : n \geq 1\}$ is a Pólya sequence with parameter α if for every $U \in \mathcal{A}$,*

$$P(\theta_1 \in U) \sim \alpha(U)/\alpha(\Theta) \tag{3.32}$$

$$P(\theta_{n+1} \in U|\theta_1, \ldots, \theta_n) \sim \alpha_n(U)/\alpha_n(\Theta), \tag{3.33}$$

where $\alpha_n = \alpha + \sum_{i=1}^n \delta_{\theta_i}$.

For finite n, a Pólya sequence represents a sequence of draws from an urn that begins with $\alpha(\theta)$ balls of color θ where after each draw, the chosen ball is replaced and another ball of the same color is added.

Blackwell and MacQueen (1973) prove that the measure $\alpha_n = \alpha + \sum_{i=1}^n \delta_{\theta_i}$ converges to a random limiting distribution P whose distribution is a DP_α. They also show that given P, the variables $\theta_1, \ldots, \theta_n$ are independent with distribution P. In other words, the Pólya urn scheme represents the hierarchical model $P \sim DP_\alpha$, and $\theta_1, \ldots, \theta_n|P \sim P$. The Pólya urn scheme nicely illustrates the posterior distribution of P given θ_1. After θ_1 is drawn, it is clear that the sequence $\theta_2, \theta_3, \ldots$ is a Pólya sequence starting with $\alpha(\theta_1) + 1$ balls of color θ_1 and $\alpha(\theta)$ balls of color θ for $\theta \neq \theta_1$. Therefore, the limiting distribution of P given θ_1 is a Dirichlet process with measure $\alpha + \delta_{\theta_1}$. Furthermore, we see that the distribution of θ_{n+1} given $\theta_1, \ldots, \theta_n$ is $\alpha + \sum_{i=1}^n \delta_{\theta_i}$.

Illustration: Chinese Restaurant Process

The Pólya urn scheme shows that there is a positive probability that some balls will be of the same color. After the n^{th} draw, the colors of the balls provide a random partition of $\{1, \ldots, n\}$; the sets are defined by groups of balls that have the same color θ. This fact is famously illustrated by the Chinese restaurant process. The first person sits at a table, and orders a dish, which is chosen randomly with distribution H. The second person will join his table with probability $1/(1+\nu)$, otherwise he will sit at a new table, and order a dish according to probability H. In general, after n people have been seated, the next customer will choose to sit at a new table with probability $\nu/(n+\nu)$. Otherwise, he will join one of the existing tables with probabilities proportional to the number of people at the table already. On a technical note, we will assume that this particular restaurant has an infinite number of tables, and that each table seats an infinite number of people. If θ_i represents the i^{th} person's dish, then it is easy to see that

$$F(\theta_{n+1}|\nu, H, \theta_1, \ldots, \theta_n) \sim \sum_{j=1}^{n} \frac{1}{n+\nu}\delta_{\theta_j} + \frac{\nu}{n+\nu}H. \tag{3.34}$$

Let $\vec{t} = \{t_1, \ldots, t_n\}$ denote the table at which the i^{th} customer is sitting. For each $i \leq n$, define m_i as follows. If $i = \min\{j : t_j = t_i\}$ then the i^{th} customer is the first person at their table and we set $m_i = \nu$, otherwise $m_i = |\{j < i : t_j = t_i\}|$ is the number of previous customers at the table. The distribution of a particular seating arrangement is

$$P(\vec{t} = \{t_1, \ldots, t_n\}) = \prod_{i=1}^{n} \frac{m_i}{\nu + i - 1}, \tag{3.35}$$

or if we pull out the denominators and group the numerators by table,

$$P(\vec{t} = \{t_1, \ldots, t_n\}) = \left(\prod_{j=1}^{k} \prod_{i:t_i=j} m_i \right) \prod_{i=1}^{n} (\nu + i - 1)^{-1}, \tag{3.36}$$

where $k = \max\{t_i \in \vec{t}\}$ is the number of occupied tables.

Consider a particular table, say j. For the first person to arrive at this table, $m_i = \nu$. For the second person, $m_i = 1$. For the third, $m_i = 2$, and so on. Therefore, $\prod_{i:t_i=t_j} m_i = \nu \cdot \prod_{i=1}^{n_j - 1} i$, where n_j is the number of customers at table j. Equation 3.36 is equivalent to

$$P(\vec{t} = \{t_1, \ldots, t_n\}) = \nu^k \left(\prod_{j=1}^{k} \prod_{i=1}^{n_j - 1} i \right) \prod_{i=1}^{n} (\nu + i - 1)^{-1}. \tag{3.37}$$

Thus, the number of number of tables, k, and the number of people at each table $(n_1, \ldots, n_k)$, are sufficient statistics for the seating arrangement. Furthermore, conditional on the seating configuration, the dishes are a sequence of k independent and identically distributed variables with distribution H. This shows that the sequence $(\theta_1, \ldots, \theta_n)$ is exchangeable.

The correspondence between a Chinese restaurant process and a Pólya urn scheme is evident. We begin with ν balls in the urn. The distribution of X_1 is $\overline{\alpha} = H$. After the first n draws, the ν original balls are mixed with n additional balls with colors matching X_1 through X_n. With probability $\nu/(n + \nu)$ the next draw will select one of the original balls and X_{n+1} will have distribution H. On the other hand, we may draw the ball representing X_i with probability $1/(n + \nu)$. In this event, $X_{n+1} = X_i$. Therefore, the distribution of X_{n+1} is a mixture of H and n atomic distributions which is given in Equation 3.34.

3.3 Mixtures of Dirichlet Processes

Antoniak (1974) developed properties for mixtures of Dirichlet processes. Notably, these mixtures arise as posterior distributions when a Dirichlet process is sampled with noise. Basically, we consider an index set $(\mathcal{U}, \mathcal{B}, H^*)$, where the probability measure H^* is called a *mixing measure*. Conditional on u, P is a Dirichlet process on some space $(\Theta, \mathcal{A})$ with base measure $\alpha(u, \cdot)$. In general $\alpha(u, \Theta)$ may not be constant for $u \in U$.

We have already seen that if θ is a sample from $P \sim DP_\alpha$, then the posterior distribution of P is $DP_{\alpha+\delta_\theta}$. If we know only that θ lies in some set A, then the resulting conditional for P is a mixture of Dirichlet processes. Specifically, let P be a Dirichlet process on $(\Theta, \mathcal{A})$ and let $\theta|P \sim P$. The conditional distribution of P given $\theta \in A$ is a mixture of Dirichlet processes with index set $(A, \mathcal{A} \cap A)$, mixing distribution $H_A(\cdot) = \frac{\alpha(\cdot)}{\alpha(A)} \mathbf{1}_A(\cdot)$, and $\alpha(u, \cdot) = \alpha + \delta_u$. In other words, the conditional distribution for P is a mixture of the posteriors $P|\theta = u$, weighted by the conditional distribution of θ given $\theta \in A$.

A similar result occurs when θ is sampled with noise. We consider hierarchical models like the following:

$$P \sim DP_\alpha \tag{3.38}$$

$$\theta|P \sim P \tag{3.39}$$

$$X|(\theta, P) \sim F(X; \theta), \tag{3.40}$$

where $X \perp\!\!\!\perp P \mid \theta$. An example is when P is a Dirichlet process on the real line, and $X|\theta$ is Normal with mean θ.

Theorem 3.3.1 (Posterior Dirichlet Process under Noisy Sampling)**.** *Let P be a Dirichlet process on $(\Theta, \mathcal{A})$ with base measure α, $\theta|P \sim P$, and $X|(\theta, P) \sim F(X;\theta)$. Define H_x as the posterior distribution of θ given $X = x$ for the hierarchical model $\theta \sim \overline{\alpha}, X|\theta \sim F(X;\theta)$. The posterior distribution of P given $X = x$ is a mixture of Dirichlet processes with mixing distribution H_x on index set $(\Theta, \mathcal{A})$ and $\alpha(u, \cdot) = \alpha + \delta_u$.*

Proof. From Theorem 3.2.6 we know that the marginal distribution of θ is $\overline{\alpha}$. It follows that $X|\theta \sim F(X;\theta)$, from which we see that $\theta|X = x \sim H_x$. Given θ, P is conditionally independent of X. Hence, $P|(\theta, X) \sim DP_{\alpha+\delta_\theta}$. Integrating this with respect to the conditional distribution of $\theta|X = x$ yields

$$P|X = x \sim \int_\Theta DP_{\alpha+\delta_\theta} dH_x(\theta). \qquad \square \quad (3.41)$$

We now consider a noisy sample of size $n = 2$. The notation H_x still represents the posterior distribution of θ when $\theta \sim \overline{\alpha}$ and $X|\theta \sim F(X;\theta)$. Consider the case for which $\theta_1 \neq \theta_2$. Generalizing the above theorem, we can see that $P|(X_1 = x_1, X_2 = x_2, \theta_1 \neq \theta_2)$ is a mixture of Dirichlet processes with mixing distribution $H_{x_1} \times H_{x_2}$ on index set $(\Theta\times\Theta, \mathcal{A}\times\mathcal{A})$ and $\alpha((u_1, u_2), \cdot) = \alpha + \delta_{u_1} + \delta_{u_2}$. On the other hand, if $\theta_1 = \theta_2$, then we essentially have two noisy measurements from a sample of size 1. In this case, let H_{x_1,x_2} be the posterior distribution of θ when (X_1, X_2) are independent with distribution $F(X;\theta)$. Continuing along the same line of reasoning, $P|(X_1, X_2, \theta_1 = \theta_2)$ is a mixture of Dirichlet processes with mixing distribution H_{x_1,x_2} on index set $(\Theta, \mathcal{A})$ and $\alpha(u, \cdot) = \alpha + \delta_u$. Logically, the posterior for P given (X_1, X_2) is a mixture of these two processes weighted by the likelihood that $\theta_1 = \theta_2$. As the sample size increases, the number of mixing components

in the posterior for P increases combinatorially. For this reason, it is common to rely on Markov chain Monte Carlo techniques.

3.3.1 Gibbs Sampling for Mixtures of Dirichlet Processes

In this section, we discuss the general construction of a Gibbs sampler when a Dirichlet process is sampled with noise. Our goal for now is the determine the distribution of $\theta_1, \ldots, \theta_n$ given the noisy observations $(X_1, \ldots, X_n) = \vec{X} \in \mathbb{R}^n$. We will also use the notation $DP(\nu H)$ for $DP_{\nu H}$.

To develop a Gibbs sampling scheme for the Dirichlet mixture model specified by Equations 3.38-3.40, we need to know the conditional distribution of θ_i given the other θ_j's and the data $\vec{X}$. As an illustration, we consider the Chinese restaurant process (CRP) in which each "dish" specifies a distribution $F(X;\theta)$. The CRP makes it clear that the θ_i's are exchangeable so it is enough to specify the conditional distribution of θ_n given $(\theta_1, \ldots, \theta_{n-1}, \vec{X})$. We know that the table at which the n^{th} customer will sit depends only on the number of people at each table. Therefore, θ_n is conditionally independent of $(X_1, \ldots, X_{n-1})$ given $(\theta_1, \ldots, \theta_{n-1})$. Given Equation 3.34 we have

$$\theta_n | (X_1, \ldots, X_{n-1}, \theta_1, \ldots, \theta_{n-1}) \sim \frac{\nu}{\nu + n - 1} H + \frac{1}{\nu + n - 1} \sum_{i=1}^{n} \delta_{\theta_i}. \tag{3.42}$$

(For the remainder, we will not explicitly state the conditioning on $(X_1, \ldots, X_{n-1})$ due to the conditional independence.)

We see that the prior distribution $\pi(\theta_n|\theta_1,\ldots,\theta_{n-1})$ is a mixture of a continuous distribution and $n-1$ degenerate models. We find the posterior distribution for θ_n in light of $X_n = x_n$ by updating the posterior individually for each model and scaling the weights by the likelihood of x_n under those models. For the degenerate models, the posterior distribution of θ_n is still δ_{θ_i} and the likelihood of x_n under this model is simply $dF(x_n;\theta_i)$. On the other hand, if $\theta_n \sim H$, let H_{x_n} be the posterior update for θ_n given x_n and $f_{x_n} = \int f(X_n|\theta_n)dH(\theta_n)$ be the marginal likelihood of x_n under this model. After observing $X_n = x_n$, the posterior update for $\theta_n|(x_n,\theta_1,\ldots,\theta_{n-1})$ is

$$\theta_n|X_n,\theta_1,\ldots,\theta_{n-1} \quad \propto \quad \nu f_{X_n} H_{X_n} + \sum_{i=1}^{n-1} f(X_n;\theta_i)\delta_{\theta_i}. \tag{3.43}$$

Equation 3.43 gives a simple method for updating the n^{th} parameter in the face of data. Since some θ_i's coincide with non-zero probability, we choose to re-write this posterior in terms of the unique values. We will also explicitly express it for an arbitrary $i \leq n$. Denote the unique values of $\theta_{-i} = (\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_n)$ by $\theta_1^*,\ldots,\theta_k^*$, and let n_j be the number of times each unique value occurs. Another way to write the conditional distribution of θ_i emphasizes the clustering properties of the Dirichlet process mixture model.

$$\theta_i|\theta_{-i},X_i \sim w_0 H_{X_i} + \sum_{j=1}^{k} w_j \delta_{\theta_j^*}, \tag{3.44}$$

where the mixing weights are proportional to νf_{X_i} for w_0 and $n_j f(X_i|\theta_j^*)$ for $j > 0$.

In particular, note that the probabilities of existing parameter values are weighted by the likelihood of X_i under those parameters. Therefore, X_i is more likely to be clustered

with other observations that have high density under similar parameters. If $F(\cdot|\theta)$ is unimodal, observations that are near each other are more likely to be clustered together. The Dirichlet process mixture behaves similarly to parametric mixtures in this sense.

Generating θ_i from its full conditional can be done as follows:

1. Calculate $w_0 = \nu f_{X_i}$.
2. For $j = 1, \ldots, k$, calculate $w_j = n_j f(X_i|\theta_j^*)$.
3. Generate $U \sim \text{Unif}\Big(0, \sum_{j=0}^{k} w_j\Big)$.
4. If $U \leq w_0$, then generate $\theta_i \sim G$, otherwise set $\theta_i = \theta_j^*$, where $w_{j-1} < U \leq w_j$.

The only two pieces that depend on the specific model are: calculating the marginal f_{x_i}, and generating $\theta_i \sim H_{x_i}$ if necessary. Convenient choices of F and H enable these to be done simply. In the next subsection, we present an example where H is conjugate to the family $F(X|\theta)$ and has a direct sampling scheme, which simplifies the latter. An important fact is that f_{X_i} does not depend on $\vec{\theta}$, thus these marginals only need to be computed when the sample is initialized. This is not true if we specify additional hyperparameters for $H(\theta)$ that we also wish to update.

In theory, we can construct a Gibbs sampler wherein each iteration consists of updating each θ_i in turn, conditional on X_i and the other parameter values. MacEachern (1994) shows that this Gibbs sampler may be slow to converge and exhibit severe autocorrelation. He proposes a modified version that includes "remixing" the parameters as the last step of each iteration. MacEachern chooses to express the parameter vector as a vector of unique

values ($\vec{\theta^*}$) and a configuration vector, $\vec{t} = (t_1, \ldots, t_n)$, where $\theta_i = \theta^*_{t_i}$. In terms of the Chinese restaurant process, we understand $\vec{\theta^*}$ to be the dishes at each table, and $\vec{t}$ to be the vector of table assignments for the n customers. Essentially, sampling $(\theta_1, \ldots, \theta_n)$ implicitly samples $\vec{t}$. The final step in each iteration is then to sample $\vec{\theta^*}$ conditional on $\vec{t}$ and the data. This is easy because conditional on $\vec{t}$, the θ^*_i are a random sample from H. Thus, each update is a typical Bayesian posterior update given the cluster of observations $\{X_j : t_j = i\}$. MacEachern proves that this revised sampler still converges to the correct posterior distribution, but does so more efficiently in many cases.

3.3.2 A Dirichlet Mixture of Gaussians

To see how the Dirichlet process works, we briefly present an application due to Escobar and West (1995). In this example, $F(\cdot|\theta)$ is the Gaussian family, where $\theta = (\mu, \sigma^2)$ is the mean and the precision. Escobar and West (1995) use a Dirichlet process as a random prior law for the mean and variance of each observation. This leads to the Dirichlet mixture models that we discussed in Section 3.3. Specifically,

$$P \sim DP(\nu H) \tag{3.45}$$

$$(\mu_i, \theta_i)|P \sim P \tag{3.46}$$

$$X_i|(\mu_i, \sigma_i^2) \sim N(\mu_i, \sigma_i^2), \tag{3.47}$$

where for $(\mu, \sigma^2) \sim H$,

$$\sigma^2 \sim IG(s/2, S/2) \tag{3.48}$$

$$\mu|\sigma^2 \sim N(m, \tau\sigma^2). \tag{3.49}$$

We call this measure the Normal×Inverse-Gamma ($N \times IG$) distribution.

Due to the discrete nature of the Dirichlet process, some of the parameters will coincide with positive probability. Thus, the result is a mixture of Gaussians, but the number of components is not known and is allowed to increase as more data are observed. As noted in Section 3.3.1, we can construct a Gibbs sampler to estimate the posterior distribution. We require the posterior and marginal distributions for the simple Bayesian model when $(\mu_i, \sigma_i^2) \sim H$ and $X_i|(\mu_i, \sigma_i^2) \sim N(\mu_i, \sigma_i^2)$. These calculations are simplified because the $N \times IG$ prior specified by Equations 3.48 and 3.49 is conjugate to the Normal distribution.

The posterior distribution of θ_i given X_i is $N \times IG$ with $\sigma_i^2 \sim IG\big((1+s)/2, S_i/2\big)$, where $S_i = (S + (X_i - m)^2)/(1+\tau)$; and $\mu_i|\sigma_i^2 \sim N\big((m + \tau X_i)/(1+\tau), \tau\sigma_i^2/(1+\tau)\big)$. The marginal distribution of X_i is $T(s, m, M)$, the t-distribution with s degrees of freedom, non-centrality parameter m, and scale $M^{1/2}$, where $M = (1+\tau)S/s$. With Equation 3.44, this gives enough information to specify a Gibbs sampler.

In order to increase efficiency, we could incorporate the remixing step of MacEachern (1994). To do so, we also require the posterior distribution of the unique parameters θ_i^* given a sample $\{X_j : \theta_j = \theta_j^*\}$. In this case, we simply need to calculate the posterior for $\theta_i^* = (\mu_i, \sigma_i^2)$. We have that $\sigma_i^2 \sim IG\big((n_i + s)/2, S_i^*/2\big)$ and $\mu_i|\sigma_i^2 \sim N\big((\tau \sum_j X_j + m)/(1+$

$n\tau)\big), \tau\sigma^2/(1+n\tau))$, where $n_i = |\{j : \theta_j = \theta_j^*\}|$, $S_i^* = (S + \sum_j (X_j - m)^2)/(1 + n_i\tau)$ and the summations are over $\{j : \theta_j = \theta_i^*\}$.

3.3.3 Non-Parametric Mixtures and Clustering

The Dirichlet mixture in Section 3.3.2 is a general class of models called *non-parametric mixtures*. The benefit to non-parametric mixtures is that they allow for the gradual admission of new mixture components as more data are observed (McAuliffe et al., 2006). As with parametric mixture models, we can use non-parametric mixtures to cluster observations based on shared parameters. Unfortunately, Dirichlet mixtures tend to find too many clusters with too few members. Thus, a Gaussian component is likely to be better fit by two or more smaller components, according to the model posterior. This has been my experience in my own applications (Chapter 6), which echoes the results of Escobar and West (1995) and others. Nevertheless, the terminology of clusters and mixture components is useful for describing posterior distributions in Dirichlet process mixture models.

3.4 Generalizations of the Dirichlet Process

3.4.1 General Stick-Breaking Processes

In Section 3.2.1 we saw that we can construct a Dirichlet process with precision ν as a stick-breaking measure where $\phi_i \sim \text{Beta}(1, \nu)$, $p_1 = \phi_1$, and $p_i = \phi_i(1 - \sum_{j<i} p_i)$. A generalization of this idea is to allow the Beta parameters to depend on i, analogously to

how we generalize the Dirichlet distribution. This leads to a more general definition of a stick-breaking process.

Definition 3.4.1 (Stick-Breaking Process)**.** *Let $(Z_i)_{i=1}^N$ be a countable sequence of iid random variables with distribution H. Independent of this sequence, let $(\phi_i)_{i=1}^N$ be a countable sequence of independent Beta random variables, for which $\phi_i \sim Beta(a_i, b_i)$. Define $p_1 = \phi_1$. For $1 < i < N$, define $p_i = \phi_i \cdot \prod_{j=1}^{i-1}(1-\phi_j)$. If $N < \infty$, define $\phi_N = \prod_{j=1}^{N-1}(1-\phi_j)$, so that the ϕ-sequence sums to one. If*

$$P = \sum_{i=1}^{N} p_i \delta_{Z_i}, \tag{3.50}$$

then P is a random discrete measure whose law is a stick-breaking process.

The moniker comes from the method for assigning the random weights. We can think of a stick of unit length which represents the mass to be distributed. At each step, we break off a proportion of the remaining stick, ϕ_i, and assign this mass to the point Z_i. Ishwaran and James (2001) provide Gibbs-sampling measures for fitting the types of hierarchical models discussed in Section 3.3, but where the random measure P satisfies this more general stick-breaking process.

3.4.2 Pitman-Yor Process

The Pitman-Yor process (Pitman and Yor, 1997), or two-parameter Poisson-Dirichlet process, is a generalization of the Dirichlet process and an example of a stick-breaking measure. In addition to the base measure, H, it has a discount parameter, $0 \leq d \leq 1$, and a strength

parameter, $\nu > -d$. We discuss the generalization in the setting of the Chinese restaurant process. We consider a hierarchical model with $\theta_1, \theta_2, \ldots$ being a sequence of iid random variables with random distribution P, where P has a Pitman-Yor process prior. Similar to the Chinese restaurant process, we can define a generative scheme for the marginal distribution of the data, with P marginalized out. As before, $\theta_1 \sim H$. For a subsequent draw,

$$\theta_{n+1}|\theta_1, \ldots, \theta_{n+1} \sim \sum_{i=1}^{k} \frac{n_i - d}{n + \nu} \delta_{\theta_i^*} + \frac{\nu + kd}{n\nu} H. \tag{3.51}$$

where $\{\theta_1^*, \ldots, \theta_k^*\}$ are the unique values of the previous draws, with multiplicities $(n_1, \ldots, n_k)$. The discount parameter, d, reduces the clustering effect. For the limiting case with $d = 1$, $\theta_1, \ldots, \theta_n$ is an iid sample from H. On the other extreme, if $d = 0$, then we recover the Dirichlet process. The strength parameter ν is similar to the precision parameter of the Dirichlet process. It is equivalent to the prior sample size, and also helps control the degree of clustering. As Teh (2006) shows, the number of unique values increases stochastically with both d and ν.

The Pitman-Yor process with discount d and strength ν is a stick-breaking process, for which $a_i = 1 - d$ and $b_i = \nu + id$. It produces heavier tails than the Dirichlet process, and is especially useful in natural language processing (Teh, 2006; Goldwater et al., 2006; Wallach et al., 2008). Sudderth and Jordan (2008) used it for image processing.

3.4.3 Neutral to the Right and Tailfree Priors

Neutral to the right priors were first introduced by Doksum (1974) by relaxing some of the independence constraints for a Dirichlet process. Recall that a Dirichlet process is the only non-trivial process which is *neutral* with respect to every sequence of nested partitions. Nonetheless, the priors that Doksum introduces are neutral with respect to a broad set of partitions. These partitions can only be specified for a space with some natural ordering, e.g. $\mathbb{R}$.

Definition 3.4.2. *A random distribution function, F, and its law are said to be* neutral to the right if and only if, for all increasing sequences, $t_1 < t_2 < \ldots < t_k$,

$$\perp\!\!\!\perp \left\{ F(t_1), \frac{F(t_2) - F(t_1)}{1 - F(t_1)}, \ldots, \frac{F(t_k) - F(T_{k-1})}{1 - F(t_{k-1})} \right\}. \tag{3.52}$$

Essentially, these priors are neutral with respect to any nested sequence of partitions $\{\Pi_m\}$ such that for each m, $A_{m,1}, \ldots, A_{m,k_m}$ are sub-intervals dividing the space from left-to-right. Clearly, the Dirichlet process is neutral to the right, but Doksum (1974) shows that there are other interesting processes with this property. These more general priors satisfy some of the same desirable properties; they are closed under sampling and posterior updates are not complicated.

Related to neutral to the right priors are the tailfree priors of Freedman (1962) and Fabius (1964). A prior is tailfree with respect to the tail (s, ∞) if it is neutral to the right for all increasing sequences beginning with $t_1 = s$.

3.4.4 Pólya Trees

Like neutral to the right priors, Pólya trees arise when independence constraints of Dirichlet processes are relaxed. They are important because they are tailfree processes that, unlike a Dirichlet process, may be continuous (Ferguson, 1974). Essentially, a Pólya tree is a process which is F-neutral and neutral with respect to a specific sequence of nested partitions. Let $\Pi = \{\Pi_m\}$ be a nested sequence of partitions where $\Pi_m = \{A_{m1}, \ldots, A_{k_m}\}$. Denote the unique j such that $A_{m,i} \subseteq A_{m-1,j}$ by $\ell(m,i)$. Let $\mathscr{A} = \{a_{ij} : i = 1, 2, \ldots, j = 1, \ldots, k_i\}$ be a set of non-negative numbers associated with each set in the sequence of partitions. The random measure P is a *Pólya tree* if for every m, j, $\{P(A_{m,i}|A_{m-1,j}) : \ell(m,i) = j\}$ is a Dirichlet random variable with parameters $\{a_{m,i} : \ell(m-1,i) = j\}$. Sometimes the definition of a Pólya tree is restricted so that each $A_{m,i} \in \Pi_m$ is divided into exactly two sets in the next partition Π_{m+1}. Lavine (1992) notes that both definitions lead to the same set of available processes.

A Dirichlet process is a special case of a Pólya tree for which $a_{m,i} = \sum_{j:\ell(m+1,j)=i} a_{m+1,j}$. The posterior update for a Pólya tree is easy to calculate. For $\theta \sim P$, we increment by one each $a_{m,i}$ such that $\theta \in A_{m,i}$.

3.4.5 Hierarchical Dirichlet Process

The hierarchical Dirichlet process (Teh et al., 2006) is an extension of the Dirichlet mixture model to account for an additional level of clustering. In some settings it is desirable to have observations in one cluster inform about parameters in other clusters. That is, we

may want the distributions in each cluster to "shrink" toward some global mean. Instead of one Dirichlet process, we have K processes corresponding to K sub-populations. A straightforward application of this idea is $P_i \sim DP(\nu H_0), \theta_i | P_i \sim P_i, X_i | \theta_i \sim F(X; \theta_i)$. The problem, as Teh et al., describe it is that no learning takes place between the sub-populations. This is due to the fact that H_0 is continuous which leads to completely distinct atoms for each P_i. The solution is to find a distribution for H_0 which is both discrete and flexible... like the Dirichlet process. Therefore, Teh et al. put a Dirichlet process prior on H_0. The full model is:

$$\begin{aligned} H &\sim DP(\nu_0, H_0) \\ P_i|H &\sim DP(\nu, H) \\ \theta_i|P_i &\sim P_i \\ X_i|\theta_i &\sim F(X; \theta_i) \end{aligned}$$

This process can be illustrated by an extended urn model. At the beginning, we have a "top urn" containing $H_0(\theta)$ balls of color θ and for each of the K groups we have an urn with ν white balls. Each sample is made by choosing a ball from the urn for the observation's group. If the ball is not-white, then its color identifies θ and we replace the ball and one more of the same color. If the ball *is* white, then θ is the color of a ball drawn from the top urn and we replace the ball and one more of the same color to the "top urn". We also replace the white ball and another ball matching θ to the group urn.

In the Chinese restaurant franchise analogy, we consider a chain of an infinite number of restaurants with a shared menu, wherein each restaurant has an infinite number of

tables. Each time somebody enters a restaurant, he chooses a table as in a simple Chinese restaurant process at that specific restaurant with parameters ν and H. For example, suppose there are m_k people at the restaurant and m_{kj} people at the j^{th} table there. The new customer will sit at table j with probability $m_{kj}/(m_k + \nu)$ and he will sit at a new table with probability $\nu/(m_k + \nu)$. Additionally, the first person at each table decides on a dish for the table which is drawn from a global menu. The choice of a new dish is weighted by the number of tables that already are eating that dish —across the entire franchise. For example, suppose that there are currently n tables being used across the entire franchise and that n_i of those tables are eating dish i. The probability that the new customer chooses dish i is $n_i/(n + \nu_0)$. On the other hand, the customer will choose a new dish drawn from H_0, with probability $\nu_0/(n + \nu_0)$.

Teh et al. (2006) apply their hierarchical Dirichlet process to document modeling. Each document in a corpus represents a sub-population and is represented by a distribution over some unknown number of topics. In this situation, it is sensible to have different topics shared partly between different documents, and the hierarchical Dirichlet process enables this. Furthermore, they extend this idea to another layer of hierarchy representing a situation with multiple collections of documents. Here, the "doubly" hierarchical Dirichlet process enables sharing of topics among documents in a collection, but also between collections. Under this scheme, documents from the same collection are modeled to be contain more similar topics than documents from different collections.

3.4.6 Spatial and Dependent Dirichlet Process

MacEachern (2000) developed a class of Dirichlet processes which depend on some covariate for the use of nonparametric models of error in regression. His criteria of a flexible prior that varied smoothly as the covariate changed led him to propose the use of *dependent Dirichlet processes* (DDPs).

Recall the stick-breaking construction of Sethuraman (1994) with random weights $\{p_i\}$ from the stick-breaking construction and random atoms $\{Z_i\}$ from the underlying base measure, H. Let $\mathcal{X}$ be the covariate space. One method to construct a DDP is to specify a set of atoms $\{Z_{ix}\}$ for each $x \in \mathcal{X}$. MacEachern (2000) calls this process a "single-p DDP". More general forms of the DDPs are constructed by allowing base measure H to depend on x so that the expected error varies in $\mathcal{X}$. For "multiple-p DDP", one can generate a different set of random weights for each $x \in \mathcal{X}$ and in this case one may consider changing the precision to specify differing levels of confidence across $\mathcal{X}$. If the stochastic processes which generates the atoms and the weights are continuous in $\mathcal{X}$, then the resulting DDP will be continuous in $\mathcal{X}$ and satisfy both of MacEachern's desiderata. The posterior process is a collection of Dirichlet processes for $x \in \mathcal{X}$. (He also briefly describes dependent versions of other non-parametric priors.) When the covariate $x \in \mathcal{X}$ represents location, a DDP is also known as a spatial Dirichlet process (Gelfand et al., 2005; Duan et al., 2007).

Griffin and Steel (2006) present an interesting twist on the DDP concept that they call an *order-based dependent Dirichlet process* (πDDP). This process is again defined using a stick-breaking representation. Let $\phi_1, \phi_2, \ldots \sim \text{Beta}(a_i, b_i)$ be the Beta sequence for a stick-

breaking prior and define $V_0 = 0$. In a typical stick-breaking process, $p_i = \phi_i \prod_{j<i}(1-\phi_j)$. In a πDDP, each covariate $x \in \mathcal{X}$ specifies a partial permutation $\pi(x)$ of $\mathbb{N}$, meaning that $\pi(x)$ is a permutation of a possibly proper subset of $\mathbb{N}$. In other words, $\pi(x) = \{\pi_i(x)\}$ where $\pi_i(x) \in \mathbb{N}$ and $\pi_i(x) = \pi_j(x)$ only if $i = j$. The stick-breaking weights are then $p_{ix} = \phi_{\pi_i(x)} \prod_{j<i}(1 - \phi_{\pi_j(x)})$. An important component in a πDDP is to specify the stochastic distribution which generates $\pi(x)$ to be continuous in $\mathcal{X}$. This ensures that when $x, y \in \mathcal{X}$ are close, they have similar weights and therefore similar marginal processes.

3.4.7 The Indian Buffet Process (Beta Process)

Recall that the Chinese restaurant process clusters observations based on the tables at which customers sit. Customers who sit at the same table are considered to be clustered together. In the Dirichlet mixture model, observations in the same cluster share a common parameter (or more generally a common distribution.) The Indian buffet process of Griffiths and Ghahramani (2006) is a culinary analog for soft-clustering. Instead of belonging to a single cluster, observations have some set of features belonging to multiple clusters. This arrangement is typically called *soft clustering*.

Imagine a buffet with an infinite number of dishes. Customers enter one at a time and decide to sample some subset of the available dishes. The first customer will sample a Pois(ν) number of dishes, each one generated according to some base measure H. The following customers, $i = 2, 3, \ldots$ will taste some of the dishes sampled by previous customers as well as some new dishes. The probability that customer i tries a previous dish j is $(\nu/K_{i-1} + m_{i-1,j})/(\nu/K_{i-1} + i - 1)$, where $m_{i-1,j}$ is the number of previous customers who

have tried dish j, and K_i is the total number of dishes sampled by the first i customers. The i^{th} customer will also sample a $\text{Pois}(\nu/i)$ number of new dishes, generated according to H. Each dish in the analogy represents a latent feature and the customers are observations. Let X_i be the i^{th} observation and set $Z_{ij} = 1$ if observation i has feature j and 0 otherwise: $X_i \sim F(X; Z_{i1}, \ldots, Z_{iK_i})$ where K_i is the total number of features active (dishes sampled) so far.

There are several similarities between the Chinese restaurant process and the Indian buffet process. The order in which customers enter does not change the probability so the observations are exchangeable. As in the Chinese restaurant process, we see a strengthening effect such that popular dishes are more likely to be tried. Thibaux and Jordan (2007) find an underlying stochastic process for the distribution of features and also present a hierarchical version. Teh et al. (2007) also provide a stick-breaking construction, though it is not quite a stick-breaking measure.

By DeFinetti's theorem, because the observations are exchangeable, there must be a latent measure P such that $X_1, \ldots, X_n$ are independent. For example, in the Chinese restaurant process, the underlying measure is the Dirichlet process. Thibaux and Jordan (2007) show that the underlying measure is a *Beta process* first defined by Hjort (1990) for modeling hazard rates in survival analysis. Specifically, it is a special case called a Bernoulli process which can be generated as follows for some base measure β. Denote the atoms of β by $c_1, \ldots, c_k$ and let $\beta_0 = \beta - \sum_{i=1}^{k} \beta(c_i)\delta_{c_i}$ be the continuous part of β. Let $N \sim \text{Pois}(\beta_0(\Theta))$ and for each $i = 1, \ldots, N$ let $Z_i \sim \overline{\beta_0}$. For $i = 1, \ldots, k$, let $b_i \sim$

Bernoulli($\beta(c_i)$). The random measure $P = \sum_{i=1}^{N} \delta_{z_i} + \sum_{i=1}^{k} b_i \delta_{c_i}$ is a Bernoulli process with base measure β.

The stick-breaking construction for the Bernoulli process with base measure β proceeds as follows (Teh et al., 2007). Let $\phi_1, \phi_2, \ldots$ be independent Beta($\overline{\beta}, 1$) random variables and set $\phi_0 = 1$. Define the random weights $p_i = \phi_i \prod_{j<i} \phi_j$ and the random atoms $Z_1, Z_2, \ldots \sim \overline{\beta}$, where the atoms are independent of each other and the weights. The measure $P = \sum_i p_i \delta_{Z_i}$ is a Bernoulli process with base measure β. Teh et al. point out that this result could open up the Indian buffet process to various generalizations, as the Sethuraman construction did for the Chinese restaurant process. They also note a remarkable relationship between the Bernoulli process and the Dirichlet process. In both constructions, we begin with a stick of length 1 and proceed at step i by breaking off a proportion ϕ_i of the remaining stick. The difference is that in the Dirichlet process we record the broken off piece as the random weight; in the Bernoulli process we record the remaining length of the stick.

Chapter 4

The Hyper Dirichlet Process

In this chapter we present the theory for the hyper Dirichlet process (Asci et al., 2006; Heinz, 2009). This prior will allow us to specify conditional independence constraints while retaining the flexibility of Dirichlet processes. Dawid and Lauritzen (1993) define hyper Markov priors by specifying the complete version as the clique marginals and forming the hyper Markov combination of those marginals. For example, the hyper inverse Wishart law is the hyper Markov combination of "regular" inverse Wishart laws on each clique. We will define the hyper Dirichlet process in the same way. That is, a hyper Dirichlet process is the hyper Markov combination of hyperconsistent Dirichlet processes. We will also find necessary and sufficient conditions for the hyper Dirichlet process to be a "regular" Dirichlet process. In this case, the construction of the random measure is much simpler. We will then remove one of these conditions, which results in a *graphical Dirichlet process*. This is a Dirichlet process with a Markov base measure that is "not quite hyper Markov"

process. Nevertheless it retains much of the benefit of using a hyper Dirichlet process in the type of mixtures that we discussed in Section 3.3 when the random measure is integrated out. In the terminology of a Chinese restaurant process, the choice of dishes (θ) will be hyper Markov given the table assignments.

4.1 Consistency of Dirichlet Processes

Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a decomposable graph with a perfect ordering of cliques $\mathscr{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_k)$ and separators $\mathscr{S} = (\mathbf{S}_2, \ldots, \mathbf{S}_k)$. For convenience of notation, we may include an empty separators, $\mathbf{S}_1 = \emptyset$; in this case we use the convention that if $X = (X_v : v \in \mathbf{V})$ has distribution F, then the marginal $F_\emptyset(x_\emptyset) = 1$.

In some cases, it is easier to specify a prior for certain marginals, e.g. a prior for each clique. Dawid and Lauritzen (1993) show that, as long as the prior laws are consistent, there is a unique (hyper) Markov law with those marginals. Following this strategy, we would like to specify Dirichlet process priors for each clique. In this section we present the simple method of ensuring that these processes are consistent. Because the parameter of a Dirichlet process is a finite measure, we present a slight generalization of the definition of consistency for probability measures and Markov combinations.

Definition 4.1.1 (Consistency of Finite Measures)**.** *Let μ be a finite measure over $(\mathcal{X}_\mathbf{A}, \mathcal{F}_\mathbf{A})$ and λ be a finite measure over $(\mathcal{X}_\mathbf{B}, \mathcal{F}_\mathbf{A})$. We say that μ and λ are consistent if they induce the same marginal measure over $\mathbf{A} \cap \mathbf{B}$. That is, μ and λ are consistent if*

$$\mu(\mathcal{X}_{\mathbf{A}\setminus\mathbf{B}} \times U) = \lambda(\mathcal{X}_{\mathbf{B}\setminus\mathbf{A}} \times U) \quad \forall\, U \in \mathcal{F}_{\mathbf{A}\cap\mathbf{B}}. \tag{4.1}$$

Recall that $\overline{\mu}$ is the probability measure proportional to μ. Equation 4.1 holds automatically if $\mathbf{A} \cap \mathbf{B} = \emptyset$, otherwise, it holds if and only if the following two conditions are satisfied:

1. $\overline{\mu}$ and $\overline{\lambda}$ are consistent.

2. $\mu(X_{\mathbf{A}}) = \lambda(X_{\mathbf{B}})$.

If μ and λ are probability measures, then the second condition is trivial. Thus, this definition is exactly a generalization of Definition 2.2.3.

Consider these two conditions in the context of base measures for Dirichlet processes. $\overline{\mu}$ is the prior guess about the probability distribution of $X_{\mathbf{A}}$, and $\overline{\lambda}$ is the prior guess for $X_{\mathbf{B}}$. The first condition therefore states that the priors must agree about the distribution of $X_{\mathbf{A}\cap\mathbf{B}}$. It is reasonable to require that our prior be coherent in this way. The second condition states that the prior sample sizes for both sets of variables must be equal. This constraint is perhaps less desirable. It would be perfectly logical to be more certain about certain dimensions than others. Unfortunately, any measure on $\mathcal{X}_{\mathbf{A}\cup\mathbf{B}}$ must satisfy

$$\alpha(\mathcal{X}_{\mathbf{A}\cup\mathbf{B}}) = \int_{\mathcal{X}_{\mathbf{A}\cup\mathbf{B}}} d\alpha = \int_{\mathcal{X}_{\mathbf{A}}} \int_{\mathcal{X}_{\mathbf{B}\setminus\mathbf{A}}} d\alpha = \int_{\mathcal{X}_{\mathbf{A}}} d\alpha_{\mathbf{A}} = \alpha_{\mathbf{A}}(\mathcal{X}_{\mathbf{A}}). \tag{4.2}$$

Similarly, we have $\alpha_{\mathbf{B}}(\mathcal{X}_{\mathbf{B}}) = \alpha(\mathcal{X}_{\mathbf{A}\cup\mathbf{B}}) = \alpha_{\mathbf{A}}(\mathcal{X}_{\mathbf{A}})$. Therefore, if $\mu(\mathcal{X}_{\mathbf{A}}) \neq \lambda(\mathcal{X}_{\mathbf{B}})$ there can be no measure α on $\mathcal{X}_{\mathbf{A}\cup\mathbf{B}}$ satisfying $\alpha_{\mathbf{A}} = \mu$ and $\alpha_{\mathbf{B}} = \lambda$. This is not specific to non-

parametric priors; Dawid and Lauritzen (1993) mention this constraint in their discussion of the hyper Dirichlet distribution. There they mention that if $\mathcal{G}$ is connected, then the prior counts for each clique have the same grand total.

In some situations, this constraint is not too severe. Using stick-breaking notation, we express $\mu = \nu_1 H^{(1)}$ and $\lambda = \nu_2 H^{(2)}$. The consistency conditions translate to $H^{(1)}_{\mathbf{A}\cap\mathbf{B}} = H^{(2)}_{\mathbf{A}\cap\mathbf{B}}$ and $\nu_1 = \nu_2$. If only the second condition fails, then it is still possible to find $H = H^{(1)} \star H^{(2)}$. Employing the stick-breaking condition, we can generate random atoms from H. The problem lies in assigning weights to each atom. Fortunately, in density estimation, the value of the prior precision (ν) is typically small compared to the sample size (n). If the estimate is robust to changes in ν, we may justifiably scale the base measures so that ν_1 and ν_2 are equal. In this case, it is only important that $H^{(1)}$ and $H^{(2)}$ are consistent. In other words, the base measures μ and λ only need to be *proportional* to each other over $\mathcal{X}_{\mathbf{A}\cap\mathbf{B}}$.

There may be other situations in which scale *is* important. Unfortunately, as Equation 4.2 shows, we cannot find a suitable base measure for the prior that satisfies both μ and λ. Without a suitable prior, there can be no suitable posterior. If the goal is to estimate a distribution and there is genuine concern about the precision of the prior estimate, then both conditions must be satisfied.

Subsequently, we assume that both consistency conditions are satisfied. This leads to a natural extension of a Markov combination to finite measures. We have defined consistency of base measures in terms of consistency of probability measures. Thus, we

generalize Markov combinations to include consistent finite measures by scaling them to probability measures, finding the Markov combination, and rescaling the measures.

Definition 4.1.2 (Markov Combination of Finite Measures)**.** *Let μ be a finite measure on $(\mathcal{X}_{\mathbf{A}}, \mathcal{F}_{\mathbf{A}})$. Let λ be a finite measure on $(\mathcal{X}_{\mathbf{B}}, \mathcal{F}_{\mathbf{B}})$ that is consistent with μ. The Markov combination of μ and λ is denoted $\mu \star \lambda$, where*

$$\mu \star \lambda = \mu(\mathcal{X}_{\mathbf{A}}) \cdot [\overline{\mu} \star \overline{\lambda}] = \lambda(\mathcal{X}_{\mathbf{B}}) \cdot [\overline{\mu} \star \overline{\lambda}]. \tag{4.3}$$

This definition is a generalization of Proposition 2.2.4 for probability measures. Note that the Markov combination defined in this way is unique almost everywhere, since $[\overline{\mu} \star \overline{\lambda}]$ is unique almost everywhere.

It is easy to show that the $\bar{\cdot}$ and $\star$ operations commute (with respect to composition).

Theorem 4.1.3. *If μ and λ are consistent measures, then $\overline{\mu \star \lambda} = \overline{\mu} \star \overline{\lambda}$.*

Proof.

$$\begin{aligned} \overline{\mu \star \lambda} &= \frac{[\mu \star \lambda]}{[\mu \star \lambda](\mathcal{X}_{\mathbf{A} \cup \mathbf{B}})} && (4.4) \\ &= \frac{\mu(\mathcal{X}_{\mathbf{A}}) \cdot [\overline{\mu} \star \overline{\lambda}]}{\mu(\mathcal{X}_{\mathbf{A}}) \cdot [\overline{\mu} \star \overline{\lambda}](\mathcal{X}_{\mathbf{A} \cup \mathbf{B}})} && (4.5) \\ &= \overline{\mu} \star \overline{\lambda}. \qquad \square && (4.6) \end{aligned}$$

Writing the base measures in the precision-probability measure notation, set $\mu = \nu H^{(1)}$ and $\lambda = \nu H^{(2)}$. Theorem 4.1.3 states that $\overline{\mu \star \lambda} = H^{(1)} \star H^{(2)}$. Therefore, the Markov combination of $\nu H^{(1)}$ and $\nu H^{(2)}$ can be written $\nu(H^{(1)} \star H^{(2)})$.

To decide if two Dirichlet process laws are hyperconsistent, we need to know their marginal laws.

Theorem 4.1.4 (Marginal Dirichlet Process)**.** *Let $\mathscr{P} = DP_\alpha$ be a Dirichlet process law on some cross-product space $(\Theta_{\mathbf{V}}, \mathcal{A}_{\mathbf{V}})$. For any $\mathbf{A} \subseteq \mathbf{V}$, the marginal law $\mathscr{P}_{\mathbf{A}}$ is a Dirichlet process law on $(\Theta_{\mathbf{A}}, \mathcal{A}_{\mathbf{A}})$ with base measure $\alpha_{\mathbf{A}}$.*

Proof. Let $A_1, \ldots, A_n$ be a measurable partition of $\Theta_{\mathbf{A}}$ and set $B_i = A_i \times \Theta_{\mathbf{V}\setminus\mathbf{A}}$ for each $i \leq n$. For a random measure $P \sim DP_\alpha$, we have $(P(B_1), \ldots, P(B_n)) \sim \mathrm{Dir}(\alpha(B_1), \ldots, \alpha(B_n))$. $P_{\mathbf{A}}(A_i) = P(B_i)$ and $\alpha_{\mathbf{A}}(A_i) = \alpha(B_i)$ for all i, so $(P_{\mathbf{A}}(A_1), \ldots, P_{\mathbf{A}}(A_n)) \sim \mathrm{Dir}(\alpha_{\mathbf{A}}(A_1), \ldots, \alpha_{\mathbf{A}}(A_n))$. Therefore, $P_{\mathbf{A}} \sim DP_{\alpha_{\mathbf{A}}} = \mathscr{P}_{\mathbf{A}}$. □

This leads to the following criterion for determining if two Dirichlet processes are hyperconsistent.

Theorem 4.1.5 (Hyperconsistent Dirichlet Process Laws)**.** *Let $\mathscr{Q} = DP_\mu$ be a Dirichlet process prior on $(\Theta_{\mathbf{A}}, \mathcal{A}_{\mathbf{A}})$ and $\mathscr{R} = DP_\lambda$ be a Dirichlet process on $(\Theta_{\mathbf{B}}, \mathcal{A}_{\mathbf{B}})$. $\mathscr{Q}$ and $\mathscr{R}$ are hyper consistent laws if and only if μ and λ are consistent as per Definition 4.1.1.*

Proof. By Theorem 4.1.4, the marginal law $\mathscr{Q}_{\mathbf{C}} = DP_{\mu_{\mathbf{C}}}$ and by symmetry $\mathscr{R}_{\mathbf{C}} = DP_{\lambda_{\mathbf{C}}}$. These marginals are equal if and only if $\mu_{\mathbf{C}}(U) = \lambda_{\mathbf{C}}(U)$ for every $U \in \mathcal{F}_{\mathbf{C}}$ (i.e. μ and λ are consistent finite measures.) □

Let $\mathcal{G}$ be a decomposable graph with clique set $\mathscr{C}$ and a set marginal laws $\{DP_{\alpha_{\mathbf{C}}} : \mathbf{C} \in \mathscr{C}\}$. If these laws are hyperconsistent, then we will say that their hyper Markov combination

is a *hyper Dirichlet process*. From Theorem 4.1.5, we see that the base measures must be pairwise consistent so we may represent the set $\{\alpha_{\mathbf{C}}\}$ by the unique Markov combination.

Definition 4.1.6. *Let α be a Markov measure on a decomposable graph $\mathcal{G}$ with cliques $\mathbf{C}_1, \ldots, \mathbf{C}_k$. The hyper Dirichlet law with base measure α is denoted HDP_α and defined by*

$$HDP_\alpha = \odot(DP_{\alpha_{\mathbf{C}}} : \mathbf{C} \in \mathscr{C}). \tag{4.7}$$

The hyper Dirichlet process so-defined is a strong hyper Markov measure.

Theorem 4.1.7. *The hyper Dirichlet process law HDP_α is strong hyper Markov.*

Proof. Let $\mathbf{C}$ be any clique and choose any $\mathbf{A} \subset \mathbf{C}$, letting $\mathbf{A}^c = \mathbf{C} \setminus \mathbf{A}$. By Proposition 2.3.6, it is enough to verify that $P_{\mathbf{A}^c|\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{A}}$. Choose any $\theta_{\mathbf{A}} \in \Theta_{\mathbf{A}}$ such that $P_{\mathbf{A}}(\theta_{\mathbf{A}}) > 0$, and $B \in \mathcal{A}_{\mathbf{A}^c}$. Let $B^c = \Theta_{\mathbf{A}^c|\mathbf{A}} \setminus B$. If $H_{\mathbf{A}}(\theta_{\mathbf{A}}) = 0$, then it is almost sure that $Z_{\mathbf{A}i} = \theta_{\mathbf{A}}$ for exactly one value of i. Therefore, $P_{\mathbf{A}^c|\mathbf{A}}(B|\theta_{\mathbf{A}}) = \delta_{Z_{\mathbf{A}^c i}}$, which has distribution $H_{\mathbf{A}^c|\mathbf{A}}(B|\theta_{\mathbf{A}})$ and is independent of $P_{\mathbf{A}}$. Henceforth, assume $H_{\mathbf{A}}(\theta_{\mathbf{A}}) > 0$. In this case, we note that $\{B \times \{\theta_{\mathbf{A}}\}, B^c \times \{\theta_{\mathbf{A}}\}, \Theta_{\mathbf{A}^c} \times \Theta_{\mathbf{A}} \setminus \{\theta_{\mathbf{A}}\}\}$ is a partition of $\Theta_{\mathbf{C}}$ and therefore

$$\begin{aligned} &\big(P_{\mathbf{C}}\left(B \times \{\theta_{\mathbf{A}}\}\right),\ P_{\mathbf{C}}\left(B^c \times \{\theta_{\mathbf{A}}\}\right),\ 1 - P_{\mathbf{A}}\left(\theta_{\mathbf{A}}\right)\big) \\ &\quad\sim\ \mathrm{Dir}\big(H_{\mathbf{C}}(B \times \{\theta_{\mathbf{A}}\}),\ H_{\mathbf{C}}(B^c \times \{\theta_{\mathbf{A}}\}),\ 1 - H_{\mathbf{A}}(\theta_{\mathbf{A}})\big). \end{aligned} \tag{4.8}$$

Using the Gamma representation of the Dirichlet distribution, it follows that

$$\Big(\frac{P_{\mathbf{C}}(B\times\{\theta_{\mathbf{A}}\})}{P_{\mathbf{C}}(B\times\{\theta_{\mathbf{A}}\})+P_{\mathbf{C}}(B^c\times\{\theta_{\mathbf{A}}\})}\ ,\quad \frac{P_{\mathbf{C}}(B\times\{\theta_{\mathbf{A}}\})}{P_{\mathbf{C}}(B\times\{\theta_{\mathbf{A}}\})+P_{\mathbf{C}}(B^c\times\{\theta_{\mathbf{A}}\})}\Big) \sim \mathrm{Dir}\big(H_{\mathbf{C}}(B\times\{\theta_{\mathbf{A}}\}),\ H_{\mathbf{C}}(B^c\times\{\theta_{\mathbf{A}}\})\big). \tag{4.9}$$

Note that the left-hand side is exactly $(P_{\mathbf{A}^c|\mathbf{A}}(B|\theta_{\mathbf{A}}), P_{\mathbf{A}^c|\mathbf{A}}(B^c|\theta_{\mathbf{A}}))$, which we see does not depend on $P_{\mathbf{A}}$. □

As we reviewed in Section 2.3.2, Dawid and Lauritzen (1993) proved that strong hyper Markov laws have many useful properties. The implications are discussed in the Section 4.5.

4.2 The Dirichlet Process as a Hyper Dirichlet Process

By Definition 2.3.4, the hyper Dirichlet process exists. Furthermore, the law DP_α for the entire graph will have the correct marginals by Theorem 4.1.4. Therefore, $DP_\alpha = HDP_\alpha$ if and only if DP_α is hyper Markov, but this is not generally true. We begin by finding the necessary conditions for DP_α to be a hyper Dirichlet process. This will lead us to a constructive definition of the hyper Dirichlet process for more general cases in Section 4.3. In Section 4.4, we will relax the conditions to develop the class of "graphical" Dirichlet process that are not hyper Markov, but do retain useful independence properties for the marginal distribution of the observations.

Some extra notation is useful. If μ is a measure on a space $(\Theta, \mathcal{A})$, we will denote the set of atoms by $\Theta^+ = \{\theta : \mu(\theta) > 0\}$. If Θ is empty, then we say that μ is *continuous* (also *non-atomic*). We call the measure μ *discrete* if $\mu(\Theta^+) = \mu(\Theta)$. If a measure is neither continuous nor discrete, we say it is *mixed*. A distribution which is either mixed or discrete

is *atomic*. For a general measure μ, let $\mu^+ = \sum_{\theta \in \Theta^+} \mu(\theta)\delta_\theta$ and set $\mu^0 = \mu - \mu^+$. Thus, a measure can be decomposed into discrete and continuous measures that are mutually singular by $\mu = \mu^+ + \mu^0$. We allow the possibility that any or all of these measures are identically zero.

Note that a discrete measure, μ can be uniquely expressed by a set of (distinct) atoms and their masses $\widetilde{\mu} = \{(\theta, \mu(\theta)) : \mu(\theta) > 0\}$. We call $\widetilde{\mu}$ the *table representation* of μ as it is basically a pmf table that is taught in introductory statistics classes. Importantly, the table representation is invertible. Given the set of atom-weight pairs, $\widetilde{\mu} = \{(\theta_i, m_i)\}$, we can recover $\mu = \sum_i m_i \delta_{\theta_i}$. Therefore, from an information standpoint we can interchange μ and $\widetilde{\mu}$ in independence relationships as long as μ is discrete. Since Dirichlet processes are almost surely discrete, this will be useful for proving hyper Markovity.

4.2.1 The Dirichlet Process on Two Connected Cliques

Throughout this section, $\mathcal{G}$ will be a decomposable graph that has two cliques $\mathbf{A}$ and $\mathbf{B}$ with non-empty separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. $\mathscr{Q} = DP(\nu Q)$ on $(\Theta_{\mathbf{A}}, \mathcal{A}_{\mathbf{A}})$ and $\mathscr{R} = DP(\nu R)$ on $(\Theta_{\mathbf{B}}, \mathcal{B}_{\mathbf{B}})$ will be hyperconsistent Dirichlet process laws. By Theorem 4.1.5 Q and R are consistent measures, so let $H = Q \star R$ and $\mathscr{L} = DP(\nu H)$. Our goal for this section is to find necessary and sufficient conditions for $\mathscr{L}$ to be the hyper Markov combination $\mathscr{Q} \odot \mathscr{R}$. We note that by Theorem 4.1.4 and the definition of a Markov combination, $H = Q \star R$ implies that $\mathscr{L}_{\mathbf{A}} = \mathscr{Q}$ and $\mathscr{L}_{\mathbf{B}} = \mathscr{R}$. Therefore we need to find conditions such that for $P \sim \mathscr{L}$, $P \in \mathscr{M}(\mathcal{G})$ a.s.$[\mathscr{L}]$ and $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} | P_{\mathbf{S}}$.

Using the stick-breaking construction, $P = \sum_i p_i \delta_{Z_i}$, where $Z_1, Z_2, \ldots$ have distribution H and are independent of each other. Let $\vec{Z}$ denote the infinite-dimensional vector $(Z_1, Z_2, \ldots)$. Denote by $Z_{\mathbf{D}i}$ the marginal value of Z_i for some $\mathbf{D} \subseteq \mathbf{V}$ and let $\vec{Z}_{\mathbf{D}} = (Z_{\mathbf{D}1}, Z_{\mathbf{D}2}, \ldots)$ be the sequence of marginal values. We denote the distinct values of $\{Z_1, Z_2, \ldots\}$ by $\widetilde{\theta} = \{\theta_1, \theta_2, \ldots\}$, where each θ_i is unique. For some $\mathbf{D} \subseteq \mathbf{V}$, $\widetilde{\theta}_{\mathbf{D}} = \{\theta_{\mathbf{D}1}, \theta_{\mathbf{D}2}, \ldots\}$ shall be the distinct values of $\{Z_{\mathbf{D}1}, Z_{\mathbf{D}2}, \ldots\}$.

The reason for the distinction between $\vec{Z}_{\mathbf{D}}$ and $\widetilde{\theta}_{\mathbf{D}}$ is that $P_{\mathbf{D}}$ is equivalent to $\{(\theta_{\mathbf{D}i}, P_{\mathbf{D}}(\theta_{\mathbf{D}i})) : P_{\mathbf{D}}(\theta_{\mathbf{D}}) > 0\}$. Therefore, to show a property such as $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} | P_{\mathbf{S}}$ we will need to work with the unordered, unique values instead of the original ordered vectors. This is not a trivial substitution. As a counterexample, suppose for $i = 1, 2, 3$ that $Z_i \sim$ Bernoulli(.5) and given Z_i, X_i and Y_i are independent uniform random variables on $(Z_i, Z_i + 1)$. Clearly, $(X_1, X_2, X_3) \perp\!\!\!\perp (Y_1, Y_2, Y_3) \mid (Z_1, Z_2, Z_3)$. On the other hand, if $\{Z_1, Z_2, Z_3\} = \{0, 1\}$, then $\{X_1, X_2, X_3\}$ will have two elements greater than 1 if and only if $\{Y_1, Y_2, Y_3\}$ does. Therefore, the conditional independence of the ordered vectors does not transfer to the unordered sets. It is possible that for infinite vectors the independence property would transfer due to an asymptotic property of empirical distributions. Nonetheless, we may not want to rely on this even if it is true. For example, if we wish to generate a random Dirichlet process, we need to approximate it with a finite stick-breaking construction. Fortunately, the next theorem provides a sufficient condition to ensure conditional independence in the sets.

Lemma 4.2.1. *Suppose $\vec{X} = (X_1, \ldots, X_N)$ is a possibly infinite sequence of iid random variables and $\vec{Z} = (Z_1, \ldots, Z_N)$ is another sequence of iid random variables having the*

same length. Let Y be any random variable or sequence. If $\vec{X} \perp\!\!\!\perp Y \mid \vec{Z}$ and $Z_i \neq Z_j$ for all $i \neq j$, then

$$\widetilde{X} \in Y \mid \widetilde{Z}, \tag{4.10}$$

where $\widetilde{X}$ and $\widetilde{Z}$ are the unordered set of unique values in $\vec{X}$ and $\vec{Z}$, respectively.

Proof. Since $\vec{Z}$ is a sequence of unique values, we see that $\widetilde{Z}$ has N elements. Set $\widetilde{Z} = \{Z'_1, \ldots, Z'_N\}$, where the ordering is arbitrary and serves only to distinguish the elements. Let $B = \{\vec{X} : \{X_1, X_2, \ldots\} = \widetilde{X}\}$ be the set of vectors $\vec{X}$ whose unordered distinct values make up the set $\widetilde{X}$.

$$F(\widetilde{X} \mid Y, \vec{Z}, \widetilde{Z}) \quad = \quad \sum_{\vec{X} \in B} \prod_{i=1}^{N} F(X_i \mid Z_i). \tag{4.11}$$

Let σ be a permutation of $\{1, \ldots, N\}$. Certainly, $\vec{X} \in B$ if and only if $(X_{\sigma(1)}, \ldots, X_{\sigma(N)}) \in B$.

$$F(\widetilde{X} \mid Y, \vec{Z}, \widetilde{Z}) = \sum_{\vec{X} \in B} \prod_{i=1}^{N} F(X_{\sigma(i)} \mid Z_i). \tag{4.12}$$

By change of variables $j = \sigma(i)$,

$$F(\widetilde{X} \mid Y, \vec{Z}, \widetilde{Z}) = \sum_{\vec{X} \in B} \prod_{j=1}^{N} F(X_j \mid Z_{\sigma^{-1}(j)}). \tag{4.13}$$

This holds for any permutation, so choose σ such that $Z_{\sigma^{-1}(j)} = Z'_j$. Using this σ in Equation 4.13 yields

$$F(\widetilde{X} \mid Y, \vec{Z}, \widetilde{Z}^*) = \sum_{\vec{X} \in B} \prod_{j=1}^{N} F(X_j \mid Z'_j), \tag{4.14}$$

whence $\widetilde{X} \perp\!\!\!\perp (Y, \vec{Z}) \mid \widetilde{Z})$ and the lemma follows. □

Note that $P_{\mathbf{A}}$ is not just a set of atoms, but a set of atom-weight pairs $\{\theta_{\mathbf{A}}, P_{\mathbf{A}}(\theta_{\mathbf{A}})\}$. Therefore we need to incorporate the random weights into Lemma 4.2.1. Since the weights are not iid, we need a more general theorem. We state this formally as Lemma 4.2.2. Because the sequence of weights are independent of the random atoms, the proof follows very similar logic. We include it here for completeness as we will use the result repeatedly.

Lemma 4.2.2. *Let $\vec{X}, \vec{Z}$, and Y be as in Lemma 4.2.1. Let $\vec{p} = (p_1, \ldots, p_N)$ be a possibly non-iid sequence of the same length as $\vec{X}$ and let W be any random variable, such that $(\vec{p}, W) \perp\!\!\!\perp (\vec{X}, \vec{Z}, Y)$. If $\vec{X} \perp\!\!\!\perp Y \mid \vec{Z}$ and $Z_i \neq Z_j$ for all $i \neq j$, then*

$$\{(X_i, p_i)\} \perp\!\!\!\perp (Y, W) \mid \{(Z_i, p_i)\} \tag{4.15}$$

Proof. The proof follows almost identically to Lemma 4.2.1. Denote $\{(X_i, p_i)\}$ by $\widetilde{X}^*$ and $\{(Z_i, p_i)\}$ by $\widetilde{Z}^*$. Let B be the set of sequences $((X_1, p_1) \ldots (X_N, p_N))$ whose unordered unique values form the set $\widetilde{X}^*$. We note again that the set $\widetilde{Z}^*$ has N elements and we denote them by some arbitrary index, $\widetilde{Z}^* = \{(Z'_1, p'_1), \ldots, (Z'_N, p'_N)\}$.

Given that $(\vec{p}, W) \perp\!\!\!\perp (\vec{X}, Y, \vec{Z})$, we see immediately that

$$F(\vec{X}, \vec{p} | Y, W, \vec{Z}, \vec{p}, \widetilde{Z}^*) = \prod_{i=1}^{N} F(X_i | Z_i) \cdot \mathbb{I}(p_i = p_i), \tag{4.16}$$

and

$$F(\widetilde{X}^* | Y, W, \vec{Z}, \vec{p}, \widetilde{Z}^*) = \sum_{(\vec{X}, \vec{p}) \in B} \prod_{i=1}^{N} F(X_i | Z_i) \cdot \mathbb{I}(p_i = p_i). \tag{4.17}$$

Let σ be a permutation of $1, \ldots, N$ such that $Z_i = Z'_{\sigma(i)}$ (and hence $p_i = p'_{\sigma(i)}$). We have

$$F(\widetilde{X}^*|Y, W, \vec{Z}, \vec{p}, \widetilde{Z}^*) = \sum_{(\vec{X},\vec{p})\in B} \prod_{i=1}^{N} F(X_{\sigma(i)}|Z_i) \cdot \mathbb{I}\left(p_{\sigma(i)} = p_i\right) \tag{4.18}$$

$$= \sum_{(\vec{X},\vec{p})\in B} \prod_{i=1}^{N} F(X_i|Z_{\sigma^{-1}(i)}) \cdot \mathbb{I}\left(p_i = p_{\sigma^{-1}(i)}\right) \tag{4.19}$$

$$= \sum_{(\vec{X},\vec{p})\in B} \prod_{i=1}^{N} F(X_i|Z'_i) \cdot \mathbb{I}\left(p_i = p'_i\right) \tag{4.20}$$

We see that $\widetilde{X}^* \perp\!\!\!\perp (Y, W, \vec{Z}, \vec{p}) \mid \widetilde{Z}^*$ and the lemma follows. □

Immediately from Lemma 4.2.2 we find a sufficient condition for $DP(\nu Q \star R)$ to be hyper Markov. This condition will serve as a starting point for finding a necessary and sufficient condition.

Theorem 4.2.3. *Let $\mathscr{Q} = DP(\nu Q)$ over $(\Theta_{\mathbf{A}}, \mathbf{A}_{\mathbf{A}})$ and $\mathscr{R} = DP(\nu R)$ over $(\Theta_{\mathbf{B}}, \mathcal{B}_{\mathbf{B}})$ be hyperconsistent Dirichlet process laws. If $Q_{\mathbf{A}\cap\mathbf{B}} = R_{\mathbf{A}\cap\mathbf{B}}$ is continuous, then the hyper Markov combination $\mathscr{L} = \mathscr{Q} \odot \mathscr{R} = DP(\nu H)$, where $H = Q \star R$ is the Markov combination of the base measures.*

Proof. $\mathscr{L}$ has the appropriate marginals by Theorem 4.1.4; we need only prove it is hyper Markov. Suppose $P \sim \mathscr{L}$. It is easy to see that $P \in \mathscr{M}(\mathcal{G})$. Since the atoms $\vec{Z}_{\mathbf{S}}$ are almost surely distinct, we have that $P_{\mathbf{A}|\mathbf{S}} = \delta_{Z_i}$, where Z_i is the unique atom such that $Z_{\mathbf{S}i} = \theta_{\mathbf{S}}$. Therefore, $\theta_{\mathbf{A}}$ is completely determined by $\theta_{\mathbf{S}}$. To complete the proof, we must show that $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} \mid P_{\mathbf{S}}$. Consider the stick-breaking construction of a Dirichlet process. Let $P = \sum_{i=1}^{n} p_i \delta_{Z_i}$, where p_i are the stick-breaking weights and $Z_1, Z_2, \ldots$ are independent with distribution H. Let $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. Since H is Markov, $\vec{Z}_{\mathbf{A}} \perp\!\!\!\perp \vec{Z}_{\mathbf{B}} \mid \vec{Z}_{\mathbf{A}\cap\mathbf{B}}$. Furthermore,

$H_{\mathbf{S}}$ is continuous, so the marginal atoms $\vec{Z}_{\mathbf{S}}$ are almost surely distinct and we may apply Lemma 4.2.2 with $\vec{X} = \vec{Z}_{\mathbf{A}}, Y = \vec{Z}_{\mathbf{B}}, \vec{Z} = \vec{Z}_{\mathbf{S}}$ and $\vec{p} = W$.

$$\{(Z_{\mathbf{A}i}, p_i)\} \perp\!\!\!\perp (\vec{Z}_{\mathbf{B}}, \vec{p}) \mid \{(Z_{\mathbf{S}i}, p_i)\}. \tag{4.21}$$

Again noting that the marginal atoms $\vec{Z}_{\mathbf{S}}$ are distinct, we realize that $\{(Z_{\mathbf{S}i}, p_i)\}$ is exactly $\widetilde{P}_{\mathbf{S}}$. Therefore, we may condition on $P_{\mathbf{S}}$ instead. Furthermore, $P_{\mathbf{A}}$ and $P_{\mathbf{B}}$ are functions of $\{(Z_{\mathbf{A}i}, p_i)\}$ and $(\vec{Z}_{\mathbf{B}}, \vec{p})$. Therefore, these are also independent given $P_{\mathbf{S}}$, which is what we wanted to show. □

Asci et al. (2006) provide a different proof of this result that relies on the method of moments. By contrast, our proof relies on Lemma 4.2.2, which is more general in the sense that no distributional assumptions are needed. We use this advantage in Section 4.6 to generalize our work to other stick-breaking processes.

Note that it is vital to the proof of Theorem 4.2.3 that $H_{\mathbf{S}}$ is continuous. This ensures that the random atoms are distinct over $\Theta_{\mathbf{S}}$, which implies that the stick-breaking weights are the probabilities of the distinct atoms in $P_{\mathbf{S}}$. In other words, $P_{\mathbf{S}}$ identifies the random weights up to a permutation. In the general case, some of the $Z_{\mathbf{S}i}$ may coincide. If they do, $P_{\mathbf{S}}$ no longer identifies the stick-breaking weights. For example, if $Z_{\mathbf{S}1} = Z_{\mathbf{S}2}$, then $P_{\mathbf{S}}$ identifies $p_1 + p_2$, but not the individual weights. (To be more precise, even that statement is optimistic; if Z is an atom of $H_{\mathbf{S}}$, we know only that an almost surely countably infinite number of $Z_{\mathbf{S}i} = Z$, and have no method of telling which ones.) Therefore, if $H_{\mathbf{S}}$ is not discrete, the masses may break the conditional probability relationship even though the

atoms are generated by a Markov distribution. To illustrate this, consider an example for which $H_{\mathbf{S}}$ is a point mass. For $H \in \mathscr{M}(\mathcal{G})$, this implies that $H_{\mathbf{A}} \perp\!\!\!\perp H_{\mathbf{B}}$. Further suppose that $H_{\mathbf{A}}$ and $H_{\mathbf{B}}$ are non-atomic. In this case, for $P \sim DP(\nu H)$, $P_{\mathbf{S}}$ is a constant and carries no information. On the other hand $P_{\mathbf{B}}$ identifies the stick-breaking weights modulo permutation. This in turn identifies the masses in $P_{\mathbf{A}}$, though not the atoms. Therefore, $P_{\mathbf{A}}$ and $P_{\mathbf{B}}$ are not independent given $P_{\mathbf{S}}$.

Before we are ready to generalize to other Dirichlet processes, we present a simple result for constructing Dirichlet processes. Simply put, we consider a two-stage process which takes advantage of Property 3.1.5 of the Dirichlet distribution. This theorem will allow us to construct P from two independent Dirichlet processes, one of which satisfies the conditions of Theorem 4.2.3. For notational simplicity, we will use the convention that if P is a Dirichlet process with a base measure equal to zero everywhere, then $P = 0$ everywhere.

Theorem 4.2.4. *Suppose H is a probability measure on a space $(\Theta, \mathcal{A})$ and $\nu > 0$. Let $\Theta^{(1)}, \ldots, \Theta^{(k)}$ be any finite measurable partition of Θ and for each i set $H^{(i)}(\cdot) = H(\cdot \cap \Theta^{(i)})$. If $(h_1, \ldots, h_k) \sim Dir(\nu H(\Theta_1^{(i)}), \ldots, \nu H(\Theta^{(k)}))$; and $P^{(i)} \sim DP(\nu H^{(i)})$ are independent Dirichlet processes that are also independent of $\vec{h}$, then*

$$P = \sum_{i=1}^{k} h_i P^{(i)} \sim DP(\nu H) \tag{4.22}$$

Proof. Let $A_1, \ldots, A_n$ be a measurable partition of Θ and set $A_j^i = A_j \cap \Theta^{(i)}$. $\{B_j^i\}$ is also a measurable partition. By construction, $H^{(i)}$ and $H^{(i')}$ are mutually singular whenever $i \neq i'$. Therefore, $H(A_j^i) = H^{(i)}(A_j^i)$ for all i, j. We note that $(P^{(i)}(A_1^i), \ldots, P^{(i)}(A_n^i)) \sim$

$\text{Dir}(\nu H(A_1^i), \ldots, \nu H(A_n^i))$; $(h_1, \ldots, h_k) \sim \text{Dir}(\nu H(\Theta^{(1)}), \ldots, \nu H(\Theta^{(k)}))$; and $\nu H(A_i) = \sum_{j=1}^n \nu H(A_j^i)$ for all i, j. By Property 3.1.5 of the Dirichlet distribution,

$$(h_i P^{(i)}(A_j^i) : i = 1 \ldots k, j = 1 \ldots n) \sim \text{Dir}(\nu H(A_j^i) : i = 1 \ldots k, j = 1 \ldots n). \tag{4.23}$$

Because $H^{(i)}$ and $H^{(i')}$ are mutually singular for $i \neq i'$, so too are $P^{(i)}$ and $P^{(i')}$. Therefore, for each A_j in the partition we have $P(A_j) = \sum_{i=1}^k h_i P^{(i)}(A_j) = \sum_{i=1}^k h_i P^{(i)}(A_j^i)$. Furthermore, $H(A_j) = \sum_{i=1}^k H(A_j^i)$. Using the additivity property of the Dirichlet distribution (Property 3.1.3), we see

$$(P(A_1), \ldots, P(A_n)) \sim DP(\nu H(A_1), \ldots, \nu H(A_n)). \qquad \square \tag{4.24}$$

By considering the limit as $n \to \infty$, Asci et al. (2006) show that this theorem holds for countable partitions of Θ. They rely on this more general proof to find conditions for a Dirichlet process to be the hyper Dirichlet process. For our purposes, we develop the same conditions relying only on n finite, however their limiting case leads to an illuminating constructive definition of the hyper Dirichlet process (Proposition 4.3.1).

As an example of Theorem 4.2.4, we next construct a Dirichlet process from two independent Dirichlet processes based on one of the marginals.

Example 4.2.1. *Suppose H is a measure on $(\Theta, \mathcal{A})$ and $\nu > 0$. Let $\Theta^{(+)} = \{\theta : H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0\}$ and $\Theta^{(0)} = \Theta \setminus \Theta^{(+)}$. Define the measures, $H^{(+)}(\cdot) = H(\cdot \cap \Theta^{(+)})$ and $H^{(0)}(\cdot) = H(\cdot \cap \Theta^{(0)})$. If $(h_+, h_0) \sim Dir(\nu H(\Theta^{(+)}), \nu H(\Theta^{(0)}))$; $P^{(+)} \sim DP(\nu H^{(+)})$; and $P^{(0)} \sim DP(\nu H^{(0)})$, then $P = h_+ P^{(+)} + h_0 P^{(0)} \sim DP(\nu H)$.*

Before showing the main result of this section, we prove one more lemma.

Lemma 4.2.5. *Let $S_1, \ldots, S_k$ be a finite measurable partition of $\Theta_{\mathbf{S}}$ and set $\Theta^{(i)} = \Theta_{\mathbf{V}\setminus\mathbf{S}} \times S_i$. Then $\Theta^{(1)}, \ldots, \Theta^{(k)}$ is a decomposition of Θ. Decompose $P \sim DP(\nu H)$ by $\sum_{i=1}^{k} h_i P^{(i)}$ as in Theorem 4.2.4. The law $DP(\nu H)$ is hyper Markov if and only if the laws of $P^{(i)}$ are each hyper Markov.*

Proof. We first show that P is almost surely Markov if and only if every $P^{(i)}$ is almost surely Markov. Choose any $\theta_{\mathbf{B}} \in \Theta_{\mathbf{B}}$. Note that $\theta_{\mathbf{B}}$ is in the support of P if and only if it is in the support of $P^{(i)}$ for some i. Furthermore, $P_{\mathbf{A}|\mathbf{B}}(\cdot|\theta_{\mathbf{B}}) = P^{(i)}_{\mathbf{A}|\mathbf{B}}(\cdot|\theta_{\mathbf{B}})$, since the various $P^{(j)}$s are mutually singular. Therefore, the LHS depends on $\theta_{\mathbf{S}}$ alone if and only if the RHS does. That is, P is Markov if and only if each $P^{(i)}$ such that $P^{(i)}(\Theta_{\mathbf{B}}) > 0$ is Markov. If any other $P^{(i)}$ exist such that $P^{(i)}(\Theta_{\mathbf{B}}) = 0$, then they are trivially Markov.

To complete the proof we must show that $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} \mid P_{\mathbf{S}}$ if and only if the same conditional independence holds for each $P^{(i)}$. Since the random $P^{(i)}$s are mutually independent and also independent of $\vec{h}$ we have that $P^{(i)}_{\mathbf{A}} \perp\!\!\!\perp P^{(i)}_{\mathbf{B}} \mid P^{(i)}_{\mathbf{S}}$ for all i, if and only if

$$(P^{(1)}_{\mathbf{A}}, \ldots, P^{(k)}_{\mathbf{A}}, \vec{h}) \perp\!\!\!\perp (P^{(1)}_{\mathbf{B}}, \ldots, P^{(k)}_{\mathbf{B}}, \vec{h}) \mid (P^{(1)}_{\mathbf{S}}, \ldots, P^{(k)}_{\mathbf{S}}, \vec{h}) \tag{4.25}$$

Now take $\mathbf{D}$ to be any of $\mathbf{A}, \mathbf{B}$, or $\mathbf{S}$. We note that $P_{\mathbf{D}}|(P^{(1)}_{\mathbf{D}}, \ldots, P^{(k)}_{\mathbf{D}}, \vec{h})$ is the function $\sum_{i=1}^{k} h_i P^{(i)}_{\mathbf{D}}$ and that this function is invertible since $h_i = P_{\mathbf{D}}(\Theta^{(i)})$ and $P^{(i)}_{\mathbf{D}}(\cdot) = P_{\mathbf{D}}(\cdot)/h_i$. Therefore the relation in Equation 4.25 is equivalent to $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} \mid P_{\mathbf{S}}$. □

We are now ready to find the necessary and sufficient conditions for $\mathscr{L} = DP(\nu H)$ to be a hyper Dirichlet process law. The proof will rely on Theorems 4.2.3 and 4.2.4 as well as Lemma 4.2.5.

Theorem 4.2.6. *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a graph with two cliques $\mathbf{A}$ and $\mathbf{B}$ and non-empty separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. The random measure $P \sim DP(\nu H)$ on $(\Theta, \mathcal{A})$ is a hyper Dirichlet process if and only if*

(i) H is Markov on $\mathcal{G}$ and

(ii) For each θ such that $H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0$, at least one of $H_{\mathbf{A}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ and $H_{\mathbf{B}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ is degenerate.

Proof. (Sufficiency.) We will decompose P as in Lemma 4.2.4. Let $\Theta^{(+)} = \{\theta \in \Theta : H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0\}$, and $\Theta^{(0)} = \Theta \setminus \Theta^{(+)}$. Let $\Theta^{(1)}$ be the set of $\theta \in \Theta^{(+)}$ such that $H_{\mathbf{A}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ is degenerate, and set $\Theta^{(2)} = \Theta^{(+)} \setminus \Theta^{(1)}$. $\Theta^{(0)}, \Theta^{(1)}, \Theta^{(2)}$ is a partition of Θ. Let $(h_0, h_1, h_2) \sim \mathrm{Dir}\big(\nu H(\Theta^{(0)}), \nu H(\Theta^{(1)}), \nu H(\Theta^{(2)})\big)$, and for $i = 0, 1, 2$ set $H^{(i)}(\cdot) = H(\cdot \cap \Theta^{(i)})$. Suppose $P^{(i)} \sim DP(\nu H^{(i)})$ are independent Dirichlet process (also independent of (h_0, h_1, h_2)). By Theorem 4.2.4, $P \stackrel{d}{=} h_0 P^{(0)} + h_1 P^{(1)} + h_2 P^{(2)}$. By Lemma 4.2.5, P is a hyper Dirichlet process if $P^{(0)}$, $P^{(1)}$, and $P^{(2)}$ are.

The law of P^0 is hyper Markov by Theorem 4.2.3. We now show that the law of $P^{(1)}$ is also hyper Markov. Consider the stick-breaking representation of $P^{(1)} = \sum_{i=1}^{\infty} p_i \delta_{Z_i}$, where $Z_1, Z_2, \ldots$ are independent with distribution $H^{(1)}$. For each i, $Z_i \in \Theta^{(1)}$, $H_{\mathbf{A}|\mathbf{B}}(\cdot|Z_{\mathbf{S}i})$ is degenerate by construction. Therefore, given $Z_{\mathbf{S}i}$, the conditional probability of $Z_{\mathbf{A}i}$ is equal to one for some element of $\Theta^{(1)}$. We denote that element by the function $a(Z_{\mathbf{S}i})$.

Certainly, $Z_{\mathbf{A}i} = \theta_{\mathbf{A}} = a(Z_{\mathbf{S}i})$ for each i, so therefore, $Z_{\mathbf{A}i} = \theta_{\mathbf{A}}$ if and only if $a(Z_{\mathbf{S}i}) = \theta_{\mathbf{A}}$. Therefore, for any element $\theta_{\mathbf{A}} \in \Theta_{\mathbf{A}}^{(1)}$,

$$P_{\mathbf{A}}^{(1)}(\theta_{\mathbf{A}}) = \sum_{i=1}^{\infty} p_i \mathbb{I}\left(Z_{\mathbf{A}i} = \theta_{\mathbf{A}}\right) = \sum_{i=1}^{\infty} p_i \mathbb{I}\left(a(Z_{\mathbf{S}i}) = \theta_{\mathbf{A}}\right). \tag{4.26}$$

Let $s(\theta_{\mathbf{A}}) = \{\theta_{\mathbf{S}} : a(\theta_{\mathbf{S}}) = \theta_{\mathbf{A}}\}$ and group the addends in the rightmost summation by shared value of $Z_{\mathbf{S}i}$.

$$\begin{aligned} P_{\mathbf{A}}^{(1)}(\theta_{\mathbf{A}}) &= \sum_{\theta_{\mathbf{S}} \in s(\theta_{\mathbf{A}})} \sum_{i=1}^{\infty} p_i \mathbb{I}\left(Z_{\mathbf{S}i} = \theta_{\mathbf{S}}\right) && (4.27) \\ &= \sum_{\theta_{\mathbf{S}} \in s(\theta_{\mathbf{A}})} P_{\mathbf{S}}^{(1)}(\theta_{\mathbf{S}}). && (4.28) \end{aligned}$$

From Equation 4.28, we see that the set of atoms of $P_{\mathbf{A}}^{(1)}$ is the set of $\theta_{\mathbf{A}}$ such that $s(\theta_{\mathbf{A}})$ has positive probability under $P_{\mathbf{S}}^{(1)}$. Furthermore, the probability of those atoms is almost surely $P_{\mathbf{S}}^{(1)}(s(\theta_{\mathbf{A}}))$. We stress that the functions $s(\cdot)$ is determined by the underlying measure $H^{(1)}$ and it is not random. Therefore, we have shown that $P_{\mathbf{A}}^{(1)}$ is a non-random function of $P_{\mathbf{A}}^{(1)}$. It follows that $P_{\mathbf{A}}^{(1)}$ is conditionally independent of anything when $P_{\mathbf{S}}^{(1)}$ is known. Furthermore, since $Z_{\mathbf{A}i}$ is determined by $\theta_{\mathbf{S}i}$, it must hold that $\theta_{\mathbf{A}}$ is determined by $Z_{\mathbf{S}}$ for $\theta \sim P$. Therefore, the law of $P^{(1)}$ satisfies both conditions of a hyper Markov law. We infer by symmetry that the law of $P^{(2)}$ is also hyper Markov, because $\Theta^{(2)} = \Theta^{(0)} \setminus \Theta^{(1)} \subseteq \{\theta \in \Theta^{(0)} : H_{\mathbf{B}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}}) \text{ is degenerate}\}$.

(Necessity). We first show that (ii) is not by itself sufficient and therefore (i) is required, then complete the proof by showing that (ii) is also necessary. Begin by supposing (ii)

holds, but not (i). Note that if condition (ii) is satisfied and H is discrete, then H must be Markov (i.e.- both conditions are satisfied). To see this, suppose H is discrete and satisfies (ii). Choose any atom $(\theta_{\mathbf{B}}, \theta_{\mathbf{S}})$. If $H_{\mathbf{A}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ is degenerate, then $\theta_{\mathbf{A}}|(\theta_{\mathbf{B}}, \theta_{\mathbf{S}})$ is a non-random function of $\theta_{\mathbf{S}}$. If $H_{\mathbf{B}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ is degenerate, then $\theta_{\mathbf{B}}$ is a non-random function of $\theta_{\mathbf{S}}$ and therefore $(\theta_{\mathbf{B}}, \theta_{\mathbf{S}})$ provides the same information as $\theta_{\mathbf{S}}$. In either case, we see that every conditional given $(\theta_{\mathbf{A}}, \theta_{\mathbf{S}})$ is a function of $\theta_{\mathbf{S}}$ alone, which is the very definition of conditional independence. Note further that if the continuous part of H is Markov and (ii) holds, then a similar argument shows that H itself is Markov. Therefore, we may assume that H is not purely discrete and that the continuous part is not hyper Markov.

Decompose H and P as into $H^{(+)}, H^{(0)}, P^{(+)}$ and $P^{(0)}$ as in Example 4.2.1. In part, this means that $P^{(0)} \sim DP(\nu H^{(0)})$. We have already shown that $H^{(0)}$ is neither zero nor Markov. Therefore, we choose $A \in \mathcal{A}_{\mathbf{A}}; \theta_{\mathbf{B}}, \theta^*_{\mathbf{B}} \in \Theta_{\mathbf{B}}$; and $\theta_{\mathbf{S}} \in \Theta_{\mathbf{S}}$ such that $H(A|\theta_{\mathbf{B}}, \theta_{\mathbf{S}}) \neq H(A|\theta^*_{\mathbf{B}}, \theta_{\mathbf{S}})$. Note that $\mathbf{S} \subset \mathbf{A}$ so that we may assume $A = A' \times \{\theta_{\mathbf{S}}\}$ for some $A' \subset \Theta_{\mathbf{A}\setminus\mathbf{S}}$. Because $H^{(0)}_{\mathbf{S}}(\theta_{\mathbf{S}}) = 0$, if $P^{(0)}_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0$ then it is almost surely the case that exactly one atom is equal to $\theta_{\mathbf{S}}$, say $Z_{\mathbf{S}i}$. Therefore, $P_{\mathbf{A}}(A) = P_{\mathbf{S}}(\theta_{\mathbf{S}})$ if $Z_{\mathbf{A}i} \in A$, and $P_{\mathbf{A}}(A) = 0$ otherwise, but this event depends on whether or not $Z_{\mathbf{B}i} = \theta_{\mathbf{B}}$, which in turn is equivalent to the event $P_{\mathbf{B}}(\theta_{\mathbf{B}}) > 0$. Therefore, $P_{\mathbf{A}}(A)$ and $P_{\mathbf{B}}(\theta_{\mathbf{B}})$ are not conditionally independent given $P_{\mathbf{S}}$. Hence, we have shown that (i) is necessary.

To finish the proof, suppose (ii) fails. Choose s such that $H_{\mathbf{S}}(s) > 0$ but neither $H_{\mathbf{A}|\mathbf{S}}(\cdot|s)$ nor $H_{\mathbf{A}|\mathbf{B}}(\cdot|s)$ is degenerate. Choose a partition (A^*_1, A^*_2) of $\Theta_{\mathbf{A}\setminus\mathbf{S}}$ such that both sets have positive probability under $H_{\mathbf{A}|\mathbf{S}}(\cdot|s)$ and a similar partition (B^*_1, B^*_2) of $\Theta_{\mathbf{B}\setminus\mathbf{S}}$. Define $A_i = A^*_i \times \{s\}$, $B_i = B^*_i \times \{s\}$, $D_{ij} = A^*_i \times \{s\} \times B^*_j$, $h_{ij} = \nu H(D_{ij})$, and

$p_{ij} = P(D_{ij})$. Regarding $\{P(D_{ij})\}$, conditioning on $P_{\mathbf{S}}$ is equivalent to conditioning on $P_{\mathbf{S}}(s)$, since the Dirichlet process is F-neutral. The conditional distribution of $\{P(D_{ij})\}$ given $P_{\mathbf{S}}(s)$ satisfies

$$P_{\mathbf{S}}(s)^{-1} \cdot (p_{11}, p_{12}, p_{21}, p_{22}) \sim \text{Dir}(h_{11}, h_{12}, h_{21}, h_{22}), \tag{4.29}$$

By additivity of Dirichlet vectors, the conditional distribution of $(P_{\mathbf{A}}(A_1), P_{\mathbf{A}}(A_2))$ given $P_{\mathbf{S}}(s)$ is

$$P_{\mathbf{S}}(s)^{-1} \cdot (P_{\mathbf{A}}(A_1), P_{\mathbf{A}}(A_2)) = P_{\mathbf{S}}(s)^{-1} \cdot (p_{11} + p_{12}, p_{21} + p_{22}) \sim \text{Dir}(h_{11} + h_{12}, h_{21} + h_{22}). \tag{4.30}$$

On the other hand, if we condition on both $P_{\mathbf{S}}(s)$ and $P_{\mathbf{B}}(B_1)$ then Equations 4.29 and 4.30 are subject to the constraints $p_{11} + p_{21} = P_{\mathbf{B}}(B_1)$ and $p_{12} + p_{22} = \mathbf{B}(B_2)$. For convenience, we will write $w = p_{11} + p_{12}$ and $v = p_{11} + p_{21}$. The above analysis raises the question of finding the distribution of one sub-total (w) when you know the distribution of an intersecting sub-total (v). We will show that w and v are not independent and infer that $P_{\mathbf{A}}(A_1)$ is not independent of $P_{\mathbf{B}}(B_1)$ given $P_{\mathbf{S}}(s)$.

From the neutrality of Dirichlet vectors, we know that $p_{11}/v \sim$Dir(h_{11}, h_{21}) and $p_{12}/(1-v) \sim$Dir(h_{12}, h_{22}).

$$
\begin{aligned}
f(w|v) &= \int_0^w f(p_{11}|v) f(w - p_{11}|v) dp_{11} && (4.31)\\
&= \int_0^w \mathrm{dBeta}\left(\frac{p_{11}}{v}; h_{11}, h_{21}\right) \cdot \mathrm{dBeta}\left(\frac{w - p_{11}}{1 - v}; h_{12}, h_{22}\right) dp_{11} && (4.32)\\
&\propto \int_0^w \left(\frac{p_{11}}{v}\right)^{h_{11}-1} \left(1 - \frac{p_{11}}{v}\right)^{h_{21}-1} \\
&\quad \times \left(\frac{w - p_{11}}{1 - v}\right)^{h_{12}-1} \left(1 - \frac{w - p_{11}}{1 - v}\right)^{h_{22}-1} dp_{11}. && (4.33)\\
& && (4.34)
\end{aligned}
$$

This certainly does not look constant with respect to v, and we conclude by proving it is not. We adopt the notation $r_n(w)$ to indicate a polynomial of degree n in w. We will also use h_+ as an abbreviation for $h_{11} + h_{12} + h_{21} + h_{22}$.

$$
\begin{aligned}
f(w|v) &\propto \int_0^w r_{h_+-5}(p_{11}) p_{11}^{(h_+-4)} v^{-(h_{11}+h_{21}-2)} (1 - v)^{-(h_{21}+h_{22}-2)} dp_{11} && (4.35)\\
&\propto r_{h_+-4}(w) \frac{1}{h_+ - 3} w^{(h_+-3)} v^{-(h_{11}+h_{21}-2)} (1 - v)^{-(h_{21}+h_{22}-2)}. && (4.36)
\end{aligned}
$$

We conclude that $f(w|v)$ is not constant with respect to v. Thus, $P_{\mathbf{A}}(A_1)$ and $P_{\mathbf{B}}(B_1)$ are not independent given $P_{\mathbf{S}}$ and we see that condition (ii) is necessary for a hyper Dirichlet process, which completes the proof. □

In Section 4.5 we will see that even without condition (ii), a Dirichlet process with a Markov base measure retains many useful properties.

4.2.2 The Dirichlet Process for Connected Decomposable Graphs

In this section we consider a connected decomposable graph, $\mathcal{G}$, with a perfect ordering of cliques given by $\mathscr{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$. Since $\mathcal{G}$ is connected, none of the separators $\mathbf{S}_2, \ldots, \mathbf{S}_k$ are empty. We denote the histories by $\mathbf{H}_i = \cup_{j=1}^{i} \mathbf{C}_i$. We will construct a hyper Dirichlet process for the entire graph.

Theorem 4.2.7. *Let $\mathcal{G}$ be a decomposable graph with a perfect ordering of cliques $\mathbf{C}_1, \ldots, \mathbf{C}_k$. Denote the i^{th} separator and history by $\mathbf{S}_i$ and $\mathbf{H}_i$. $P \sim DP(\nu H)$ is a hyper Dirichlet process on $\mathcal{G}$ if and only if:*

(i) H is Markov with respect to $\mathcal{G}$.

(ii) For each i, s such that $H_{\mathbf{S}_i}(s) > 0$, at least one of $H_{\mathbf{H}_{i-1}|\mathbf{S}_i}(\cdot|s)$ and $H_{\mathbf{C}_i|\mathbf{S}_i}(\cdot|s)$ is degenerate.

Proof. We show this by induction on k. Theorem 4.2.6 shows the case where $k = 2$. Now suppose $k > 2$ and the theorem holds for $k - 1$. We can apply Theorem 4.2.6 again with $\mathbf{A} = \mathbf{H}_{k-1}, \mathbf{B} = \mathbf{C}_k$, and $\mathbf{S} = \mathbf{A} \cap \mathbf{B} = \mathbf{S}_k$. □

We conclude the section by showing that if a Dirichlet process is hyper Markov, then so is the posterior process.

Theorem 4.2.8. *Suppose $P \sim DP(\nu H)$ is a hyper Dirichlet process on a graph $\mathcal{G}$ and that $\theta_1, \ldots, \theta_n$ are an iid sample from P. The posterior law of P, namely $DP(\nu H + \sum_{i=1}^{n} \delta_{\theta_i})$ is also a hyper Dirichlet process.*

Proof. It is well-known that the posterior law is $DP(\nu H + \sum_{i=1}^{n} \delta_{\theta_i})$ and by Corollary 2.3.8, (strong) hyper Markov laws are closed under sampling. □

4.3 General Version of a Hyper Dirichlet Process

Section 4.2 provided the necessary and sufficient conditions for a Dirichlet process to be *hyper* Dirichlet process. Asci et al. (2006) arrive at the same conclusion through very different means. While we concentrated on the stick-breaking definition of a Dirichlet process, Asci et al. (2006) characterized the Dirichlet process as the limit of Dirichlet distributions as the dimension increases to infinity.

We noted in Section 4.2.1, that a Markov base measure is not enough to ensure that a Dirichlet process is hyper Markov. This is due to the fact that the random marginals may share information about the stick-breaking weights. This suggests that we could form a hyper Dirichlet process by generating the random atoms from the Markov base measure and generating the weights in a different way. Using this reasoning, Asci et al. (2006) provide a constructive definition of a hyper Dirichlet process.

For simplicity, we return to the case where $\mathcal{G}$ is connected with cliques $\mathbf{A}$ and $\mathbf{B}$ and non-empty separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. Suppose $P \sim DP(\nu H)$ and $H \in \mathscr{M}(\mathcal{G})$. Let $S^+ = \{\theta_{\mathbf{S}} : H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0\}$ be the atoms of $H_{\mathbf{S}}$; set $S^0 = \Theta_{\mathbf{S}} \setminus S^+$. We can decompose P into mutually singular Dirichlet processes as in Theorem 4.2.4 by $P = h_+ P^{(+)} + h_0 P^{(0)}$. Since $H_{\mathbf{S}}^{(0)}$ is non-atomic by construction, we know that $P^{(0)}$ is hyper Markov. Therefore,

we can construct $P^{(0)}$ as a regular Dirichlet process; we need only alter the stick-breaking weights for $P^{(+)}$. The next proposition provides the details of this construction.

Proposition 4.3.1 (Asci et al. (2006))**.** *Let $\mathcal{G}$ be a graph with two cliques $\mathbf{A}$ and $\mathbf{B}$ having non-empty separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. Suppose $\nu > 0$ and H is a Markov measure on $(\Theta, \mathcal{A})$. Let $S^+ = \{s_i : H_{\mathbf{S}}(s_i) > 0\}$ be the atoms of $H_{\mathbf{S}}$. Define $S^{(0)} = \Theta_{\mathbf{S}} \setminus S^+$ and for $i > 0$ define $S^{(i)} = \{s_i\}$. For $i \geq 0$, also define $\Theta^{(i)} = \Theta_{\mathbf{V}\setminus\mathbf{S}} \times S^{(i)}$, $H^{(i)}(\cdot) = H(\cdot \cap \Theta^{(i)})$, and $\nu_i = H_{\mathbf{S}}(S^{(i)})$. Suppose the following Dirichlet processes are mutually independent:*

$$P^* \sim DP(\nu H_{\mathbf{S}}) \tag{4.37}$$

$$P^{(0)} \sim DP(\nu_0 H^{(0)}) \tag{4.38}$$

$$P^{(i)}_{\mathbf{A}} \sim DP(\nu_i H^{(i)}_{\mathbf{A}}) \tag{4.39}$$

$$P^{(i)}_{\mathbf{B}} \sim DP(\nu_i H^{(i)}_{\mathbf{B}}) \tag{4.40}$$

For $i \geq 0$, define $h_i = P^*(S^{(i)})$. For $i > 0$, define the random measure $P^{(i)} = P^{(i)}_{\mathbf{A}} \star P^{(i)}_{\mathbf{B}}$. (This is possible since both random marginals give probability one to $\{s_i\}$.) The measure $P = \sum_{i=0}^{|S^+|} h_i P^{(i)}$ has law $HDP(\nu H)$.

This construction reveals a serious problem with the general case of hyper Dirichlet processes. For each atom of $\mathbf{S}$, we need the markov combination of two Dirichlet processes. This requires $2+2m_1$ stick-breaking sequences, where m_1 is the number of atoms in $\mathbf{S}$. Now suppose that the graph had three cliques $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ instead of two and let m_i be the number of atoms in the i^{th} separator. In this case, we would form the hyper Markov combination of a hyper Dirichlet process on $\Theta_{\mathbf{A}\cup\mathbf{B}}$ and a Dirichlet process on $\Theta_{\mathbf{C}}$. Therefore, we for

each $i = 1, \ldots, m_2$, we would form a Markov combination of a Dirichlet process and a hyper Dirichlet process, each of which requires $2m_1 + 2$ stick-breaking sequences. For the continuous part of $H_{\mathbf{S}_2}$, we require another $2m_1 + 2$ stick-breaking sequences. Finally, we have the regular Dirichlet process to weight the various components. In total, we see that we would require $2m_2m_1 + 2(m_1 + m_2) + 1$ stick-breaking sequences! Therefore, it is important to discuss the conditions that allow us to construct a hyper Dirichlet process with a relatively few number of stick-breaking sequences.

Note that a hyper Dirichlet process with only one stick-breaking sequence would be a Dirichlet process. Therefore, the theory of Section 4.2 provides a good starting ground. In particular, Theorem 4.2.7 provides conditions for a Dirichlet process to be hyper Markov. It is beneficial to consider how general these conditions on H are. Requiring H to be Markov is entirely reasonable; it is exactly what is required in a parametric setting. Therefore, we focus on condition (ii). If H is continuous and the graph is connected, then this condition is trivially satisfied, but two classes of models are excluded here. First, one may desire an atom in P for a specific variable to force a certain value (e.g. zero) to have positive probability. If this variable is in a separator, condition (ii) states that at least one clique marginal is degenerate when a the atom occurs. In this case, condition (ii) is a handicap and if we require many such atoms, then we will need a large number of stick-breaking sequences. More importantly, condition (ii) of Theorem 4.2.7 is not reasonable for disconnected graphs. If the graph is disconnected, then condition (ii) essentially means that all but one connected components must have a degenerate distribution. We can see this because an empty separator always has the same value (i.e.- $\emptyset$) and in this case we

may interpret the degenerate conditional distribution as implying a Dirac measure for one of the connected components. More concretely, suppose $\mathbf{A}$ and $\mathbf{B}$ are in different connected components. If $P_{\mathbf{A}}$ and $P_{\mathbf{B}}$ are generated by the same stick-breaking procedure $DP(\nu H)$, then the weights in $P_{\mathbf{A}}$ provide information about the distribution $P_{\mathbf{B}}$ unless one of them is degenerate. This is clearly unacceptable, so the Dirichlet process is not useful for a hyper Markov prior on a disconnected graph. Fortunately, if the base measure on each connected component is continuous, then the hyper Dirichlet process for a disconnect graph is just a cross product of mutually independent Dirichlet processes. This only requires one stick-breaking sequence per connected component.

As noted, Theorem 4.2.6 agrees with the result shown by Asci et al. (2006), though our proof is significantly different. Asci et al. also provide a very nice constructive definition. Unfortunately, we have just seen that it has limited practical use. As discussed above, the issue lies not with the construction itself, but with the very nature of the hyper Dirichlet process. Namely, the complexity of generating a random hyper Dirichlet process can increase greatly if it is not also a Dirichlet process. For this reason, our concentration in the present work differs dramatically from theirs. Instead of concentrating on updating existing methods to handle hyper Dirichlet processes, we focus on how to apply a simple Dirichlet process to more situations. In Section 4.4, we discuss a third class of Dirichlet processes, obtained by requiring condition (i) but not condition (ii) in Theorem 4.2.7. These processes, which we call *graphical Dirichlet processes*, lie in some sense between the hyper Dirichlet process and the simple Dirichlet process. A graphical Dirichlet process retains the simplicity of a Dirichlet process, but we see in Section 4.5 that there is much

to be gained in terms of hyper Markov models. Essentially, I prove that if we are going to integrate out the random Dirichlet process, as per usual, then the graphical Dirichlet process leads to the same independence structure as the more complicated hyper Dirichlet process.

4.4 Graphical Dirichlet Process

In Section 4.3, we saw that the construction of a hyper Dirichlet process can be unwieldy when it is not also a Dirichlet process. Unfortunately, Dirichlet processes do not work as hyper Markov laws in some situations, most notably for disconnected graphs. Let us examine the conditions of Theorem 4.2.7. In terms of the stick-breaking process, condition (i) ensures that the random atoms have a Markov distribution. Condition (ii) ensures that information about the stick-breaking weights does break the conditional independence structure. Of course, in many applications of the Dirichlet process, the actual measure P will be integrated out. In this case, we care not about the independence structure of P, but about the induced independence structure on the sample $(\vec{\theta})$ and the noisy observations $(\vec{X})$. Therefore, we will consider the following class of Dirichlet processes that satisfy condition (i) but not necessarily condition (ii):

Definition 4.4.1 (Graphical Dirichlet Process)**.** *If $\nu > 0$ and $H \in \mathscr{M}(\mathcal{G})$, then $P \sim DP(\nu H)$ is a* graphical Dirichlet process *on $\mathcal{G}$.*

From the Chinese restaurant process, we see that the unique values of θ are iid with distribution H. Thus, the unique values satisfy the conditional independence expressed

by $\mathcal{G}$, even though the full $\vec{\theta}$ does not. We will consider this class of Dirichlet processes more thoroughly in the discussion of independence properties of a hyper Dirichlet process mixture in the next section.

4.5 Hyper Dirichlet Process Mixtures

In this section, we discuss Dirichlet process mixture models when a hyper Dirichlet process is used. It is a hierarchical model specified by a base measure H, a real number ν, and a transition measure $F(X;\theta)$.

$$\begin{aligned} P &\sim HDP(\nu H) \\ \theta_i|P &\sim P \\ X_i|P,\theta_i &\sim F(X;\theta_i) \end{aligned}$$

The θ_i's and X_i's are mutually independent in the listed conditional distributions. We use the notation $\vec{\theta}$ and $\vec{X}$ to refer to the full samples. We shall understand the nodes of $\mathcal{G}$ to be either marginal values of θ or X as context dictates. This is a slight abuse of notation, but we shall focus on models for which $F(X;\theta_i) \in \mathscr{M}(\mathcal{G})$, meaning that X and θ have the same conditional independence structure. In this case, we shall refer to the model as a *hyper Dirichlet mixture model*. We also point out that the setting has slightly changed from the previous sections. Particularly in Section 4.2 we considered θ to be an observation from a random distribution P. We therefore referred to the distribution of θ and the base measure H as "Markov". In this section, θ is in turn a random distribution for X. Therefore, we may consider its law (and H) to be hyper Markov when considered in relation to X. Both

terms imply the same conditional independence structure of θ. The only difference is that saying that the law (or H) is hyper Markov implies $\theta \in \mathscr{M}(\mathcal{G})$ almost surely. We first note that since they hyper Dirichlet process law is strong hyper Markov, the posterior law of $P|\vec{\theta}$ is also strong hyper Markov by Corollary 2.3.8.

Consider now the full hierarchical model, in which θ is observed with noise. That is, we do not observe $\vec{\theta}$ directly; we observe $\vec{X}$, where $X_i \sim F(X;\theta_i)$. If $F(X;\theta_i)$ is Markov with respect to the same graph $\mathcal{G}$, then the observations are Markov (given $\vec{\theta}$). Since we have purposely constructed P to be hyper Markov, this is to be expected.

Theorem 4.5.1. *In the hyper Dirichlet mixture model, the law of $\vec{\theta}|P$ is hyper Markov and the distribution of $\vec{X}|(\vec{\theta},P)$ is Markov.*

Proof. Since P is hyper Markov, $\theta_{\mathbf{A}} \perp\!\!\!\perp \theta_{\mathbf{B}}|\theta_{\mathbf{A}\cap\mathbf{B}}$ for any decomposition $(\mathbf{A},\mathbf{B})$ of $\mathcal{G}$. Furthermore, $\vec{X}_{\mathbf{A}}|(\vec{X}_{\mathbf{B}},\vec{\theta},P) \sim \prod_{i=1}^n F(X_{\mathbf{A}i}|X_{\mathbf{B}i},\theta_i) = \prod_{i=1}^n F(X_{\mathbf{A}i}|X_{\mathbf{S}i},\theta_i)$. □

Note that $\vec{\theta}|P$ is typically not strong hyper Markov. Given $X_{\mathbf{B}}$, the conditional probability of $X_{\mathbf{A}} \in A$ is equal to $\sum F_{\mathbf{A}}(A;\theta_{\mathbf{A}})P_{\mathbf{A}|\mathbf{B}}(\theta_{\mathbf{A}}|\theta_{\mathbf{B}})H_{x_{\mathbf{B}}}(\theta_{\mathbf{B}}|X_{\mathbf{B}})$, where $H_{x_{\mathbf{B}}}$ is the posterior distribution of $\theta_{\mathbf{B}}$ given $X_{\mathbf{B}}$ and the sum is over the set where the posterior is positive. Letting $F_{\mathbf{B}}(X_{\mathbf{B}})$ be the marginal distribution of $X_{\mathbf{B}}$,

$$P_{\mathbf{A}|\mathbf{B}}(\theta_{\mathbf{A}}|\theta_{\mathbf{B}})H_{x_{\mathbf{B}}}(\theta_{\mathbf{B}}|X_{\mathbf{B}}) = \frac{P(\theta_{\mathbf{A}},\theta_{\mathbf{B}})}{P_{\mathbf{B}}(\theta_b)}\frac{P_{\mathbf{B}}(\theta_{\mathbf{B}})F_{\mathbf{B}}(X_{\mathbf{B}};\theta_{\mathbf{B}})}{F_{\mathbf{B}}(X_{\mathbf{B}})}. \tag{4.41}$$

$\vec{\theta}|P$ is strong hyper Markov only if this quantity is independent of $\theta_{\mathbf{B}}$. Furthermore, when $\vec{\theta}|P$ is not strong hyper Markov, there is no guarantee that $\vec{X}|P$ is Markov. If instead we integrate over the random measure P, we get the next theorem.

Theorem 4.5.2. *In the hyper Dirichlet mixture model, the law of $\vec{\theta}$ is hyper Markov and the distribution of $\vec{X}|\vec{\theta}$ is Markov.*

Proof. Let $(\mathbf{A}, \mathbf{B})$ be a decomposition of $\mathcal{G}$. $HDP(\nu H)$ is strong hyper Markov. By Corollary 2.3.10, $\vec{\theta}_{\mathbf{A}} \perp\!\!\!\perp \vec{\theta}_{\mathbf{B}} \mid \vec{\theta}_{\mathbf{A}\cap\mathbf{B}}$. Furthermore, the distribution of $\vec{X}$ given $\vec{\theta}$ does not depend on P so it is still Markov when P is integrated out. □

Again, the marginal law for $\vec{\theta}$ is typically not strong hyper Markov. Certainly, for a sample of size $n = 1$, we know that the marginal law for θ_1 is H which we may choose to be strong hyper Markov. This does not apply in general for $n > 1$. That is, even if H is strong hyper Markov, the marginal distribution of $\vec{\theta}$ is not strong hyper Markov. To see this, suppose H is continuous and let $\mathbf{A}, \mathbf{B}$ be an decomposition of $\mathbf{V}$. For $n = 2$, consider the two cases $\theta_{\mathbf{B}1} = \theta_{\mathbf{B}2}$ and $\theta_{\mathbf{B}1} \neq \theta_{\mathbf{B}2}$. In the first case, $X_{\mathbf{A}1}$ and $X_{\mathbf{A}2}$ are two observations from a single $\theta_{\mathbf{A}}$. To be more specific, let $d\mu_i$ be the distribution of $\theta_{\mathbf{A}}$ given $X_{\mathbf{B}} = x_{\mathbf{B}i}$ when $\theta \sim H$ and $X|\theta \sim \theta$; let $d\mu$ be the distribution of $\theta_{\mathbf{A}}$ given $X_{\mathbf{B}1} = x_{\mathbf{B}1}$ and $X_{\mathbf{B}2} = x_{\mathbf{B}2}$ when $\theta \sim H$ and $X_1, X_2|\theta$ are independent with common distribution θ. If $\theta_{\mathbf{B}1} = \theta_{\mathbf{B}2}$, then

$$F(x_{\mathbf{A}1}, x_{\mathbf{A}2}|X_{\mathbf{B}1} = x_{\mathbf{B}1}, X_{\mathbf{B}2} = x_{\mathbf{B}2}) = \int F(x_{\mathbf{A}1}|\theta_{\mathbf{A}})F(x_{\mathbf{A}2}|\theta_{\mathbf{A}})d\mu(\theta_{\mathbf{A}}). \tag{4.42}$$

On the other hand, if $\theta_{\mathbf{B}1} \neq \theta_{\mathbf{B}2}$, then

$$F(x_{\mathbf{A}1}, x_{\mathbf{A}2}|X_{\mathbf{B}1} = x_{\mathbf{B}1}, X_{\mathbf{B}2} = x_{\mathbf{B}2}) = \int F(x_{\mathbf{A}1}|\theta_{\mathbf{A}1})F(x_{\mathbf{A}2}|\theta_{\mathbf{A}2})d\mu_1(\theta_{\mathbf{A}_1})d\mu_2(\theta_{\mathbf{A}_2}). \tag{4.43}$$

Therefore, we see that $F_{\mathbf{A}|\mathbf{B}}$ is not independent of $\vec{\theta}_{\mathbf{B}}$ in general.

4.5.1 Graphical Dirichlet Process Mixtures

Let us examine the distribution of $\theta_2|\theta_1$ more closely when H is strong hyper Markov. As noted, this distribution is a mixture of H and δ_{θ_1}. If we know $\theta_2 \sim H$, then θ_2 has strong hyper Markov law. If we know $\theta_2 \sim \delta_{\theta_1}$, then the law of θ_2 is still strong hyper Markov. Therefore, the reason θ_2 does not have a strong hyper Markov law must be due to the weighting of these two components. We illustrate this using the imagery of the Chinese restaurant process. When the n^{th} customer enters, there are $k \leq n-1$ tables occupied by previous customers. We are curious about the conditional probability of θ_n given $\theta_1, \ldots, \theta_{n-1}$. Therefore, we condition on knowing the tables for the previous $n-1$ customers and the dish at each table, $\{\theta_j^* : j \leq k\}$. If we know which table the n^{th} customer chose, then either we know θ_n (previous table), or we know $\theta_n \sim H$ (new table.) This corresponds to each mixing component being (strong) hyper Markov if H is (strong) hyper Markov.

We now consider the entire mixture distribution of θ combining all tables. Recall that this n^{th} customer will sit at the j^{th} table with probability $n_j/(\nu + n_j)$, where n_j is the number of previous customers at the table. With probability $\nu/(\nu + n - 1)$ he will sit at

a new table. Suppose now that we are curious which table the n^{th} customer chose. If we know $\theta_{\mathbf{A}}$ for some $\mathbf{A} \in \mathbf{V}$, then this gives a lot information about his table. Indeed, we can rule out any table where $\theta^*_{\mathbf{A}j} \neq \theta_{\mathbf{A}n}$. This corresponds to the conditional probability of $\theta|\theta_{\mathbf{A}n}$ and we note that the mixing probabilities have changed. To continue the analogy, suppose we do not know $\theta_{\mathbf{A}n}$, but we have a noisy observation of it ($X_{\mathbf{A}n}$). In this case, the probability that the n^{th} customer chose an occupied table j must be updated - in the sense of a Bayesian posterior - by the probability of $\theta^*_{\mathbf{A}j}$ given $X_{\mathbf{A}n}$ for previous tables. This in turn changes the conditional probability.

In the preceding analysis, the issue at play is that the conditional independence structure may be broken because marginal values change the posterior probability of the tables. Therefore, it is instructive to represent $\vec{\theta}$ by the unique values of θ^*_j and the table assignments $\vec{t} = (t_1, \ldots, t_n)$. We will consider the distribution of $\vec{\theta}$ and $\vec{X}$ conditional on the table assignments $\vec{t}$. Importantly, we will no longer assume that $DP(\nu H)$ is hyper Markov; we require only that H is a hyper Markov law for θ. In other words, these properties do not require condition (ii) of Theorem 4.2.7. Thus, $DP(\nu H)$ is a graphical Dirichlet process law as in Definition 4.4.1.

Theorem 4.5.3. *Suppose $\vec{\theta} = (\theta_1, \ldots, \theta_n)$ is a sample of size n from a Dirichlet process $P \sim DP(\nu H)$, where H is Markov on some graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$. Conditional on θ_i, let $X_i \sim F(X; \theta_i)$ independently of everything else, where $F(X; \theta_i)$ is also Markov on $\mathcal{G}$. Denote the vector of observations by $\vec{X} = (X_1, \ldots, X_n)$ and the unique values of $\vec{\theta}$ by $\vec{\theta}^* = (\theta^*_1, \ldots, \theta^*_k)$. For each i, choose t_i such that $\theta_i = \theta^*_{t_i}$. Under these conditions:*

(i) the conditional law of $\vec{\theta}$ given $\vec{t}$ is hyper Markov; if H is strong hyper Markov, then the conditional law is strong hyper Markov.

(ii) If H is strong hyper Markov, then the conditional distribution of $\vec{X}|\vec{t}$ is Markov.

Proof. We first point out that condition (ii) is implied by condition (i) and Corollary 2.3.10. Therefore, we need only show condition (i), which we do by induction on the sample size, n. Let $\mathscr{L}^n$ denote the distribution of $\vec{\theta}$, and for $n > 1$, let $\mathscr{L}^{(n)}$ denote the conditional distribution of θ_n given $(\theta_1, \ldots, \theta_{n-1}, \vec{t})$. For a sample of size $n = 1$, $\vec{t} = (1)$ is constant and $\mathscr{L}^1 = H$ which is either hyper Markov of strong hyper Markov by supposition. Suppose now that $n > 1$ and that (i) holds for $n - 1$. Let $(\mathbf{A}, \mathbf{B})$ be any decomposition of $\mathcal{G}$. If there exists $i < n$ such that $t_j = t_n$, then $\mathscr{L}^{(n)} = \delta_{\theta_j}$; if no such j exists, then $\mathscr{L}^{(n)} = H$. In either case $\mathscr{L}^{(n)}_{\mathbf{A}|\mathbf{B}} = \mathscr{L}^{(n)}_{\mathbf{A}|\mathbf{A}\cap\mathbf{B}}$ and so $\mathscr{L}^n = \mathscr{L}^{(n)}\mathscr{L}^{n-1}$ is hyper Markov. Furthermore, suppose H is strong hyper Markov. If $\mathscr{L}^{(n)} = H$, then it must be strong hyper Markov. If there exists $j < n$ such that $t_i = t_j$ then $\mathscr{L}^{(n)}$ is degenerate and therefore $\theta_{\mathbf{B}n}$ is constant. In either case, $\mathscr{L}^{(n)}$ is strong hyper Markov and so is $\mathscr{L}^n = \mathscr{L}^{(n)}\mathscr{L}^{n-1}$. □

Theorem 4.5.3 provides insight into the effect of incorporating a graphical model into a Dirichlet mixture. Simply put, the observations are considered to be members of an unknown number of latent classes. The graph describes the conditional independence structure within each class. Importantly, we see that this theorem requires only that H is a hyper Markov law for θ. Therefore, this framework can be used without regard for condition (ii) in Theorem 4.2.7 if we are willing to condition on class membership. In particular, we may incorporate disconnected graphical models to the mixture without

requiring a stick-breaking sequence for each connected component. Finally, it is important to note that conditioning on the latent class membership is common in Dirichlet mixtures. For example, in the Gibbs sampler of Section 3.3.1, the conditional distribution of θ_i given everything else is expressed as a mixture model and θ_i is randomized by first choosing its class membership according to the posterior probabilities.

Theorem 4.5.3 will also be useful in Chapter 7. Graph selection requires integrating out $\vec{\theta}$ to find the marginal distribution of $\vec{X}$ given a graph $\mathcal{G}$. Theorem 4.5.3 will allow us to integrate out everything except the latent class memberships. This will give rise to a simple Monte Carlo estimation of the marginal in which latent classes are sampled from the Chinese restaurant process. This estimation is simple since the class memberships can be chosen based solely on the number of previous observations in the class without regard to any probability calculations or distributional constraints.

4.6 Other Hyper Markov Stick-Breaking Measures

We now consider hyper Markov versions of other stick-breaking measures. As a simple generalization, if the marginal laws for the cliques of a graph are hyperconsistent stick-breaking processes, then we will call their hyper Markov combination a *hyper stick-breaking process*. The first question is of course, "When are the marginal laws hyperconsistent?" It is easily seen that marginal measure

$$P_{\mathbf{A}} = \sum_i p_i \delta_{Z_{\mathbf{A}i}(A)} \tag{4.44}$$

has a stick-breaking law with the same Beta parameters and base measure $H_{\mathbf{A}}$. Therefore, two stick-breaking laws are hyperconsistent when the underlying base measure, H, is Markov.

Next we consider generalizing the construction of the hyper Dirichlet process in Proposition 4.3.1. As before, we see that the number of stick-breaking sequences can be unwieldy. Therefore, it is beneficial to understand the conditions for which a stick-breaking process is a *hyper* stick-breaking process. As a technical note, the construction of Proposition 4.3.1 relies on the fact that the finite dimensional distributions are Dirichlet. Therefore, it may not be applicable to all stick-breaking laws. In this case, it is even more imperative to know when a simple stick-breaking process is hyper Markov as this guarantees a constructive definition.

While our analysis in Theorems 4.2.6 and 4.2.7 relied on the fact that the Dirichlet distribution is neutral, Heinz (2009) provides a proof without regard to the distribution of weights. As it is the stick-breaking weights that distinguish a Dirichlet process from other stick-breaking measures, it shows that the framework of hyper Dirichlet process can extended to other stick-breaking measures. Note that if $H_{\mathbf{S}}$ is non-atomic, then we may apply Theorem 4.2.3 which relied on Lemma 4.2.2 but not any distributional assumptions of the random atoms. Therefore, we jump immediately to the more general case when $H_{\mathbf{S}}$ is mixed. Heinz points out that the following conditions are sufficient, but perhaps not necessary. A lot of the work of this proof is provided by Lemma 4.2.2, which we purposely made general enough to accommodate this theorem.

Theorem 4.6.1 (Hyper Stick-Breaking Process)**.** *Suppose $\mathcal{G}$ is a decomposable graph with two cliques $\mathbf{A}$ and $\mathbf{B}$ with non-empty separator $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. Let H be a probability measure on $\Theta_{\mathbf{A}\cup\mathbf{B}}$ and let $\{a_i, b_i\}$ be a countable sequence of non-negative numbers such that $a_i + b_i > 0$ for all i. Define $\mathscr{L}$ to be the stick-breaking process with parameters $(H, \vec{a}, \vec{b})$. $\mathscr{L}$ is hyper Markov on $\mathcal{G}$ if the following conditions hold:*

(i) H is Markov on $\mathcal{G}$ and

(ii) $H_{\mathbf{D}|\mathbf{S}}(\cdot|\theta_{\mathbf{S}})$ must be degenerate for all $\theta_{\mathbf{S}}$ such that $H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0$, where $\mathbf{D}$ can be either $\mathbf{B}$ or $\mathbf{A}$. (In contrast to Theorem 4.2.6, the same set of conditionals must be used for all $\theta_{\mathbf{S}}$s.)

Proof. Without loss of generality, we shall assume that $\mathbf{D} = \mathbf{A}$ for the second condition. Define $A' = \mathbf{A} \setminus \mathbf{S}$ and $B' = \mathbf{B} \setminus \mathbf{S}$. Note that $\mathbf{A} = \mathbf{S} \cup \mathbf{A}'$, so that $Z_{\mathbf{A}i} = Z_{\mathbf{A}j} \Rightarrow (Z_{\mathbf{A}i}, Z_{i\mathbf{A}'}) = (Z_{\mathbf{S}j}, Z_{i\mathbf{A}'})$. In other words, the second condition can be expressed equivalently as an "if and only if" statement:

$$Z_{\mathbf{S}i} = Z_{\mathbf{S}j} \iff Z_{\mathbf{A}i} = Z_{\mathbf{A}j} \text{ a.s.}[H]. \tag{4.45}$$

Consider $P \sim \mathscr{L}$. The hyper Markov property has two conditions:

1. $\mathbb{P}(P \in \mathscr{M}(\mathcal{G})) = 1$, and
2. $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} | P_{\mathbf{S}}$.

The first condition follows from Equation 4.45. Let $\theta = (\theta_{\mathbf{A}'}, \theta_{\mathbf{S}}, \theta_{\mathbf{B}'})$ be any point in Θ such that $P_{\mathbf{A}}(\theta_{\mathbf{A}}) > 0$. That is, there exists some i such that $Z_{\mathbf{A}i} = \theta_{\mathbf{A}}$. Equation 4.45 states that $Z_{\mathbf{S}j} = Z_{\mathbf{S}i} = \theta_{\mathbf{S}}$ if and only if $Z_{\mathbf{A}j} = Z_{\mathbf{A}i} = \theta_{\mathbf{A}}$ (a.s.$[H]$). Hence, $\{j : Z_{\mathbf{S}j} = \theta_{\mathbf{S}}\} = \{j : Z_{\mathbf{A}j} = \theta_{\mathbf{A}}\}$. Using the stick-breaking representation, we write the distribution of $\theta_{\mathbf{B}}|\theta_{\mathbf{A}}$.

$$P_{\mathbf{B}|\mathbf{A}}(\theta_{\mathbf{B}}|\theta_{\mathbf{A}}) = \frac{\sum_{i:Z_{\mathbf{A}i}=\theta_{\mathbf{A}}} w_i 1_{\{Z_{\mathbf{B}i}=\theta_{\mathbf{B}}\}}}{\sum_{i:Z_{\mathbf{A}i}=\theta_{\mathbf{A}}} w_i} \tag{4.46}$$

$$= \frac{\sum_{i:Z_{\mathbf{S}i}=\theta_{\mathbf{S}}} w_i 1_{\{Z_{\mathbf{B}i}=\theta_{\mathbf{B}}\}}}{\sum_{i:Z_{\mathbf{S}i}=\theta_{\mathbf{S}}} w_i} \tag{4.47}$$

$$= P_{\mathbf{B}|\mathbf{S}}(\theta_{\mathbf{B}}|\theta_{\mathbf{S}}). \tag{4.48}$$

Therefore, $P \in \mathscr{M}(\mathcal{G})$.

It remains to show that $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} \mid P_{\mathbf{S}}$. Since P is almost surely discrete, we will use the table representation, $P_{\mathbf{D}} = \{(\theta_{\mathbf{D}}, P_{\mathbf{D}}(\theta_{\mathbf{D}})) : P_{\mathbf{D}}(\theta_{\mathbf{D}}) > 0\}$, where $\mathbf{D}$ is any of $\mathbf{A}, \mathbf{B}$, or $\mathbf{S}$. Furthermore, we will partition these sets based on whether or not $H_{\mathbf{S}}(\theta_{\mathbf{S}}) > 0$.

$$\widetilde{\theta}^{+}_{\mathbf{D}} = \{(\theta^{+}_{\mathbf{D}}, P_{\mathbf{D}}(\theta^{+}_{\mathbf{D}})) : P_{\mathbf{D}}(\theta^{+}_{\mathbf{D}}) > 0, H_{\mathbf{S}}(\theta^{+}_{\mathbf{S}}) > 0\}; \tag{4.49}$$

$$\widetilde{\theta}^{0}_{\mathbf{D}} = \{(\theta^{0}_{\mathbf{D}}, P_{\mathbf{D}}(\theta^{0}_{\mathbf{D}})) : P_{\mathbf{D}}(\theta^{0}_{\mathbf{D}}) > 0, H_{\mathbf{S}}(\theta^{0}_{\mathbf{S}}) = 0\}, \tag{4.50}$$

where $\mathbf{D}$ is either $\mathbf{A}, \mathbf{B}$ or $\mathbf{S}$. Note that $H_{\mathbf{S}}(\theta_{\mathbf{S}})$ is not random so $(\widetilde{\theta}^{+}_{\mathbf{D}}, \widetilde{\theta}^{0}_{\mathbf{D}})$ is equivalent to the original table representation and hence also to $P_{\mathbf{D}}$.

Because H is Markov, the independence properties we have at hand are in terms of the original *ordered* sequence of atoms, and not in terms of these sets. Therefore, it is useful to define the following, where we once again understand $\mathbf{D}$ to be any of $\mathbf{A}, \mathbf{B}$, and $\mathbf{S}$:

$$\vec{Z}^{+}_{\mathbf{D}} = (Z_{\mathbf{D}i} : H_{\mathbf{S}}(Z_{\mathbf{S}i}) > 0) \tag{4.51}$$

$$\vec{Z}^{0}_{\mathbf{D}} = (Z_{\mathbf{D}i} : H_{\mathbf{S}}(Z_{\mathbf{S}j}) = 0) \tag{4.52}$$

$$\vec{p}^{+} = (p_i : H_{\mathbf{S}}(Z_{\mathbf{S}i}) > 0) \tag{4.53}$$

$$\vec{p}^{0} = (p_i : H_{\mathbf{S}}(Z_{\mathbf{S}i}) = 0) \tag{4.54}$$

By the Markov property, $\vec{Z}^{0}_{\mathbf{A}} \perp\!\!\!\perp \vec{Z}^{0}_{\mathbf{B}} \mid \vec{Z}^{0}_{\mathbf{S}}$. Since the atoms are iid draws from H, we also have $\vec{Z}^{0}_{\mathbf{A}} \perp\!\!\!\perp (\vec{Z}^{+}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{S}})$. Combining these two expressions yields

$$\vec{Z}^{0}_{\mathbf{A}} \perp\!\!\!\perp (\vec{Z}^{0}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{S}}) \mid \vec{Z}^{0}_{\mathbf{S}}. \tag{4.55}$$

By construction, $H_{\mathbf{S}}(z) = 0$ for any $z \in \vec{Z}^{0}_{\mathbf{S}}$. Therefore, the atoms $Z^{0}_{\mathbf{S}i}$ are unique. We recall that the stick-breaking weights are independent of the atoms, and note that $\vec{p}^{0}$ is a sequence of the same length as $\vec{Z}^{0}_{\mathbf{A}}$ and $\vec{Z}^{0}_{\mathbf{S}}$. Thus, we can apply Lemma 4.2.2 where we choose $X = \vec{Z}^{0}_{\mathbf{A}}, Z = \vec{Z}^{0}_{\mathbf{S}}, Y = (\vec{Z}^{0}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{S}}), \vec{p} = \vec{p}^{0}$ and $W = (\vec{p}^{+}, \vec{p}^{0})$.

$$\{(Z^{0}_{\mathbf{A}i}, p^{0}_{i})\} \perp\!\!\!\perp (\vec{Z}^{0}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{B}}, \vec{Z}^{+}_{\mathbf{S}}, \vec{p}^{0}, \vec{p}^{+}) \mid \{(Z^{0}_{\mathbf{S}i}, p^{0}_{i}).\} \tag{4.56}$$

Again we note that the atoms $Z^{0}_{\mathbf{S}i}$ are distinct, so $\{(Z^{0}_{\mathbf{S}i}, p^{0}_{i})\} = \widetilde{\theta}^{0}_{\mathbf{S}}$. We note that $\widetilde{\theta}^{0}_{\mathbf{A}}$ is a function of $\{(Z^{0}_{\mathbf{A}i}, p^{0}_{i})\}$ and that the triplet $(\widetilde{\theta}^{0}_{\mathbf{B}}, \widetilde{\theta}^{+}_{\mathbf{B}}, \widetilde{\theta}^{+}_{\mathbf{S}})$ is a function of the quintet. Therefore, Equation 4.56 implies

$$\widetilde{\theta}^0_{\mathbf{A}} \perp\!\!\!\perp (\widetilde{\theta}^0_{\mathbf{B}}, \widetilde{\theta}^+_{\mathbf{B}}, \widetilde{\theta}^+_{\mathbf{S}}) \mid \widetilde{\theta}^0_{\mathbf{S}}. \tag{4.57}$$

Note that the conditional independence also holds if we condition on $\widetilde{\theta}^+_{\mathbf{S}}$ and remove it from the triplet.

$$\widetilde{\theta}^0_{\mathbf{A}} \perp\!\!\!\perp (\widetilde{\theta}^0_{\mathbf{B}}, \widetilde{\theta}^+_{\mathbf{B}}) \mid (\widetilde{\theta}^0_{\mathbf{S}}, \widetilde{\theta}^+_{\mathbf{S}}). \tag{4.58}$$

This almost proves the lemma. We need only include $\widetilde{\theta}^+_{\mathbf{A}}$ on the left-hand side. By the conditions of the lemma, $H_{\mathbf{A}|\mathbf{S}}(\cdot|s_i)$ is degenerate for each $s_i \in \vec{Z}^+_{\mathbf{S}}$. Thus $Z^+_{\mathbf{A}}$ is a function of $Z^+_{\mathbf{S}}$, which we denote by $Z^+_{\mathbf{A}} = (h(s_1), h(s_2), \ldots)$. Furthermore, it follows from Equation 4.45 that $P_{\mathbf{A}}(h(s_i)) = P_{\mathbf{S}}(s_i)$. Therefore,

$$\widetilde{\theta}^+_{\mathbf{A}} = \{(h(s), P_{\mathbf{S}}(s)) : s \in \vec{Z}^+_{\mathbf{S}}\} = \{(h(s), P_{\mathbf{S}}(s)) : (s, P_{\mathbf{S}}(s)) \in \widetilde{\theta}^+_{\mathbf{S}}\}. \tag{4.59}$$

Since $\widetilde{\theta}^+_{\mathbf{A}}$ is a function of $\widetilde{\theta}^+_{\mathbf{S}}$, we may include it in Equation 4.58 to get

$$(\widetilde{\theta}^0_{\mathbf{A}}, \widetilde{\theta}^+_{\mathbf{A}}) \perp\!\!\!\perp (\widetilde{\theta}^0_{\mathbf{B}}, \widetilde{\theta}^+_{\mathbf{B}}) \mid (\widetilde{\theta}^0_{\mathbf{S}}, \widetilde{\theta}^+_{\mathbf{S}}), \tag{4.60}$$

which is equivalent to $P_{\mathbf{A}} \perp\!\!\!\perp P_{\mathbf{B}} \mid P_{\mathbf{S}}$. □

The extent to which this is useful is undecided. Except for Dirichlet processes, stick-breaking processes are not F-neutral with respect to every sequence of partitions. Therefore, there are probably more conditions necessary for a stick-breaking law to be strong

hyper Markov. If the law is not strong hyper Markov, then the marginal distribution of $\vec{\theta}$ is not guaranteed to be marginally Markov. It is likely that graphical stick-breaking mixtures will still be useful for some stick-breaking priors. Ishwaran and James (2003) discuss a more general Chinese restaurant process for certain stick-breaking measures. For these measures, the unique observations $\vec{\theta}^*$ are still an iid sample from the base measure H and therefore the proof of Theorem 4.5.3 still applies. Therefore, if H is strong hyper Markov, then the observations still have a Markov distribution conditional on the latent class assignments. In analogy to Section 4.4, we could use a *graphical* stick-breaking process to induce conditional independence constraints within each component of a mixture distribution.

Chapter 5

Algorithms for Graphical Model Selection

In Chapter 2, we discussed the necessary theory for a stochastic model search. In this section, we present implementation of those theories. Specifically, we show algorithms to decide if the decomposable graph, $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ will continue to be decomposable when an edge $e = (a, b)$ is toggled. Of course, the fundamental decision is to test whether or not $e \in \mathbf{E}$. There are a variety of ways to do this with various benefits and costs. We present a somewhat novel algorithm in this chapter based on an "oriented" junction tree. If the $e \in \mathbf{E}$ is in a clique $\mathbf{C}$, then Theorem 2.1.7 states that the edge can be removed if and only if e is not contained in a neighbor of $\mathbf{C}$ in the junction tree. Therefore, we will need algorithms to find $\mathbf{C}$ and iterate over its neighbors. On the other hand, if $e \notin \mathbf{E}$, then Theorem 2.1.8 states that the edge can be added if $\mathbf{A} \cap \mathbf{B}$ is a separator on the path

between **A** and **B**, where **A** and **B** are the endpoints of the shortest path from a to b. As such, we require algorithms to find the shortest path and then traverse it. In any case, we will need to update the junction tree representation; for the sake of efficiency, we desire an algorithm that performs the update locally.

All of these algorithms have been tested for every possible complication I could conceive of. I also randomly tested these algorithms by randomly selecting 10000 edges in a graph of size $|\mathbf{V}| = 30$, deciding if those edges could be toggled and updating the graph if needed. The test program finished without any problems, and the transitions that I checked manually were correct.

5.1 Graphical Representations

Computer representations of graphical models must balance the speed of information access with memory and update efficiency. In essence, information that is stored in memory is faster to access because it need not be computed, but this increases the amount of memory needed to represent the graph. A classic example of this interplay is a simple test for adjacency. Suppose $\mathcal{G}$ is a graph with p nodes. One representation is a $p \times p$ binary matrix, with bit $g_{ab} = 1$ if $a \sim b$. This representation allows adjacency tests in constant time, but requires p^2 bits. One way to reduce the memory cost, especially if $\mathcal{G}$ is sparse, is to store only the 1s (i.e.- the extant edges). If the graph is represented by p sets, where the a^{th} set contains a's neighbors, then the complexity of an adjacency test is $\log(n)$, where n

is the average number of neighbors for each node. Thus, memory is saved at the cost of computational speed for adjacency tests.

I used three main factors for deciding on a representation. Namely, I considered which statistics of the graph were needed, how transient those statistics were, and how efficiently those statistics could be updated. Obviously, when choosing a representation, it is important to consider what actions we will need to take. For example, in examining hyper Markov models, we need to know the set of cliques in $\mathcal{G}$. Neither the full matrix nor sets of neighbors mentioned above allow this to be done easily, so they are not very good representations for our purpose. Another important consideration is the permanence of the requisite statistics. There is no purpose to write information that is likely to change before it is needed again. Finally, we aim to explore various graphs by adding and removing edges, so the efficiency of updating the representation is significant. Specifically, we prefer representations that can be maintained via local updates. As an example, I originally considered the feasibility of a graphical representation based on a perfect vertex elimination scheme, however this ordering of nodes can be drastically altered by the addition or removal of a single edge. Therefore, I abandoned this concept in favor of a junction tree representation. As we shall soon see, if we make changes to $\mathbf{A}$ and $\mathbf{B}$ in a junction tree, then we need only update cliques on the path connecting them. If we remove a set $\mathbf{A}$, then we need only update the neighboring cliques.

5.1.1 Oriented Junction Tree Representation

The basic junction tree representation is a collection of cliques that must be linked together in some fashion. The natural choice is to associate with each clique a list of pointers that reference its neighbors. This allows us to identify all the neighbors of a clique with ease. On the other hand, there is no efficient way to find a path between two specific cliques. To see this, imagine trying to find a path from $\mathbf{A}$ to $\mathbf{B}$ in the junction tree. To do so, we need to see if $\mathbf{B}$ is one of $\mathbf{A}$'s neighbors. If not, we need to see if $\mathbf{B}$ is one of $\mathbf{A}$'s neighbor's neighbors. This process must continue, spreading out like a search party, until we find $\mathbf{B}$. In a worst-case scenario, it is possible to search the entire graph before finding the path! The solution we implement is to orient the junction tree by identifying one clique as the "root", which we denote here by $\mathbf{C}_0$. If $\mathbf{A} \neq \mathbf{C}_0$, then exactly one neighbor of $\mathbf{A}$ (possibly $\mathbf{C}_0$ itself) lies on the path between $\mathbf{A}$ and $\mathbf{C}_0$. Borrowing terminology from directed graphs, we call this neighbor the *parent* of $\mathbf{A}$, denoted $\mathbf{A}_{\text{par}}$, and say that $\mathbf{A}$ is a *child* of $\mathbf{A}_{\text{par}}$. Thus, we can easily find paths from $\mathbf{A}$ to $\mathbf{C}_0$ and from $\mathbf{B}$ to $\mathbf{C}_0$ by traveling "upward" from parent to parent. If $\mathbf{B}$ is on the path from $\mathbf{A}$ or vice versa, then we are done. Otherwise, we concatenate the first path with the reverse of the second path to find the complete path from $\mathbf{A}$ to $\mathbf{C}_0$ to $\mathbf{B}$. Note that if $\mathbf{C}_0 = \{a, b\}$, and the edge (a, b) is removed, then we may need to remove $\mathbf{C}_0$ from the graph and choose a new root clique. For simplicity, we add an empty pseudo-clique to the junction tree as the root, $\mathbf{C}_0 = \emptyset$. As a result, the root remains constant and all of the true cliques has a parent.

A second problem with the basic clique representation is finding the subgraph of cliques which contain a given node. To wit, if we wish to test if a and b are neighbors, we do not want to traverse the entire graph trying to find the cliques that contain a and b! Therefore, in addition to the oriented junction tree, our representation will include a vector of pointers where the a^{th} pointer references a clique that contains a. We will denote this vector by $\vec{v} = (v_1, v_2, \ldots, v_p)$. By the junction property, all cliques that contain a form a connected set in the junction tree. Therefore, we can find all such cliques locally once we have a starting place.

5.1.2 Clique Representations

As mentioned in Section 5.1.1, one of the basic component of the junction tree representation is the set of cliques. Obviously, each clique should identify the nodes it contains. Testing for adjacency requires us to check if the two given nodes are members of a clique. Furthermore, we may need to insert or remove elements as edges are added or deleted. We could consider any of the various Standard Template Library (STL) containers. The three most common containers are linked lists, vectors, and sets. Vectors afford the ability to check for inclusion in constant time, but do not allow efficient deletion of elements from the middle of the vector. Conversely, linked lists allow efficient deletion of elements, but the complexity of inclusion tests increases linearly with the size of the clique. Therefore, we choose to represent the cliques by *sets*. Sets are *ordered* containers, which allows efficient deletion and insertion of elements and the complexity of inclusion tests increases only logarithmically with the size of the clique.

For calculating Markov distribution functions, we also need the separators in a perfect clique ordering. Therefore, we associate with each clique $\mathbf{C}_i$, its separator $\mathbf{S}_i$ for some perfect ordering. Note that a perfect ordering can be constructed by beginning with the children of $\mathbf{C}_0$, which are the cliques that are direct neighbors of $\mathbf{C}_0$. We next include all cliques which are two edges away from $\mathbf{C}_0$, then three edges, and so on until all cliques have been enumerated. By this construction, each $\mathbf{C}_i$ is included after its parent, which contains $\mathbf{S}_i$ by the junction property. Hence, the separator associated with a clique is simply the intersection of that clique with its parent. We have a choice between storing these separators or calculating them when needed. By the following reasoning, the separators are fairly persistent when edges are toggled. Recall that if $a \sim b$ and the edge can be removed, then (a, b) is an edge in exactly one clique. Similarly, if $a \not\sim b$, and the edge can be added, then (a, b) is an edge in exactly one clique of the new graph. In either case, only one separator needs to be updated for each move. Furthermore, we shall see that only minor changes need to be made to the separator. Thus, it is beneficial to store the separators to enhance speed. We will denote the separator associated with clique $\mathbf{C}$ by $\mathbf{C}_{\text{sep}}$. As with the cliques, we represent the separators as a set of elements. Looking forward, we note that when edges are added, we may need to update the junction tree to maintain a perfect clique ordering. Thus if $\mathbf{A}$ is the parent of $\mathbf{B}$ in the current graph, this relationship may be reversed in the updated graph. In this case, the separator $\mathbf{A} \cap \mathbf{B}$ remains the same, but it is associated with $\mathbf{A}$ instead of $\mathbf{B}$ in the new graph. Therefore, it is beneficial to store the separators outside of the cliques and simply have each clique contain a pointer to its associated separator. In doing so, we can easily swap separators

by swapping the pointers. This is called a *shallow swap*, and is much more efficient than a *deep swap*, which entails element by element copying and erasing.

We also choose to record the residuals, though the benefits are less clear cut. Our reasoning is that they are required for generating hyper inverse Wishart and hyper Normal random variables. Like the separators, most residuals do not change for each graph update, so we store their value rather than compute them on demand.

5.2 Algorithms and Proofs

In this section, we provide algorithms and proofs for performing the requisite tests as well as for updating the junction tree representation.

5.2.1 Adjacency Test

The first necessary test is to decide if a and b are neighbors. Recall that our extended junction tree representation includes a pointers to cliques $\mathbf{A}$ and $\mathbf{B}$ that contain a and b. We can test for adjacency by exploring the subgraph of cliques that contain a starting at $\mathbf{A}$. In fact, the following algorithm does even better as the entire subgraph need not be explored necessarily.

Algorithm 5.2.1 (Adjacency Test)**.**

1. *Define* $\mathbf{C} \leftarrow \mathbf{A}$.
2. *While* $a \in \mathbf{C}$*:*

(a) *If* $b \in \mathbf{C}$, *return* TRUE.

(b) *Set* $\mathbf{C} \leftarrow \mathbf{C}_{\text{par}}$.

3. *Set* $\mathbf{C} \leftarrow \mathbf{B}$.

4. *While* $b \in \mathbf{C}$*:*

(a) *If* $a \in \mathbf{C}$, *return* TRUE.

(b) *Set* $\mathbf{C} \leftarrow \mathbf{C}_{\text{par}}$.

5. *Return* FALSE

Proof. Note that if any ancestor of $\mathbf{A}$ contains (a, b), say $\mathbf{D}$, then the junction property ensures us that a is a member of every clique on the path from $\mathbf{A}$ to $\mathbf{D}$. Therefore, the first while loop will continue until $\mathbf{C} = \mathbf{D}$ or $\mathbf{C}$ is some other clique containing (a, b). In either case, the algorithm returns TRUE correctly. Similarly, if any ancestor of $\mathbf{B}$ contains (a, b), then the second while loop returns TRUE. Now suppose that no ancestors of $\mathbf{A}$ or $\mathbf{B}$ contain (a, b). In this case, the algorithm returns FALSE. Therefore, we must show that no cliques in $\mathcal{G}$ contains (a, b). For a contradiction, suppose $\mathbf{D} \supseteq \{a, b\}$ exists. Let $p^{(a)} = (p_1^{(a)}, \ldots, p_{n_a}^{(a)})$ be the path from $\mathbf{A}$ to $\mathbf{C}_0$ and let $p^{(b)} = (p_1^{(b)}, \ldots, p_{n_b}^{(b)})$ be the path from $\mathbf{B}$ to $\mathbf{C}_0$. Note that these paths intersect, if only because $p_{n_a}^{(a)} = \mathbf{C}_0 = p_{n_b}^{(b)}$. Therefore, choose $i = \min\{i : p_i^{(a)} \in p^{(b)}\}$ and j such that $p_j^{(b)} = p_i^{(a)}$. We see that $(\mathbf{A} = p_1^{(a)}, \ldots, p_i^{(a)} = p_j^{(b)}, \ldots, p_1^{(b)} = \mathbf{B})$ is the unique path from $\mathbf{A}$ to $\mathbf{B}$. Note that none of the cliques on this path contain both a and b. In particular, $\mathbf{D}$ is not on the path.

Therefore, either $\mathbf{A}$ is on the path from $\mathbf{D}$ to $\mathbf{B}$, or $\mathbf{B}$ is on the path from $\mathbf{D}$ to $\mathbf{A}$. In either case, the junction property is violated, so $\mathbf{D}$ cannot exist. □

Note that we can test $a \in \mathbf{A}_{\text{par}}$ by seeing if a is in the separator of $\mathbf{C}$. This is because we've already established $a \in \mathbf{C}$, so if a is also in the parent, then it must be in the separator of $\mathbf{C}$ by our construction. This increases the speed of the test since a smaller set can be checked.

We make two small changes to the algorithm that do not alter the decision process. First, if the algorithm finds a clique which contains (a, b), then we have it return a pointer to this clique instead of a simple TRUE value. This pointer is useful for testing if an edge can be added, as in Section 5.2.2. Secondly, we update the pointers v_a and v_b in the following way. If the first while loop finds that $a \notin \mathbf{C}_{\text{par}}$, then we update $v_a = \mathbf{C}$; we do the same for v_b in the second loop. The primary benefit to this approach is that if $a \not\sim b$, then v_a and v_b point to the highest cliques in the tree that contain a to b when the algorithm ends. In Section 5.2.3, this will be helpful for finding the shortest path from a to b, which is necessary for deciding if the edge (a, b) may be inserted. A secondary benefit is that this may increase the efficiency of future adjacency checks involving a and b. This is because the algorithm is most efficient when $\mathbf{A}$ and $\mathbf{B}$ are the highest cliques in the junction tree which contain a and b. In this situation, the algorithm only needs to examine one or two cliques.

5.2.2 Dropping an Edge

In this section, we provide the algorithms for dropping edges to a decomposable graph. Of course, before attempting to drop an edge, we should check to see that the edge exists using Algorithm 5.2.1. As discussed above, if $a \sim b$, then the adjacency test gives us (a pointer to) a clique, say $\mathbf{C}$, that contains (a, b). The first algorithm tests if a missing edge can be added without losing decomposability.

Algorithm 5.2.2 (Test for Edge Deletion)**.**

1. *If* $a \in \mathbf{C}_{\text{sep}}$ *and* $b \in \mathbf{C}_{\text{sep}}$ *return* FALSE.
2. *Let* $\mathbf{D}$ *iterate over the children of* $\mathbf{C}$*:*
   (a) *If* $a \in \mathbf{D}_{\text{sep}}$ *and* $b \in \mathbf{D}_{\text{sep}}$ *return* FALSE.
3. *Return* TRUE

Proof. By Lemma 2.1.7, the edge (a, b) can be dropped if and only if no neighbors of $\mathbf{C}$ contain (a, b). The first step checks $\mathbf{C}$'s parent, and the loop checks all of $\mathbf{C}$'s children. The decision is made when a neighbor containing (a, b) is found, or when all neighbors have been examined. □

If the test evaluates TRUE, then we can use the next algorithm to update the extended junction tree representation. Once again, we take advantage of our pointer to $\mathbf{C}$, which we now know is the only clique containing (a, b). Let $\mathbf{U} = \mathbf{C} \setminus \{a, b\}$. After the edge (a, b) is removed, we know that $\mathbf{C}$ is no longer a clique and that $\mathbf{U}^a = \mathbf{U} \cup \{a\}$ and $\mathbf{U}^b = \mathbf{U} \cup \{b\}$

are complete sets. Is $\mathbf{U}^a$ a clique? This depends on whether or not there already exists another clique, say $\mathbf{D} \supseteq \mathbf{U}^a$, that contains $\mathbf{U}^a$. Therefore, in order to update the graph, we need to see if such a clique exists. We note that $\mathbf{D}$ may not be unique, however, if $\mathbf{D}$ does exist, then $\mathbf{D} \cap \mathbf{C} = \mathbf{U}^a$. Thus by the junction property, every clique on the path between $\mathbf{D}$ and $\mathbf{C}$ must also contain $\mathbf{U}^a$. Hence, it is enough to check if a neighbor of $\mathbf{C}$ contains $\mathbf{U}^a$. Of course, a similar test is performed for a clique containing $\mathbf{U}^b$. The details are presented in the next algorithm.

Denote the current graph by $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and the new graph by $\mathcal{G}^* = (\mathbf{V}, \mathbf{E} \backslash \{(a, b)\})$. If neither $\mathbf{U}^a$ nor $\mathbf{U}^b$ is contained by another clique of $\mathcal{G}$, then both are cliques in $\mathcal{G}^*$. In this case, $\mathbf{U}^a$ can be formed by removing b from $\mathbf{C}$ and $\mathbf{U}^b$ is created by making a new clique. On the other hand, suppose $\mathbf{U}^a$ is contained in another clique of $\mathcal{G}$, but $\mathbf{U}^b$ is not. In this case, $\mathbf{U}^a$ is not a clique of $\mathcal{G}^*$, and we need only remove a from $\mathbf{C}$. Finally, if both $\mathbf{U}^a$ and $\mathbf{U}^b$ are contained by other cliques of $\mathcal{G}$. In this case, we need to remove $\mathbf{C}$ completely to form $\mathcal{G}^*$. Therefore, we see that there are three general cases, which we will need to handle separately. Furthermore, we treat the case where $\mathbf{U}^a$ or $\mathbf{U}^b$ is contained in the parent specially. This reduces the number of necessary operations to update the separators and residuals. In general, we will come across several examples in which we know one set is a subset of another, so we can check equality by comparing cardinality. In any case, we need to also update separators, residuals, and possibly the structure of the tree.

The algorithm is somewhat simpler if $\mathbf{U}^a$ or $\mathbf{U}^b$ is contained in the $\mathbf{C}_{\text{par}}$. Note that $\mathbf{C}_{\text{par}}$ cannot contain both a and b by Lemma 2.1.7. Thus at least one of a and b is in $\mathbf{C}_{\text{res}}$. Furthermore, if $|\mathbf{C}_{\text{res}}| = 1$, then only a or b is in the residual. This means that we can

test ($\mathbf{U}^a \subset \mathbf{C}_{\text{par}}$ or $\mathbf{U}^b \subset \mathbf{C}_{\text{par}}$), by checking if $|\mathbf{C}_{\text{res}}| = 1$, which is a single comparison. For clarity, we focus on the oriented junction tree and discuss updating the accompanying vector of pointers afterward.

Algorithm 5.2.3 (Drop edge: $|\mathbf{C}_{\text{res}}| == 1$)**.**

1. *If* $\mathbf{C}_{\text{res}}$ *contains* a*, then swap the values of* a *and* b*.*
2. *Set* $\mathbf{A} \leftarrow \mathbf{C}_{\text{par}}$*.*
3. *Initialize* $\mathbf{B} \leftarrow$ NULL*.*
4. *Let* $\mathbf{D}$ *iterate over the children of* $\mathbf{C}$*:*
   (a) *If* $|\mathbf{D}_{\text{sep}}| == |\mathbf{C}| - 1$ *and* $b \in \mathbf{D}_{\text{sep}}$*, then:*
      i. *Set* $\mathbf{B} \leftarrow \mathbf{D}$
      ii. *Remove* $\mathbf{B}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$
      iii. *stop iterating*
5. *If* $\mathbf{B} ==$ NULL *then:*
   (a) *Remove* a *from* $\mathbf{C}$ *and* $\mathbf{C}_{\text{sep}}$*.*
   (b) *Let* $\mathbf{D}$ *iterate over the children of* $\mathbf{C}$*:*
      i. *If* $|\mathbf{D}_{\text{sep}}|$ *contains* a*, then:*
         A. *Set* $\mathbf{D}_{\text{par}} \leftarrow \mathbf{A}$
         B. *Remove* $\mathbf{D}_{\text{par}}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$

C. Insert $\mathbf{D}_{\text{par}}$ *into* $\vec{\mathbf{A}}_{\text{ch}}$.

5′. else:

(a) Set $\mathbf{B}_{\text{par}} \leftarrow \mathbf{A}$.

(b) Insert $\mathbf{B}$ *into* $\vec{\mathbf{A}}_{\text{ch}}$.

(c) Insert b *into* $\mathbf{B}_{\text{res}}$.

(d) Remove b *from* $\mathbf{B}_{\text{sep}}$.

(e) Let $\mathbf{D}$ *iterate over the children of* $\mathbf{C}$*:*

i. If $|\mathbf{D}_{\text{sep}}|$ *contains* b*, then:*

A. Set $\mathbf{D}_{\text{par}} \leftarrow \mathbf{B}$

B. Remove $\mathbf{D}_{\text{par}}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$

C. Insert $\mathbf{D}_{\text{par}}$ *into* $\vec{\mathbf{B}}_{\text{ch}}$.

i′. else:

A. Set $\mathbf{D}_{\text{par}} \leftarrow \mathbf{B}$

B. Remove $\mathbf{D}_{\text{par}}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$

C. Insert $\mathbf{D}_{\text{par}}$ *into* $\vec{\mathbf{B}}_{\text{ch}}$.

(f) Delete $\mathbf{C}$

Proof. Line 1 ensures that $\mathbf{C}_{\text{par}}$ contains $\mathbf{U}^a$, by swapping the values of a and b. This is done to simplify the remaining algorithm. Line 3 and the loop at Line 4 decides if $\mathbf{U}^b$ is a clique in $\mathcal{G}^*$. As noted, we do this by seeing if a neighboring clique contains $\mathbf{U}^b$. Since $\mathbf{D}_{\text{sep}}$

cannot contain both a and b, we have that $|\mathbf{D}_{\text{sep}}| = |\mathbf{C}| - 1$ must imply either $\mathbf{D}_{\text{sep}} = \mathbf{U}^a$ or $\mathbf{D}_{\text{sep}} = \mathbf{U}^b$. We decide between these two by checking if $b \in \mathbf{D}_{\text{sep}}$.

In the event that no $\mathbf{B} \supset \mathbf{U}^b$ is found, we know that $\mathbf{U}^b$ is a clique in $\mathcal{G}*$. Rather than create $\mathbf{U}^b$ from scratch, we simply update $\mathbf{C}$. We note that $\mathbf{U}^b = \mathbf{C} \backslash \{a\}$, and $\mathbf{U}^b_{\text{sep}} \cap \mathbf{C}_{\text{par}} = \mathbf{U} = \mathbf{C}_{\text{sep}} \setminus \{a\}$. These values are set by 5a. We also note that $\mathbf{U}^b_{\text{res}} = \{b\} = \mathbf{C}_{\text{res}}$, so no change is needed there. Finally, we note that if $\mathbf{D} \in \vec{\mathbf{C}}_{\text{ch}}$, then $\mathbf{C}$ is in between $\mathbf{D}$ and $\mathbf{C}_{\text{par}}$ in the junction tree. Therefore, if $a \in \mathbf{D}$, we must make $\mathbf{D}$ a child of $\mathbf{A}$ rather than $\mathbf{C}$. It is easy to verify that the separator and residual for each such $\mathbf{D}$ remains unchanged and hence that the junction property is maintained.

In the event that we find a $\mathbf{B} \supset \mathbf{U}^b$, then $\mathbf{U}^b$ is not a clique of $\mathcal{G}^*$. In this case, we must remove $\mathbf{C}$ from the tree. Therefore we connect $\mathbf{B}$ directly to $\mathbf{C}_{\text{par}}$, by-passing $\mathbf{C}$. This requires making $\mathbf{C}_{\text{par}}$ the parent of $\mathbf{B}$ (Line 5′a) and adding $\mathbf{B}$ to the list of $\mathbf{C}_{\text{par}}$'s children (Line 5′b). To find $\mathbf{B}_{\text{sep}}$ in $\mathcal{G}^*$, we have that

$$\mathbf{B} \cap \mathbf{U} \subseteq \mathbf{B} \cap \mathbf{C}_{\text{par}} \subseteq \mathbf{B} \cap \mathbf{C} = (\mathbf{B} \cap \mathbf{U}) \cup \{b\}, \tag{5.1}$$

where the second relation is guaranteed by the junction property for $\mathcal{G}$. Since $b \notin \mathbf{C}_{\text{par}}$, we conclude that $\mathbf{B}_{\text{sep}} = \mathbf{B} \cap \mathbf{C}_{\text{par}} = (\mathbf{B} \cap \mathbf{C}) \setminus \{b\}$. Since b is no longer in $\mathbf{B}_{\text{sep}}$, it must be in $\mathbf{B}_{\text{res}}$. Finally, we need to connect each $\mathbf{D} \in \vec{\mathbf{C}}_{\text{ch}}$ to a new clique. We choose to connect $\mathbf{D}$ to $\mathbf{A}$ or $\mathbf{B}$, based on whether or not $b \in \mathbf{D}$. It is easy to verify that the separator and residual for each $\mathbf{D}$ remains unchanged and hence that the junction property is maintained. □

We next show the second half of the algorithm that is used when neither $\mathbf{U}^a$ nor $\mathbf{U}^b$ is contained in $\mathbf{C}_{\text{par}}$.

Algorithm 5.2.4 (Drop edge: $|\mathbf{C}_{\text{res}}| > 1$)**.**

1. *Set* $\mathbf{A} \leftarrow \mathbf{B} \leftarrow$ NULL.
2. *If* $b \in \mathbf{C}_{\text{sep}}$*, then swap values of* a *and* b.
3. *For* $\mathbf{D} \in \vec{\mathbf{C}}_{\text{ch}}$*:*
   (a) *If* $|\mathbf{D}_{\text{sep}}| = |\mathbf{C}| - 1$*:*
      i. *If* $a \in \mathbf{D}_{\text{sep}}$ *and* $\mathbf{A} ==$ NULL*, then set* $\mathbf{A} \leftarrow \mathbf{D}$ *and remove* $\mathbf{A}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$.
         - *(stop iterating if* $\mathbf{B}$ *has been found as well)*
      ii. *If* $b \in \mathbf{D}_{\text{sep}}$ *and* $\mathbf{B} ==$ NULL*, then set* $\mathbf{B} \leftarrow \mathbf{D}$ *and remove* $\mathbf{B}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$.
         - *(stop iterating if* $\mathbf{A}$ *has been found as well)*
4. *If* $\mathbf{A} ==$ NULL*, then:*
   (a) *Set* $\mathbf{A} \leftarrow \mathbf{C}$.
   (b) *Remove* b *from* $\mathbf{A}$ *and* $\mathbf{A}_{\text{res}}$.
   (c) *If* $\mathbf{B} ==$ NULL*, then:*
      i. *Create a new clique* $\mathbf{B} = \mathbf{U}^b$*, with* $\mathbf{B}_{\text{sep}} = \mathbf{U}$ *and* $\mathbf{B}_{\text{res}} = \{b\}$.
      ii. *Set* $\mathbf{B}_{\text{par}} \leftarrow \mathbf{A}$ *and insert* $\mathbf{B}$ *into* $\vec{\mathbf{A}}_{\text{ch}}$.
   (b′) *else move* b *from* $\mathbf{B}_{\text{sep}}$ *to* $\mathbf{B}_{\text{res}}$.

(d) *For* $\mathbf{D} \in \vec{\mathbf{A}}_{\text{ch}}$*:*

i. *If* $b \in \mathbf{D}_{\text{sep}}$*, then move* $\mathbf{D}$ *from* $\vec{\mathbf{A}}_{\text{ch}}$ *to* $\vec{\mathbf{B}}_{\text{ch}}$*.*

4′. *else* ($\mathbf{A}! = \text{NULL}$)*:*

(a) *Swap* $\mathbf{A}_{\text{sep}}$ *and* $\mathbf{C}_{\text{sep}}$ *and set* $\mathbf{A}_{\text{res}} = \mathbf{A} \setminus \mathbf{A}_{\text{sep}}$*.*

(b) *Set* $\mathbf{A}_{\text{par}} \leftarrow \mathbf{C}_{\text{par}}$ *and insert* $\mathbf{A}$ *into* $\vec{\mathbf{C}}_{\text{par}_{\text{ch}}}$*.*

(c) *If* $\mathbf{B} ==$ NULL*, then*

i. *Set* $\mathbf{B} \leftarrow \mathbf{C}$*.*

ii. *Remove* a *from* $\mathbf{B}$ *and* $\mathbf{B}_{\text{sep}}$

iii. *Set* $\mathbf{B}_{\text{res}} \leftarrow \{r\}$*.*

iv. *For each* $\mathbf{D} \in \vec{\mathbf{C}}_{\text{ch}}$*, if* $a \in \mathbf{D}$*, then set* $\mathbf{D}_{\text{par}} \leftarrow \mathbf{A}$ *and move* $\mathbf{D}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$ *to* $\vec{\mathbf{A}}_{\text{ch}}$*.*

(b′) *else:*

i. *Move* b *from* $\mathbf{B}$ *to* $\mathbf{B}_{\text{res}}$*.*

ii. *For each* $\mathbf{D} \in \vec{\mathbf{C}}_{\text{ch}}$

* *If* $b \in \mathbf{D}$ *then set* $\mathbf{D}_{\text{par}} \leftarrow \mathbf{B}$ *and move* $\mathbf{D}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$ *to* $\vec{\mathbf{B}}_{\text{ch}}$*, else set* $\mathbf{D}_{\text{par}} \leftarrow \mathbf{A}$ *and move* $\mathbf{D}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$ *to* $\vec{\mathbf{A}}_{\text{ch}}$*.*

(d) *Set* $\mathbf{B}_{\text{par}} \leftarrow \mathbf{A}$ *and move* $\mathbf{B}$ *from* $\vec{\mathbf{C}}_{\text{par}_{\text{ch}}}$ *to* $\vec{\mathbf{A}}_{\text{ch}}$*.*

Proof. Line 2 simplifies the algorithm by allowing us to assume $b \notin \mathbf{C}_{\text{par}}$. The loop in Line 3 finds sets $\mathbf{A}$ and $\mathbf{B}$ that contain $\mathbf{U}^a$ and $\mathbf{U}^b$, or else determines that such sets do not exist in $\mathcal{G}$. If $\mathbf{A}$ does not exist, then we simply create the clique $\mathbf{A} = \mathbf{U}^a$ by removing b

from $\mathbf{C}$ and $\mathbf{C}_{\text{res}}$. If $\mathbf{B}$ also does not exist, then we must create a new clique containing the appropriate nodes, separator and residual. We then connect this clique to $\mathbf{A}$. On the other hand, if $\mathbf{B}$ does exist, then we use Equation 5.1 to see that $\mathbf{B} \cap \mathbf{A} = (\mathbf{B} \cap \mathbf{C}) \setminus \{b\}$. In this case, $\mathbf{B}$ is already connected to $\mathbf{A}$, so no edges need to be changed. Whether or not $\mathbf{B}$ exists, we must move any children in $\vec{\mathbf{A}}_{\text{ch}}$ if they contain b in order to maintain the junction property. It is easy to see that no other separators or residuals need to change, so the junction property is preserved in the updated tree.

We now examine the lines beginning at Line $4'$, for the case where $\mathbf{A} \supset \mathbf{U}^a$ does exist in $\mathcal{G}$. Note that

$$\mathbf{U}^a \cap \mathbf{C}_{\text{par}} \subseteq \mathbf{A} \cap \mathbf{C}_{\text{par}} \subseteq \mathbf{C} \cap \mathbf{C}_{\text{par}} = \mathbf{U}^a \cap \mathbf{C}_{\text{par}}, \tag{5.2}$$

whence $\mathbf{A} \cap \mathbf{C}_{\text{par}} = \mathbf{C} \cap \mathbf{C}_{\text{par}}$. Therefore, we simply swap the two separators and update the residual of $\mathbf{A}$ by its definition. If $\mathbf{B} \supseteq \mathbf{U}^b$ does not exist, then we simply create $\mathbf{B}$ from $\mathbf{C}$ by removing a. Recall that we swapped the separators of $\mathbf{A}$ and $\mathbf{C}$, so $\mathbf{C}_{\text{sep}} = \mathbf{U}^a$ now. Therefore, we need to remove a from this separator so that $\mathbf{B}_{\text{sep}} = \mathbf{C}_{\text{sep}} = \mathbf{U}$. We also update the residual to contain the single element b. As usual, we need to move any children of $\mathbf{C}$ that contain a so that they are children of $\mathbf{A}$. Once again, it is easy to see that the separators and residuals remain the same for these sets, so this does not break the junction property. If $\mathbf{B}$ and $\mathbf{A}$ both exist, then we simply collapse the tree by removing $\mathbf{C}$ as in Algorithm 5.2.3. $\square$

To complete our algorithms for dropping edges, we must also maintain the vector of pointers for each element. This is quite simple, especially in comparison to updating the oriented junction tree. In the event that we remove a from $\mathbf{C}$, we simply check if v_a references $\mathbf{C}$. If it does, then we set v_a to reference whichever clique contains $\mathbf{U}^a$ in $\mathcal{G}^*$. We do likewise when we remove b from $\mathbf{C}$. The only other time we remove elements from $\mathbf{C}$ - and thus potentially invalidate $\vec{v}$ - is when we collapse the graph and remove $\mathbf{C}$. In this situation, it is simple to iterate over each $c \in \mathbf{C}$ and set v_c to reference either $\mathbf{A}$ or $\mathbf{B}$ if necessary. Both cliques contain $\mathbf{U}$ and therefore, the reference is valid.

As a technical note, we could simplify the algorithms for dropping an edge by simply removing a from $\mathbf{C}$, creating a new clique $\mathbf{C}^* = \mathbf{C} \setminus \{a\}$, and then collapsing the graph if needed. In contrast, our algorithm is more complicated, but more efficient because we only need to create a new clique from scratch in some cases and we make the minimal number of updates.

We also make one small change to the updating algorithm that isn't strictly necessary. Namely, if removing the edge (a, b) causes a and b to be disconnected, instead of having $\mathbf{B}$ as a child of $\mathbf{A}$, we make it a child of the empty root clique $\mathbf{C}_0$. Most importantly, this keeps the graph from becoming too *wide*. Since some of the algorithms we develop involve forming paths between a clique and $\mathbf{C}_0$, it is helpful to keep the number of edges small. Secondly, by making $\mathbf{C}_0$ the parent of $\mathbf{B}$, we keep all connected components well-separated at the top level. This means that if we ever need to quickly find all the connected components, we can do so. It turns out that this also makes displaying the graph cliques somewhat nicer since the connected components are written in contiguous pieces.

5.2.3 Adding an Edge

In this section we consider adding an edge (a, b) to a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ to obtain a new graph $\mathcal{G}^* = (\mathbf{V}, \mathbf{E} \cup \{(a, b)\})$. From Lemma 2.1.8, we know that testing if $\mathcal{G}^*$ is decomposable requires finding the shortest path between a clique $\mathbf{A}$ containing a and a clique $\mathbf{B}$ containing b. The explanation is easiest if we pretend to know $\mathbf{A}$ and $\mathbf{B}$. The process requires only a small fix to rectify this faulty assumption. If $\mathbf{A}$ is an ancestor of $\mathbf{B}$, then we can find a path from $\mathbf{B}$ to $\mathbf{A}$ by following $\mathbf{B}, \mathbf{B}_{\text{par}}, \mathbf{B}_{\text{par}_{\text{par}}}, \ldots$ until we reach $\mathbf{A}$, which is the first set containing a. The shortest path has a similar form if $\mathbf{B}$ is an ancestor of $\mathbf{A}$. On the other hand, if neither $\mathbf{A}$ nor $\mathbf{B}$ is the ancestor of the other, then we can find the shortest path by first finding the paths from $\mathbf{A}$ to $\mathbf{C}_0$ and from $\mathbf{B}$ to $\mathbf{C}_0$. Since $\mathbf{C}_0$ is an ancestor of every clique, this is easy. Note that both paths must intersect at some point, if only at $\mathbf{C}_0$. Letting $\mathbf{D}$ denote the earliest intersection, we can form the path from $\mathbf{A}$ to $\mathbf{D}$ to $\mathbf{B}$.

Thus far, we have pretended to know the endpoints, $\mathbf{A}$ and $\mathbf{B}$, of the shortest path. If we don't know these endpoints, then we simply start from the cliques $\mathbf{A}^*$ and $\mathbf{B}^*$ that are referenced by v_a and v_b. In the case where $\mathbf{A}$ is an ancestor of $\mathbf{B}$, the junction property implies that $\mathbf{B}$ is an ancestor of $\mathbf{B}^*$. Therefore, we simply find the path from $\mathbf{B}^*$ to $\mathbf{A}$. The last clique in this path that contains b must be $\mathbf{B}$, so we simply remove every set before $\mathbf{B}$ in the path. The process is analogous when $\mathbf{B}$ is an ancestor of $\mathbf{A}$. Recall that in Section 5.2.1, we decided to set v_a and v_b to be the highest-level clique in the tree that contain $\mathbf{A}$ and $\mathbf{B}$. Therefore, if $\mathbf{A}$ is an ancestor of $\mathbf{B}$, we see that v_b indeed points to

B. If **B** is an ancestor of **A**, then v_a points to **A**. In either case, we save the trouble of traveling through some unnecessary cliques. Similarly, if neither **A** nor **B** is an ancestor of the other, we can find the path from $\mathbf{A}^*$ to $\mathbf{B}^*$ and trim all but the last clique containing a and all but the last clique containing b. In this case, we see that v_a references **A** and v_b references **B**, so we save time on both ends.

The last detail we must handle is the fact that we do not know if **A** or **B** is an ancestor of the other. Therefore, the shortest path algorithm must be able to handle all three cases simultaneously. We do so by using flags that we update when we discover which case is true. We will also let $\mathbf{A}^*$ and $\mathbf{B}^*$ denote the cliques referenced by v_a and v_b. Again noting that the junction tree is oriented, it is helpful to partition the shortest path into $(p^a, p^b, \mathbf{D})$, where p^a is the path upward through **A**'s ancestors, p^b is defined similarly, and **D** is where the paths intersect. In doing so, we maintain the proper orientation toward the root clique in p^a and p^b.

Algorithm 5.2.5 (Shortest Path)**.**

1. *Set* $A_{\text{flag}} \leftarrow B_{\text{flag}} \leftarrow 0$
2. *Set* $\mathbf{C}^a = \mathbf{A}^*$*; set* $\mathbf{C}^b = \mathbf{B}^*$
3. *Set* $p^a = (\mathbf{A})$*; set* $p^b = (\mathbf{B})$*.*
4. *While* $A_{\text{flag}} == B_{\text{flag}} == 0$*:*
   (a) *If* $b \in \mathbf{C}^a_{\text{par}}$ *then set* $A_{\text{flag}} \leftarrow 2$
   (b) *elseif* $a \in \mathbf{C}^b_{\text{par}}$ *then set* $B_{\text{flag}} \leftarrow 2$

(c) *elseif* $\mathbf{C}^a_{\text{par}} == \mathbf{C}_0$ *then set* $A_{\text{flag}} \leftarrow 1$

(d) *elseif* $\mathbf{C}^b_{\text{par}} == \mathbf{C}_0$ *then set* $B_{\text{flag}} \leftarrow 1$

(e) *else*

i. *Set* $\mathbf{C}^a \leftarrow \mathbf{C}^a_{\text{par}}$*; set* $\mathbf{C}^b \leftarrow \mathbf{C}^b_{\text{par}}$

ii. *Append* $\mathbf{C}^a$ *to* p^a*; append* $\mathbf{C}^b$ *to* p^b

5. *If* $A_{\text{flag}} == 1$ *then:*

(a) *While* $B_{\text{flag}} == 0$*:*

i. *If* $a \in \mathbf{C}^b_{\text{par}}$ *then set* $B_{\text{flag}} \leftarrow 2$

ii. *elseif* $\mathbf{C}^b_{\text{par}} == \mathbf{C}_0$ *then set* $B_{\text{flag}} \leftarrow 1$

iii. *else set* $\mathbf{C}^b \leftarrow \mathbf{C}^b_{\text{par}}$*; append* $\mathbf{C}^b$ *to* p^b*.*

5′. *elseif* $B_{\text{flag}} == 1$ *then:*

(a) *While* $A_{\text{flag}} == 0$*:*

i. *If* $b \in \mathbf{C}^a_{\text{par}}$ *then set* $A_{\text{flag}} \leftarrow 2$

ii. *elseif* $\mathbf{C}^a_{\text{par}} == \mathbf{C}_0$ *then set* $A_{\text{flag}} \leftarrow 1$

iii. *else set* $\mathbf{C}^a \leftarrow \mathbf{C}^a_{\text{par}}$*; append* $\mathbf{C}^a$ *to* p^a*.*

6. *If* $A_{\text{flag}} == 2$ *then clear* p^b*; return* $(p^a, p^b, \mathbf{C}^a_{\text{par}})$*.*

7. *If* $B_{\text{flag}} == 2$ *then clear* p^a*; return* $(p^a, p^b, \mathbf{C}^b_{\text{par}})$

8. *While the last elements of* p^a *and* p^b *are equal:*

(a) *Set* $\mathbf{D}$ *to be the common last element.*

(b) Remove the last elements of p^a and p^b.

9. Return $(p^a, p^b, \mathbf{D})$

Proof. The while loop at Line 4 simultaneously constructs the path from $\mathbf{A}^*$ and $\mathbf{B}^*$ until p^a reaches $\mathbf{B}$, p^b reaches $\mathbf{A}$, or either reaches the root clique $\mathbf{C}_0$. The simple use of flags indicates why the loop stopped. A_{flag} is 1 if p^a reached $\mathbf{C}_0$, A_{flag} is 2 if p^a reached $\mathbf{B}$, and similarly for B_{flag}. Note that if the loop ends because p^a reached $\mathbf{B}$, then we have our shortest path and we are done. In this case, Lines 4c and 4d are not evaluated, so the B_{flag} remains equal to 0. The conditions at Line 5 and 5′ are FALSE and the algorithm returns the path at Line 6. Similarly, if p^b reached $\mathbf{A}$, then the conditions at Lines 5, 5′ and 6 are FALSE and the algorithm returns the path at Line 7. On the other hand, suppose that the while loop at Line 4 quit because one path, say p^a, reached $\mathbf{C}_0$. In this case, we know that $\mathbf{B}$ is not an ancestor of $\mathbf{A}$, but it may still be the case that $\mathbf{A}$ is an ancestor of $\mathbf{B}$. Therefore, if $A_{\text{flag}} == 1$, we finish constructing the path from $\mathbf{B}^*$ in Line 5a. Similarly, if p^b reached $\mathbf{C}_0$, we have $B_{\text{flag}} == 1$ and we finish constructing the path from $\mathbf{A}^*$ in Line 5′a. At this point we have fully constructed both p^a and p^b. If $\mathbf{B}$ is an ancestor of $\mathbf{A}$, then $A_{\text{flag}} == 2$ and p^a is the shortest path, which we return in Line 6. If not, and $\mathbf{A}$ is an ancestor of $\mathbf{B}$, then $B_{\text{flag}} == 2$ and p^b is the shortest path, which we return in Line 7. Finally, if neither clique is the ancestor of the other, then $A_{\text{flag}} == B_{\text{flag}} == 1$. In this case, we find the earliest point where these paths intersect and store this as $\mathbf{D}$. We trim both p^a and p^b from $\mathbf{D}$ onward, and construct the shortest path from p^a, p^b, and $\mathbf{D}$. □

Once the shortest path is found $(p^a, p^b, \mathbf{D})$, it is almost trivial to test if $\mathcal{G}^*$ is decomposable.

Algorithm 5.2.6 (Test for Edge Addition)**.**

1. *If* $\mathbf{D} == \mathbf{C}_0$*, then return* TRUE.
2. *Set* $\mathbf{S} \leftarrow \{\}$*; set* $i \leftarrow j \leftarrow 1$
3. *While* $i \leq |\mathbf{A}^*|$ *and* $j \leq |\mathbf{B}^*|$*:*
   (a) *If* $a_i < b_i$ *then set* $i \leftarrow i + 1$
   (b) *elseif* $b_i < a_i$ *then set* $j \leftarrow j + 1$,
   (c) *else set* $\mathbf{S} \leftarrow \mathbf{S} \cup \{a_i\}$, $i \leftarrow i + 1$, *and* $j \leftarrow j + 1$
4. *For each* $\mathbf{C} \in p^a$*, if* $|\mathbf{C}_{\text{sep}}| == |\mathbf{S}|$ *then return* TRUE.
5. *For each* $\mathbf{C} \in p^b$*, if* $|\mathbf{C}_{\text{sep}}| == |\mathbf{S}|$ *then return* TRUE.
6. *Return* FALSE.

Proof. By Lemma 2.1.8, we need to see if one of the separators on the path is equal to $\mathbf{A} \cap \mathbf{B}$. There are two special cases that can be solved very efficiently. We first note that if $\mathbf{D} = \mathbf{C}_0$, where $\mathbf{C}_0$ is the empty root clique, then $\mathbf{A} \cap \mathbf{B} = \emptyset$ is the separator between $\mathbf{C}_0$ both adjacent cliques on the graph. We test this case in Line 1 since it does not even require calculating the intersection. (This corresponds to a and b being disconnected, in which case it is trivial to see that the edge can be added.) If $\mathbf{S}$ is not empty, we resort to checking each separator in the shortest path. Recall that the shortest path is of the form

$$(\mathbf{A}, \mathbf{A}_{\text{par}}, \mathbf{A}_{\text{par}_{\text{par}}}, \ldots, \mathbf{D}, \ldots, \mathbf{B}_{\text{par}_{\text{par}}}, \mathbf{B}_{\text{par}}, \mathbf{B}). \tag{5.3}$$

Therefore, the separators on the path are $\{\mathbf{C}_{\text{sep}} : \mathbf{C} \in p^a \cup p^b\}$. We also note that for each clique $\mathbf{C} \in p^a \cup p^b$, the junction property ensures that $\mathbf{S} = \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{C}_{\text{sep}}$. Taking advantage of this fact, we see that $\mathbf{S} == \mathbf{C}_{\text{sep}}$ if and only if $|\mathbf{S}| == |\mathbf{C}_{\text{sep}}|$. □

At this point, we have decided if the edge (a, b) can be added to $\mathcal{G}$. We still need to determine a way to update the graph *locally* to find $\mathcal{G}^*$. Once again, let $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. When we add an edge between a and b, we can see that $\mathbf{U} = \mathbf{S} \cup \{a, b\}$ will be a clique in $\mathcal{G}^*$. Where should we connect $\mathbf{U}$ in the graph? We note that $\mathbf{U} \cap \mathbf{A} = \mathbf{S} \cup \{a\}$, but $\mathbf{S} \cup \{a\}$ is not guaranteed to be in all or even any of $\mathbf{C}_{\text{sep}}$ in the path. Therefore, we must connect $\mathbf{U}$ to $\mathbf{A}$ in order to maintain the junction property, but an analogous argument shows that we must connect $\mathbf{U}$ to $\mathbf{B}$. If $\mathbf{A}$ and $\mathbf{B}$ are neighbors in the junction tree, it is easy to see that we can simply insert $\mathbf{U}$ between them, so that $\mathbf{A} \sim \mathbf{U} \sim \mathbf{B}$. On the other hand, if $\mathbf{A}$ and $\mathbf{B}$ are not neighbors, then connecting $\mathbf{U}$ to both creates a loop in the junction tree. Therefore, we need a method to rearrange the tree so that $\mathbf{A}$ and $\mathbf{B}$ are neighbors.

Denote the shortest path from $\mathbf{A}$ to $\mathbf{B}$ by

$$(\mathbf{A} = \mathbf{A}^0, \mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^{n_a} = \mathbf{D} = \mathbf{B}^{n_b}, \ldots, \mathbf{B}^2, \mathbf{B}^1, \mathbf{B}^0 = \mathbf{B}). \tag{5.4}$$

Recall that Algorithm 5.2.6 finds a clique $\mathbf{C} \in p^a \cup p^b$ such that $\mathbf{C}_{\text{sep}} = \mathbf{S}$. It is a simple matter to extend the algorithm to return $\mathbf{C}$ and record if $\mathbf{C} \in p^a$ or $\mathbf{C} \in p^b$. Without loss

of generality, we assume $\mathbf{C} \in p^a$, say $\mathbf{C} = \mathbf{A}^i$. We rearrange the path by breaking the edge between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$ and joining $\mathbf{A}$ and $\mathbf{B}$. Thus, the path in Equation 5.4 becomes

$$(\mathbf{A}^{i+1}, \ldots, \mathbf{A}^{n_a} = \mathbf{D} = \mathbf{B}^{n_b}, \ldots, \mathbf{B}^1, \mathbf{B}^0 = \mathbf{B}, \mathbf{A} = \mathbf{A}^0, \mathbf{A}^1, \ldots, \mathbf{A}^i). \tag{5.5}$$

To check that the junction property holds, take any $\mathbf{U}$ and $\mathbf{U}'$ in the path. Note that if $\mathbf{U}$ and $\mathbf{U}'$ are on the same side of $\mathbf{A}^i$ in Equation 5.4, then any sets between them after the rearrangement were also between them before the rearrangement, so the junction property of the original path will still hold. Therefore, it is enough to show that the junction property holds if $\mathbf{U} \in \{\mathbf{A}, \ldots, \mathbf{A}^i\}$ and $\mathbf{U}' \in \{A^{i+1}, \ldots, \mathbf{B}\}$. In this case, the junction property for the original path implies that

$$\mathbf{S} = \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{U} \cap \mathbf{U}' \subseteq \mathbf{A}^i_{\text{sep}} = \mathbf{S}, \tag{5.6}$$

whence $\mathbf{U} \cap \mathbf{U}' = \mathbf{S}$. Finally, note that $\mathbf{A} \cap \mathbf{B} = \mathbf{S}$ by definition, and that every other separator in the new path is also a separator in the original path, which must contain $\mathbf{S}$ by the junction property.

Since the junction tree we utilize is oriented toward the root, we need to change the direction of some edges. For simplicity, we still assume that the break point for this rearrangement is $\mathbf{C} \in p^a$. Note that in the new path, the root is no longer an ancestor of $\mathbf{A}$ according to the internal representation, which still lists $\mathbf{A}_{\text{par}} = \mathbf{A}^1$, $\mathbf{A}^1_{\text{par}} = \mathbf{A}^2$, etc. We reach a dead end at $\mathbf{A}^i$ since it is no longer connected to $\mathbf{A}^{i+1}$! Therefore, we make $\mathbf{B}$ the parent of $\mathbf{A}$ and reverse the parent-child relationship for $\mathbf{A}, \ldots, \mathbf{A}^i$. Finally, we

note that the separator between $\mathbf{S}'$ between $\mathbf{A}^j$ and $\mathbf{A}^{j+1}$ is still the same, but after the rearrangement, $\mathbf{A}^j$ is the parent and so $\mathbf{S}' = \mathbf{A}^{j+1}_{\text{sep}}$ whereas in the previous arrangement we had $\mathbf{S}' = \mathbf{A}^{j}_{\text{sep}}$. All of these details are expressed in the next algorithm, where we denote the shortest path by $(p^a, p^b, \mathbf{D})$, the breaking point for the rearrangement by $\mathbf{C}$, and set $f = \mathbb{I}(\mathbf{C} \in p^a)$.

Algorithm 5.2.7 (Rearrange Junction Tree)**.**

1. *Set* i, v *such that* $\mathbf{C} = p_i^v$.
2. *Remove* $\mathbf{C}$ *from* $\vec{\mathbf{C}}_{\text{par}_{\text{ch}}}$.
3. *Set* $\mathbf{S} = \mathbf{C}_{\text{sep}}$.
4. *For* $j = i - 1, \ldots, 0$*:*
   (a) *Set* $\mathbf{F} \leftarrow p_j^v$*; set* $\mathbf{F}^+ \leftarrow p_{j+1}^v$.
   (b) *Remove* $\mathbf{F}$ *from* $\vec{\mathbf{F}}^+_{\text{ch}}$.
   (c) *Insert* $\vec{\mathbf{F}}^+_{\text{ch}}$ *into* $\vec{\mathbf{F}}_{\text{ch}}$*; set* $\mathbf{F}^+_{\text{par}} \leftarrow \mathbf{F}$.
   (d) *Set* $\mathbf{F}^+_{\text{sep}} \leftarrow \mathbf{F}_{\text{sep}}$*; set* $\mathbf{F}^+_{\text{res}} \leftarrow \mathbf{F} \setminus \mathbf{F}^+_{\text{sep}}$.
5. *If* $f == 0$*, then swap* $\mathbf{A}$ *and* $\mathbf{B}$.
6. *Set* $\mathbf{A}_{\text{par}} \leftarrow \mathbf{B}$*; Insert* $\mathbf{A}$ *into* $\vec{\mathbf{B}}_{\text{ch}}$.
7. *Set* $\mathbf{A}_{\text{sep}} \leftarrow \mathbf{S}$*; set* $\mathbf{A}_{\text{res}} \leftarrow \mathbf{A} \setminus \mathbf{A}_{\text{sep}}$.

Proof. This algorithm is a straightforward implementation of the preceding discussion. □

After applying Algorithms 5.2.5-5.2.7, we know that edge (a, b) can be added to the graph between the cliques $\mathbf{A}$ and $\mathbf{B}$. Recall that $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$, so $a \sim s \sim b$ for each $s \in \mathbf{S}$. Therefore, $\mathbf{C} = \mathbf{S} \cup \{a, b\}$ will be a clique in the updated graph $\mathcal{G}^*$. On the other hand, if $\mathbf{A} = \mathbf{S} \cup \{a\}$, then $\mathbf{A}$ will no longer be a clique. Once again using inclusion properties, we have that $\mathbf{S} \cup \{a\} \subseteq \mathbf{A}$ and we can test for equality by checking if $|\mathbf{A}| = |\mathbf{S}| + 1$. Similarly, if $|\mathbf{B}| = |\mathbf{S}| + 1$, then $\mathbf{B}$ is not a clique in $\mathcal{G}^*$. Hence, we have four possible cases to deal with depending on which of $\mathbf{A}$ and $\mathbf{B}$ remain cliques in $\mathcal{G}^*$.

Algorithm 5.2.8 (Add Edge)**.**

1. *If* $\mathbf{B}_{\text{par}} == \mathbf{A}$ *then swap* $\mathbf{A}$ *with* $\mathbf{B}$ *and* a *with* b.

2. *If* $|\mathbf{A}| == |\mathbf{S}| + 1$, *then:*

   (a) *If* $|\mathbf{B}| == |\mathbf{S}| + 1$ *then:*

      i. *Set* $\mathbf{C} \leftarrow \mathbf{B}$*; insert* a *into* $\mathbf{C}$.

      ii. *Insert* a *into* $\mathbf{C}_{\text{res}}$.

      iii. *Remove* $\mathbf{A}$ *from* $\vec{\mathbf{C}}_{\text{ch}}$

      iv. *For each* $\mathbf{D} \in \vec{\mathbf{A}}_{\text{ch}}$*, set* $\mathbf{D}_{\text{par}} \leftarrow \mathbf{C}$.

      v. *Splice* $\vec{\mathbf{A}}_{\text{ch}}$ *onto* $\vec{\mathbf{C}}_{\text{ch}}$.

      vi. *Delete* $\mathbf{A}$.

   (a') *else:*

      i. *Set* $\mathbf{C} \leftarrow \mathbf{A}$*; insert* b *into* $\mathbf{C}$.

      ii. *Insert* b *into* $\mathbf{C}_{\text{sep}}$.

2′. else:

(a) If $|\mathbf{B}| == |\mathbf{S}| + 1$ *then:*

i. Set $\mathbf{C} \leftarrow \mathbf{B}$*; insert* a *into* $\mathbf{C}$.

ii. Insert a *into* $\mathbf{C}_{\text{res}}$.

(a′) else:

i. Create a new set $\mathbf{C} = \mathbf{S} \cup \{a, b\}$.

ii. Set $\mathbf{C}_{\text{sep}} = \mathbf{S} \cup \{b\}$*; set* $\mathbf{C}_{\text{res}} \leftarrow \{a\}$.

iii. Remove $\mathbf{A}$ *from* $\mathbf{B}$.

iv. Insert $\mathbf{A}$ *into* $\vec{\mathbf{C}}_{\text{ch}}$*; set* $\mathbf{A}_{\text{par}} \leftarrow \mathbf{C}$.

v. Insert $\mathbf{C}$ *into* $\vec{\mathbf{B}}_{\text{ch}}$*; set* $\mathbf{C}_{\text{par}} \leftarrow \mathbf{B}$.

Proof. Line 1 simply allows us to assume that $\mathbf{B} = \mathbf{A}_{\text{par}}$. The conditional at Line 2a is the case when neither $\mathbf{A}$ nor $\mathbf{B}$ are cliques. In this case, $\mathcal{G}^*$ will have one fewer cliques than $\mathcal{G}$. Rather than remove $\mathbf{B}$ and create a new clique for $\mathbf{C}$. We simply update $\mathbf{B}$ to include a. The separator $\mathbf{C} \cap \mathbf{B}_{\text{par}} = (\mathbf{B} \cup \{a\}) \cap \mathbf{B}_{\text{par}}$. By the junction property, $a \notin \mathbf{B}_{\text{par}}$, so the separator is just $\mathbf{B} \cap \mathbf{B}_{\text{par}}$ and does not need to be changed. We also see that $\mathbf{C}_{\text{res}} = \mathbf{C} \setminus \mathbf{C}_{\text{sep}} = \mathbf{B}_{\text{sep}} \cup \{a\}$. Finally, we need to remove $\mathbf{A}$ from the tree, so its children should be connected to $\mathbf{C}$. Since $\mathbf{A} \subset C$, this does not break the junction property. The case where $\mathbf{B}$ is a clique in $\mathcal{G}^*$, but $\mathbf{A}$ is not, is treated by Line 2a′. In this case, we just update $\mathbf{A}$ to include b and realize that $\mathbf{C}_{\text{sep}} \cap \mathbf{B} = \mathbf{U} \cup \{b\} = \mathbf{A}_{\text{sep}} \cup \{b\}$. The residual $\mathbf{C}_{\text{res}} = \mathbf{A}_{\text{res}} = \{a\}$ remains unchanged. The case where $\mathbf{A}$ is a clique, but $\mathbf{B}$ is not, is handled by Line 2′a. In this case, we update $\mathbf{B}$ to find $\mathbf{C}$ exactly as in the first case (Line

2a), except that we do not move the cliques in $\vec{\mathbf{A}}_{\text{ch}}$ since we are not going to delete $\mathbf{A}$. The fourth and final case is when both $\mathbf{A}$ and $\mathbf{B}$ continue to be cliques in $\mathcal{G}^*$. In this case, we simply create the clique $\mathbf{C}$ from scratch. The separator is $\mathbf{C} \cap \mathbf{B} = (\mathbf{S} \cup \{a, b\}) \cap \mathbf{B} = \mathbf{S} \cup \{b\}$ and the residual is therefore $\mathbf{C} \setminus (\mathbf{S} \cup \{b\}) = \{a\}$. Finally, we insert $\mathbf{C}$ between $\mathbf{A}$ and $\mathbf{B}$. Thus, we set $\mathbf{C}_{\text{par}} = \mathbf{B}$ and include $\mathbf{C}$ in the list of $\mathbf{B}$'s children. In addition, $\mathbf{A}$ is no longer a child of $\mathbf{B}$. Instead it is a child of $\mathbf{C}$ and $\mathbf{A}_{\text{par}} = \mathbf{C}$. □

As a technical note, we could simplify Algorithm 5.2.8 by simply inserting a new clique $\mathbf{C}$ and collapsing the tree to remove $\mathbf{A}$ and/or $\mathbf{B}$ as necessary. In contrast, our algorithm is more complicated, but more efficient because we only need to create $\mathbf{C}$ from scratch in one out of four cases and we make the minimal number of updates. To complete our specification, we also need to update v, the vector of pointers for each element. Since we are adding elements to cliques rather than removing them, the only time the pointers may be invalidated is when $\mathbf{A}$ is deleted in 2a.vi. In this case, we simply need to check if v_u references $\mathbf{A}$ for each $u \in \mathbf{A}$. If it does, we alter it to reference $\mathbf{C}$ instead.

Chapter 6

Non-Parametric Mixtures with the HDP

We discussed in Chapter 3 how we can use Dirichlet Processes for constructing non-parametric mixture models. We continue to study this hierarchical model where $\theta_1, \ldots, \theta_n$ is a random sample from the random measure P is a Dirichlet process with precision ν and base measure H, and the datum $X_i|\theta_i$ is a noisy observation of θ_i with distribution $F(X_i; \theta_i)$ independently of everything else. In this chapter, we show an extension of this application by incorporating hyper Dirichlet processes. We consider a *hyper* Dirichlet mixture, meaning that the underlying Dirichlet process is hyper Markov. Alternatively, we can choose a graphical Dirichlet process which requires only that H is hyper Markov and $F(X_i; \cdot)$ is a family of Markov distributions. The resulting mixture is one in which the random observations satisfy the prescribed graphical structure within each mixture component. That is,

if we take the cluster membership into account, then the desired conditional independence constraints are satisfied within each component. Once again, we note that it is sufficient for this property if P is only a graphical Dirichlet process instead of a hyper Dirichlet process as shown by Theorem 4.5.3.

Throughout this chapter, for a probability measure $\mathrm{F}(\theta)$ with parameter θ, we will denote by $\frac{\partial}{\partial X}\mathrm{F}(X;\theta)$ its probability density function evaluated at the point X.

6.1 Hyper Dirichlet Mixture of Gaussians

As an illustration of a hyper Dirichlet process, we will consider a hyper Dirichlet mixture of Gaussians. This model is a multivariate and graphical extension of the work presented by Escobar and West (1995). In their model, $F(\cdot|\theta)$ is a univariate Gaussian distribution, where $\theta = (\mu, \sigma^2)$ is the mean and variance. For $\theta \sim H$

$$\sigma^2 \sim \mathrm{IG}(s/2, S/2) \tag{6.1}$$

$$\mu|\sigma^2 \sim \mathrm{N}(m, \tau V) \tag{6.2}$$

which we call a Normal $\times$ Inverse-Gamma distribution. Given $(\mu, \sigma^2) \sim H$, let $X_1, \ldots, X_n$ be a random sample from $\mathrm{N}(\mu, \sigma^2)$. The posterior distribution of (μ, σ^2) is also a Normal $\times$ Inverse-Gamma law with

$$\sigma^2|(X_1,\ldots,X_n) \sim \text{IG}\left(\frac{n+s}{2}, \frac{S+\sum_{i=1}^n (X_i-m)^2/(1+n\tau)}{2}\right) \tag{6.3}$$

$$\mu|(\sigma^2, X_1,\ldots,X_n) \sim \text{N}\left(\frac{\tau\sum_{i=1}^n X_i + m}{1+n\tau}, \frac{\tau}{1+n\tau}\sigma^2\right) \tag{6.4}$$

6.1.1 Moving to Multiple Dimensions

The Inverse-Wishart distribution is a multivariate generalization of the Inverse-Gamma distribution. Therefore, we replace the Normal $\times$ Inverse-Gamma distribution with a multivariate Normal $\times$ Inverse-Wishart distribution. The p-variate Normal distribution with mean m and covariance τV has density

$$\frac{\partial}{\partial\mu}\text{N}(\mu; m, \tau V) = (2\pi\tau)^{-p/2}|V|^{-1/2}\exp\left\{\frac{1}{2\tau}(\mu-m)^T V^{-1}(\mu-m)\right\}, \tag{6.5}$$

where $\mu, m \in \mathbb{R}^p$, τ is a positive real number, and V is a $p \times p$ positive definite matrix. Using the parameterization of Dawid (1981), the p-variate hyper Inverse-Wishart with d degrees of freedom and location (mean) D has density

$$\frac{\partial}{\partial V}\text{IW}(V; d, D) = \frac{|D|^{(d+p-1)/2}|V|^{-(d/2+p)}\exp\{-\text{tr}(DV^{-1})/2\}}{2^{(d+p-1)p/2}\Gamma_p((d+p-1)/2)}, \tag{6.6}$$

where d is a positive real number, D and V are positive definite $p\times p$ matrices, $\text{tr}(\cdot)$ denotes the trace function, and

$$\Gamma_p(x) = \pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma(x+(1-i)/2) \tag{6.7}$$

is the multivariate gamma function.

To incorporate a specific decomposable graphical model, we need to use the hyper Markov versions of these distributions as specified by Dawid and Lauritzen (1993). The hyper Normal distribution is the (hyper) Markov combination of consistent marginal Normal distributions for each clique. Let $\mathbf{A}$ and $\mathbf{B}$ be subsets of $\mathbf{V}$ with $\mathbf{S} = \mathbf{A} \cap \mathbf{B}$. For a $p \times p$ matrix V, let $V_{\mathbf{AB}} = \{v_{ij} : i \in \mathbf{A}, j \in \mathbf{B}\}$ be the submatrix with rows in $\mathbf{A}$ and columns in $\mathbf{B}$. We shall also understand $V_{\mathbf{AB}}^{-1}$ to mean the inverse of $V_{\mathbf{AB}}$ (rather than a submatrix of the entire inverse V^{-1}.) For convenience, we define $V_{\mathbf{A}} = V_{\mathbf{AA}}$. The marginals $\mathrm{N}(m_{\mathbf{A}}, V_{\mathbf{A}})$ for $\mu_{\mathbf{A}}$ and $\mathrm{N}(m_{\mathbf{B}}, V_{\mathbf{B}})$ for $\mu_{\mathbf{B}}$ are (hyper) consistent if the marginal value of $(m_{\mathbf{S}}, V_{\mathbf{S}})$ is the same whether obtained as submatrices of $(m_{\mathbf{A}}, V_{\mathbf{A}})$ or of $(m_{\mathbf{B}}, V_{\mathbf{B}})$. Likewise, the Inverse-Wishart marginals $\mathrm{IW}(d, D_{\mathbf{A}})$ and $\mathrm{IW}(d, D_{\mathbf{B}})$ are hyper consistent when the submatrix $D_{\mathbf{A}\cap\mathbf{B}}$ is the same whether obtained from $D_{\mathbf{A}}$ or $D_{\mathbf{B}}$.

Let $\mathcal{G}$ be a decomposable graph with cliques $\mathscr{C}$ and separators $\mathscr{S}$. The hyper Normal distribution has density function

$$\frac{\partial}{\partial \mu}\mathrm{HN}_{\mathcal{G}}(\mu; m, \tau V) = \prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial \mu_{\mathbf{A}}}\mathrm{N}(\mu_{\mathbf{A}}; m_{\mathbf{A}}, \tau V_{\mathbf{A}}), \tag{6.8}$$

where

$$\prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial \mu_{\mathbf{A}}}\mathrm{N}(\mu_{\mathbf{A}}; m_{\mathbf{A}}, \tau V_{\mathbf{A}}) = \frac{\prod_{\mathbf{C}\in\mathscr{C}} \frac{\partial}{\partial \mu_{\mathbf{C}}}\mathrm{N}(\mu_{\mathbf{C}}; m_{\mathbf{C}}, \tau V_{\mathbf{C}})}{\prod_{\mathbf{S}\in\mathscr{S}} \frac{\partial}{\partial \mu_{\mathbf{S}}}\mathrm{N}(\mu_{\mathbf{S}}; m_{\mathbf{S}}, \tau V_{\mathbf{S}})}. \tag{6.9}$$

For a collection of consistent clique marginal Inverse-Wishart laws, the hyper Inverse-Wishart law has density function

$$\frac{\partial}{\partial V}\text{HIW}(V; d, D) = \prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial V_{\mathbf{A}}}\text{IW}(V_{\mathbf{A}}; d, D_{\mathbf{A}}). \tag{6.10}$$

We are now ready to define a hyper Dirichlet process mixture of Gaussians.

$$P \sim HDP_{\mathcal{G}}(\nu H_{\mathcal{G}}) \tag{6.11}$$

$$(\mu_i, V_i)|P \sim P \tag{6.12}$$

$$X_i|(\mu_i, V_i) \sim \text{HN}_{\mathcal{G}}(\mu_i, V_i), \tag{6.13}$$

where $H_{\mathcal{G}}$ is a $\text{HN}_{\mathcal{G}}\times$ $\text{HIW}_{\mathcal{G}}$ law with density $\frac{\partial}{\partial\mu_i}\text{HN}_{\mathcal{G}}(\mu_i; m, \tau V_i)\cdot \frac{\partial}{\partial V_i}\text{HIW}_{\mathcal{G}}(V_i; d, D)$. We point out that we can replace the hyper Dirichlet process with a graphical Dirichlet process if needed. For example, if $\mathcal{G}$ is disconnected, the hyper Dirichlet process requires independent processes for each component, but the graphical Dirichlet process only requires one process for the entire graph. The hyper parameters for this model are ν, m, τ, d, D. Here, ν is the precision of the underlying Dirichlet process precision and indirectly specifies a prior for the number of mixture components (Antoniak, 1974). m is the center of the means for each component, and τ specifies how far apart the component locations are relative to the variance within the components. Escobar and West (1995) show empirically that the prior number of modes increases stochastically with τ. This is intuitive because when the components are farther apart, it decreases the chance that the mode for one component is occluded by one or more other component density curves. The examples in this chapter reveal that this is true in the multivariate case as well. Interestingly, we will see that larger

values of τ lead to more modes, but fewer mixture components in the posterior distribution, relative to smaller values of τ.

Our first task is to show that $H_{\mathcal{G}}$ is indeed a hyper Markov law, so that the specified model is a hyper Dirichlet mixture. In fact, as the next theorem shows, $H_{\mathcal{G}}$ is actually *strong* hyper Markov.

Theorem 6.1.1. *For $m \in \mathbb{R}^p, \tau, d > 0$, and a $p \times p$ positive definite matrix D, the $HN_{\mathcal{G}} \times HIW_{\mathcal{G}}$ law for (μ, V) with density $\frac{\partial}{\partial \mu} HN_{\mathcal{G}}(\mu; m, \tau V) \cdot \frac{\partial}{\partial V} HIW_{\mathcal{G}}(V; d, D)$ is strong hyper Markov.*

Proof. Let $\mathbf{A} \subset \mathbf{C}$ be a subset of any clique and set $\mathbf{B} = \mathbf{C} \setminus \mathbf{A}$. By Proposition 2.3.6, it suffices to show that the set of conditional distributions for $X_{\mathbf{B}}$ given $X_{\mathbf{A}} = x_{\mathbf{A}}$ are independent of $(\mu_{\mathbf{A}}, V_{\mathbf{A}})$. Recall that

$$X_{\mathbf{B}} | (X_{\mathbf{A}} = x_{\mathbf{A}}) \sim \mathrm{N}\Big(\mu_{\mathbf{B}} + V_{\mathbf{BA}} V_{\mathbf{A}}^{-1} (X_{\mathbf{A}} - \mu_{\mathbf{A}}), V_{\mathbf{B}} - V_{\mathbf{BA}} V_{\mathbf{A}}^{-1} V_{\mathbf{AB}}\Big), \tag{6.14}$$

so we must show that the mean of this Normal distribution (denoted $\mu_{\mathbf{B}|\mathbf{A}}(x_{\mathbf{A}})$) and the covariance matrix (denoted $V_{\mathbf{B}|\mathbf{A}}$) are independent of $(\mu_{\mathbf{A}}, V_{\mathbf{A}})$. Invoking the usual conditional distribution for a multivariate Normal vector, we first note that

$$\mu_{\mathbf{B}} | (\mu_{\mathbf{A}}, V) \sim \mathrm{N}\big(m_{\mathbf{B}} + \tau V_{\mathbf{BA}} (\tau V_{\mathbf{A}})^{-1} (\mu_{\mathbf{A}} - m_{\mathbf{A}}), \tau V_{\mathbf{B}|\mathbf{A}}\Big), \tag{6.15}$$

whence

$$\mu_{\mathbf{B}} - V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}\mu_{\mathbf{A}}|(\mu_{\mathbf{A}}, V) \sim \mathrm{N}\Big(m_{\mathbf{B}} + \tau V_{\mathbf{BA}}(\tau V_{\mathbf{A}})^{-1}(\mu_{\mathbf{A}} - m_{\mathbf{A}}) - V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}\mu_{\mathbf{A}}, \tau V_{\mathbf{B}|\mathbf{A}}\Big). \tag{6.16}$$

The τs cancel each other out, so that the terms containing $\mu_{\mathbf{A}}$ sum to 0. Therefore, $\mu_{\mathbf{B}} - V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}\mu_{\mathbf{A}}$ depends on $\mu_{\mathbf{A}}$ and V only through $V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}$ and $V_{\mathbf{B}|\mathbf{A}}$. Because $\mu_{\mathbf{B}|\mathbf{A}}(x_{\mathbf{A}}) = \mu_{\mathbf{B}} - V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}\mu_{\mathbf{A}} + V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}x_{\mathbf{A}}$, we infer that

$$\mu_{\mathbf{B}|\mathbf{A}}(x_{\mathbf{A}}) \perp\!\!\!\perp (\mu_{\mathbf{A}}, V_{\mathbf{A}}) \mid (V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}, V_{\mathbf{B}|\mathbf{A}}). \tag{6.17}$$

To complete our proof, we will also need the following two properties:

$$V_{\mathbf{B}|\mathbf{A}} \perp\!\!\!\perp V_{\mathbf{A}} \tag{6.18}$$

$$V_{\mathbf{BA}}V_{\mathbf{A}}^{-1} \perp\!\!\!\perp V_{\mathbf{A}} \mid V_{\mathbf{B}|\mathbf{A}} \tag{6.19}$$

We show both of these relations by citing the generative method of Carvalho et al. (2007) for creating random (hyper) Inverse Wishart matrices. From their construction, we see that

$$V_{\mathbf{B}|\mathbf{A}}|V_{\mathbf{A}} \sim \mathrm{IW}(d + |\mathbf{B}|, D_{\mathbf{B}} - D_{\mathbf{BA}}D_{\mathbf{A}}^{-1}D_{\mathbf{AB}}) \tag{6.20}$$

and

$$V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}|(V_{\mathbf{A}}, V_{\mathbf{B.A}}) \sim \mathrm{MN}(D_{\mathbf{BA}}D_{\mathbf{A}}^{-1}, V_{\mathbf{B}|\mathbf{A}}, D_{\mathbf{A}}^{-1}). \tag{6.21}$$

Neither of these distributions depends on $V_{\mathbf{A}}$, so the conditional independence properties of Equations 6.18 and 6.19 are true. Furthermore, we may incorporate $\mu_{\mathbf{A}}$ into these relationships because the distribution of $\mu_{\mathbf{A}}|(V_{\mathbf{B}|\mathbf{A}}, V_{\mathbf{BA}}V_{\mathbf{A}}^{-1}, V_{\mathbf{A}})$ is a Normal random variable with mean $m_{\mathbf{A}}$ and variance $\tau V_{\mathbf{A}}$. Therefore, we have shown

$$V_{\mathbf{B}|\mathbf{A}} \perp\!\!\!\perp (\mu_{\mathbf{A}}, V_{\mathbf{A}}) \tag{6.22}$$

$$V_{\mathbf{BA}}V_{\mathbf{A}}^{-1} \perp\!\!\!\perp (\mu_{\mathbf{A}}, V_{\mathbf{A}}) \mid V_{\mathbf{B}|\mathbf{A}} \tag{6.23}$$

Equations 6.17 and 6.23 together imply

$$\mu_{\mathbf{B}|\mathbf{A}}(x_{\mathbf{A}}) \perp\!\!\!\perp (\mu_{\mathbf{A}}, V_{\mathbf{A}}) \mid V_{\mathbf{B}|\mathbf{A}}. \tag{6.24}$$

This equation, along with Equation 6.22, implies that

$$(\mu_{\mathbf{B}|\mathbf{A}}, V_{\mathbf{B}|\mathbf{A}}) \in (\mu_{\mathbf{A}}, V_{\mathbf{A}}), \tag{6.25}$$

which is what we needed to show. □

6.1.2 Gibbs Sampling for the Hyper Dirichlet Mixture of Gaussians

In Section 3.3.1, we showed that we can construct a Gibbs sampler to sample from the posterior distribution of $\vec{\theta}|\vec{X}$. This construction requires us to compute the marginal density of X_i, when $\theta_i \sim H_{\mathcal{G}}$. We also need to generate random deviates from the posterior $\theta_i|X_i = x_i$. In order to utilize MacEachern's (1994) re-mixing scheme we also need to

sample from the posterior of the unique θ_i^* given all members of that cluster partition. As we shall see, the $\mathrm{HN}_{\mathcal{G}}\times$ $\mathrm{HIW}_{\mathcal{G}}$ prior is conjugate under sampling from the Normal distribution, which simplifies these calculations. We can compute the marginal density of X_i analytically and generate all necessary random variable generation directly.

Since the $\mathrm{HN}_{\mathcal{G}}\times$ $\mathrm{HIW}_{\mathcal{G}}$ law is strong hyper Markov, we can compute posterior updates locally for each clique. Therefore, we proceed for the time being by focusing on complete sets, for which the $\mathrm{HN}_{\mathcal{G}}\times$ $\mathrm{HIW}_{\mathcal{G}}$ law is equal to the simple N $\times$ IW law. We will consider $X_1, \ldots, X_n$ to be observations which are independent with common parameters (μ, V), where (μ, V) have a N $\times$ IW prior law. We begin by finding the conditional distribution of μ given the observations and V.

$$
\begin{aligned}
& f(X_1, \ldots, X_n, \mu | V) \\
&= \prod_{i=1}^{n} \frac{\partial}{\partial X_i} \mathrm{N}(X_i; \mu, V) \cdot \frac{\partial}{\partial \mu} \mathrm{N}(\mu; m, \tau V) && (6.26) \\
&\propto \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \Big((X_i - \mu)^T V^{-1} (X_i - \mu) \Big) - \frac{1}{2\tau} \Big((\mu - m)^T V^{-1} (\mu - m) \Big) \right\} && (6.27) \\
&\propto \exp\left\{ -\frac{1}{2} \left[\left(n + \frac{1}{\tau} \right) \mu^T V^{-1} \mu - \mu^T V^{-1} \left(\sum_{i=1}^{n} X_i + \frac{1}{\tau} m \right) \right.\right. \\
&\qquad \left.\left. - \left(\sum_{i=1}^{n} X_i + \frac{1}{\tau} m \right)^T V^{-1} \mu + \sum_{i=1}^{n} (X_i^T V^{-1} X_i) + \frac{1}{\tau} m^T V^{-1} m \right] \right\} && (6.28)
\end{aligned}
$$

$$
\begin{aligned}
\propto \quad & \exp\left\{-\frac{1}{2}\left(\frac{n\tau+1}{\tau}\right)\left[\mu^T V^{-1}\mu + \left(\frac{\tau}{n\tau+1}\right)\sum_{i=1}^{n}(X_i^T V^{-1} X_i)\right.\right. \\
& -\mu^T V^{-1}\left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right) - \left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right)^T V^{-1}\mu \\
& + \left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right)^T V^{-1}\left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right) \\
& - \left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right)^T V^{-1}\left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right) \\
& \left.\left. + \left(\frac{1}{n\tau+1}\right) m^T V^{-1} m\right]\right\} \qquad (6.29) \\
\propto \quad & \exp\left\{-\frac{1}{2}\left(\frac{n\tau+1}{\tau}\right)\left[\left(\mu - \frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right)^T V^{-1}\left(\mu - \frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}\right)\right]\right\} \\
& \cdot \exp\left\{-\frac{1}{2}\left[\sum_{i=1}^{n}(X_i^T V^{-1} X_i) - \frac{\left(\tau\sum_{i=1}^{n} X_i + m\right)^T V^{-1}\left(\tau\sum_{i=1}^{n} X_i + m\right)}{n\tau^2+\tau}\right.\right. \qquad (6.30) \\
& \left.\left. + \frac{m^T V^{-1} m}{\tau}\right]\right\}
\end{aligned}
$$

From the first factor in Equation 6.30, we see

$$
\mu|(X_1,\ldots,X_n,V) \sim \mathrm{N}\left(\frac{\tau\sum_{i=1}^{n} X_i + m}{n\tau+1}, \quad \frac{\tau}{n\tau+1}V\right). \qquad (6.31)
$$

Analyzing further, the second factor of Equation 6.30 leads to the distribution of $X_1,\ldots,X_n$ given their common covariance matrix V. Intuitively, the X_i are exchangeable with identical Normal distributions, but are no longer independent.

$$
\begin{aligned}
& f(X_1, \ldots, X_n | V) \\
& \propto \quad \exp\left\{ -\frac{1}{2} \left[\sum_{i=1}^{n} (X_i^T V^{-1} X_i) + \frac{m^T V^{-1} m}{\tau} \right.\right. \\
& \qquad \left.\left. - \frac{\left(\tau \sum_{i=1}^{n} X_i + m\right)^T V^{-1} \left(\tau \sum_{i=1}^{n} X_i + m\right)}{n\tau^2 + \tau} \right] \right\} \qquad (6.32) \\
& \propto \quad \exp\left\{ -\frac{1}{2} \left[\left(1 - \frac{\tau}{n\tau + 1}\right) \sum_{i=1}^{n} (X_i^T V^{-1} X_i) - \frac{1}{n\tau + 1} \sum_{i=1}^{n} X_i^T V^{-1} m \right.\right. \\
& \qquad - \frac{1}{n\tau + 1} m^T V^{-1} \sum_{i=1}^{n} X_i - \frac{\tau}{n\tau + 1} \sum_{i=1}^{n} \sum_{j \neq i} X_i^T V^{-1} X_j \qquad (6.33) \\
& \qquad \left.\left. + \frac{n}{n\tau + 1} m^T V^{-1} m \right] \right\}
\end{aligned}
$$

Let X and M be the $n \times p$ matrices, where the i^{th} row of X is X_i and each row of M is m. Let I_n denote the $n \times n$ identity matrix and J_{np} the $n \times p$ matrix of 1s. We note that $Q^{-1} = I_n - \tau/(n\tau + 1) J_{nn}$ is well-defined since the RHS is non-singular. In fact, it is easy to verify that $Q = I_n + \tau J_{nn}$. It is easy to see by comparing coefficients in the summands that we can rewrite Equation 6.33 by

$$
f(X_1, \ldots, X_n | V) \propto \exp\left\{ -\frac{1}{2} \mathrm{tr}\Big(V^{-1} (X - M)^T Q^{-1} (X - M) \Big) \right\}. \qquad (6.34)
$$

To see this, we will define $Y = X - M$, $U = Y^T Q^{-1}$, $S = Y^T Q^{-1} Y$, and $R = V^{-1} Y^T Q^{-1} Y$. We have the following equalities, which follow from the very definitions of cross product, transpose, and trace:

$$U_{ki} = \sum_{j=1}^{n} Y_{jk} Q_{ji}^{-1} \tag{6.35}$$

$$S_{kh} = \sum_{i=1}^{n} U_{ki} Y_{ih} = \sum_{i=1}^{n}\sum_{j=1}^{n} Y_{jk} Q_{ji}^{-1} Y_{ih} \tag{6.36}$$

$$R_{gh} = \sum_{k=1}^{p} V_{gk}^{-1} S_{kh} = \sum_{k=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n} V_{gk}^{-1} Y_{jk} Q_{ji}^{-1} Y_{ih} \tag{6.37}$$

$$\mathrm{tr}(R) = \sum_{g=1}^{p} R_{gg} = \sum_{g=1}^{p}\sum_{k=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n} V_{gk}^{-1} Y_{jk} Q_{ji}^{-1} Y_{ig} \tag{6.38}$$

Note that $Q_{ij}^{-1} = \mathbb{I}(i=j) - \tau/(n\tau+1)$, where $\mathbb{I}(i=j)$ is 1 if $i=j$ and 0 otherwise.

$$\begin{aligned}
&\mathrm{tr}(R) \\
&= \sum_{i=1}^{n}\sum_{g=1}^{p}\sum_{h=1}^{p} Y_{ig} V_{gk}^{-1} Y_{ik} - \left(\frac{\tau}{n\tau+1}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{g=1}^{p}\sum_{h=1}^{p} Y_{ig} V_{gk}^{-1} Y_{jk} && (6.39)\\
&= \sum_{i=1}^{n}(X_i - m)^T V^{-1}(X_i - m) - \left(\frac{\tau}{n\tau+1}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}(X_i - m)^T V^{-1}(X_j - m) && (6.40)\\
&= \sum_{i=1}^{n} X_i^T V^{-1} X_i - \sum_{i=1}^{n} X_i^T V^{-1} m - \sum_{i=1}^{n} m^T V^{-1} X_i + n m^T V^{-1} \\
&\quad -\frac{\tau}{n\tau+1}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} X_i^T V^{-1} X_j - n\sum_{i=1}^{n} X_i^T V^{-1} m\right. && (6.41)\\
&\qquad\qquad \left. -n\sum_{i=1}^{n} m^T V^{-1} X_i + n^2 m^T V^{-1} m\right)
\end{aligned}$$

By collecting terms, we see

$$
\begin{aligned}
&\mathrm{tr}(R) \\
&= \left(1 - \frac{\tau}{n\tau+1}\right)\sum_{i=1}^{n}(X_i^T V^{-1} X_i) - \frac{1}{n\tau+1}\sum_{i=1}^{n} X_i^T V^{-1} m \\
&\quad -\frac{1}{n\tau+1} m^T V^{-1}\sum_{i=1}^{n} X_i - \frac{\tau}{n\tau+1}\sum_{i=1}^{n}\sum_{j\neq i} X_i^T V^{-1} X_j + \frac{n}{n\tau+1} m^T V^{-1} m. \quad (6.42)
\end{aligned}
$$

Hence, $\exp\left\{-\frac{1}{2}\mathrm{tr}(V^{-1}(X-M)^T Q^{-1}(X-M))\right\}$ is equivalent to Equation 6.33. The normalizing constant for this expression is $[(2\pi V)^n (2\pi Q)^p]^{-1/2}$. The distribution of X is called a *Matrix Normal density*. Notation and theory for this distribution is presented by Dawid (1981), who would denote the distribution of $(X-M)$ by $\mathcal{N}(Q,V)$. Here, we will use the notation $MN(M,Q,V)$ since we want to consider location as well as scale. There are several ways to think about this distribution. Dawid (1981) define $(X-M)$ as the random matrix AZB, where $AA^T = Q$, $B^T B = V$, and the entries of Z are independent standard Normal random variables. Alternatively, we may write $X = AY + M$, where the rows of Y are independent $N(0,V)$. Note that the rows of Y have the same distribution as the observations when μ is known. In this sense, we may understand the effect of marginalizing out μ to be skewing the observations by $A = (I_n - \tau/(n\tau+1))^{-1/2}$, which introduces covariance between the observations. The covariance between $X_{i,j}$ and X_{i^*,j^*} is

$$
\mathrm{Cov}(X_{i,j}, X_{i^*,j^*}) = Q_{i,i^*} V_{j,j^*} = \begin{cases} (1+\tau)V_{j,j^*}, & i = i^* \\ \tau V_{j,j^*}, & i \neq i^* \end{cases}. \quad (6.43)
$$

For $i \neq i^*$, an interesting feature of Equation 6.43 is that when τ increases the covariance of $X_{i,j}$ and X_{i^*,j^*} increases relative to $\mathrm{Cov}(X_{i,j}, X_{i,j^*})$. Intuitively, when there is more

uncertainty (relative to the within group variance) about the group mean, μ, the observations provide more information about the center of the group, and hence also the other observations. In fact, as $\tau \to \infty$, the prior for μ becomes an improper uniform prior over the entire real line. Therefore, one observation provides good insight about the value of the remaining observations. In the other extreme, as $\tau \to 0$, the group mean is identically m, and the observations remain independent.

We are now ready to consider the conditional distribution of V given the observations, and the marginal distribution of $(X_1, \ldots, X_n)$.

$$
\begin{aligned}
f(X_1, \ldots, X_n, V) &= \frac{\partial}{\partial X}\mathrm{MN}(X; M, V, Q) \cdot \frac{\partial}{\partial V}\mathrm{IW}(V; d, D) && (6.44)\\
&= \frac{\exp\left\{-\frac{1}{2}\mathrm{tr}((X-M)^T Q (X-M) V^{-1})\right\}}{(2\pi)^{np/2}|V|^{n/2}|Q|^{p/2}} \\
&\quad \cdot \frac{|D|^{(d+p-1)/2}|V|^{-(d/2+p)}\exp\left\{-\frac{1}{2}\mathrm{tr}(DV^{-1})\right\}}{2^{(d+p-1)p/2}\Gamma_p((d+p-1)/2)}. && (6.45)\\
& && (6.46)
\end{aligned}
$$

Once again writing $S = (X-M)^T Q^{-1}(X-M)$, we see

$$
\begin{aligned}
f(X_1, \ldots, X_n, V) &= \frac{|D+S|^{(d+n+p-1)/2} V^{-((d+n)/2+p)} \exp\left\{-\frac{1}{2}\mathrm{tr}([D+S]V^{-1})\right\}}{2^{(d+n+p-1)p/2}\Gamma_p((d+n+p-1)/2)} \\
&\quad \cdot \pi^{-np/2}|Q|^{-p/2}\frac{\Gamma_p((d+n+p-1)/2)}{\Gamma_p((d+p-1)/2)}\frac{|D|^{(d+p-1)/2}}{|D+S|^{(d+n+p-1)/2}} \qquad (6.47)
\end{aligned}
$$

From the first factor in Equation 6.47, we see

$$
V|(X_1, \ldots, X_n) \sim \mathrm{IW}(d+n, D + (X-M)^T Q^{-1}(X-M)) \qquad (6.48)
$$

The second factor in Equation 6.47 is the marginal distribution of the data.

$$f(X_1, \ldots, X_n) = \pi^{-np/2} \frac{\Gamma_p((d+n+p-1)/2)}{\Gamma_p((d+p-1)/2)} |Q|^{-p/2} |D|^{(d+p-1)/2} |D+S|^{-(d+n+p-1)/2}. \tag{6.49}$$

The distribution in Equation 6.49 of X is called a *(Non-Central Scaled) Matrix T* density. This is a generalization of the Matrix T distribution presented by Dickey (1967), which itself is a generalization of the multivariate t-distribution. Dawid (1981) describes this as the marginal distribution of $A(X-M)$, where $\Sigma \sim \text{IW}(d, I_p)$, $(X-M)|\Sigma \sim \text{MN}(0, I_n, \Sigma)$, and $A^T A = Q$. He denotes the distribution of $(X-M)$ by $T(d; Q, D)$. Here, we will use the notation $\text{T}(d, M, Q, D)$ since we want to consider location as well as scale.

From the above analysis, it is easy to see that the posterior distribution of (μ, V) is also a N $\times$ IW law. Thus, this family is a conjugate prior for sampling from the Normal distribution. In particular, we note that the covariance matrix of μ is $\tau/(n\tau+1) \cdot V$. Since this is a scalar multiple of V, we are assured that the posterior distribution is also strong hyper Markov.

In the context of the Gibbs sampler we've discussed throughout this work, we will often work with the case $n = 1$. During each loop and for every observation X_i, we shall decide if X_i should be a member of an existing cluster or if we should create a new cluster for it. We choose the latter with probability proportional to $f(X_i)$, in which case we generate new parameter values with density $f(V|X_i)f(\mu|V, X_i)$. Therefore, it is beneficial to consider this special case. Note that $Q = 1 + \tau$ is a scalar when $n = 1$.

$$\begin{aligned}
\mu|(X_1, V) &\sim \mathrm{N}\left(\frac{\tau X_1 + m}{\tau + 1}, \frac{\tau}{\tau + 1} V\right) && (6.50)\\
X_1|V &\sim \mathrm{N}(m, (\tau + 1)V) && (6.51)\\
V|X_1 &\sim \mathrm{IW}\left(d + 1, D + \frac{1}{\tau + 1}(X_1 - m)(X_1 - m)^T\right) && (6.52)\\
X_1 &\sim \mathrm{T}(d, m, 1 + \tau, V) && (6.53)
\end{aligned}$$

The Matrix T density for X_1 reduces rather nicely to a multivariate T distribution. First consider the ratio

$$|D|^{(d+p-1)/2}|D + S|^{-(d+p)/2} = |D|^{-1/2}|I_p + D^{-1}S|^{-(d+p)/2}. \tag{6.54}$$

Noting that $S = (1+\tau)^{-1}(X - M)^T(X - M) = (1+\tau)^{-1}(X_1 - m)(X_1 - m)^T$ and applying Sylvester's Determinant Theorem to the second determinant, we see

$$|D|^{(d+p-1)/2}|D + S|^{-(d+p)/2} = |D|^{-1/2}(1 + (X_1 - m)^T D^{-1}(X_1 - m))^{-(2+p)/2} \tag{6.55}$$

We can also reduce the ratio of multivariate gamma functions to a ratio of two univariate gamma functions. Recall that the multivariate gamma can be specified recursively as $\Gamma_p(x/2) = \pi^{p(p-1)/4} \prod_{j=1}^{p}((x + 1 - j)/2)$. Therefore,

$$\frac{\Gamma_p\left(\frac{d+p}{2}\right)}{\Gamma_p\left(\frac{d+p-1}{2}\right)} = \frac{\pi^{p(p-1)/4}\Gamma\left(\frac{d+p}{2}\right)\Gamma\left(\frac{d+p-1}{2}\right)\cdots\Gamma\left(\frac{d+1}{2}\right)}{\pi^{p(p-1)/4}\Gamma\left(\frac{d+p-1}{2}\right)\cdots\Gamma\left(\frac{d+1}{2}\right)\Gamma\left(\frac{d}{2}\right)}, \tag{6.56}$$

which neatly telescopes to

$$\frac{\Gamma_p\left(\frac{d+p}{2}\right)}{\Gamma_p\left(\frac{d+p-1}{2}\right)} = \frac{\Gamma\left(\frac{d+p}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}. \tag{6.57}$$

Finally, combining Equations 6.55 and 6.57 with the remaining constants in Equation 6.49, we see that the marginal distribution of $X_{\mathbf{A}}$ is

$$f(X_1) = \frac{\pi^{-p/2}\Gamma\left(\frac{d+p}{2}\right)}{\Gamma\left(\frac{d}{2}\right)(1+\tau)^{p/2}|D|^{1/2}} \cdot \left(1 + (X_1 - m)^T D^{-1}(X_1 - m)\right)^{-(d+p)/2} \tag{6.58}$$

This distribution is a non-central, scaled version of the multivariate t-distribution, with center m and scale $(1+\tau)D$.

We are now ready to specify the distributions for our graphical model, the $\mathrm{HN}_{\mathcal{G}} \times \mathrm{HIW}_{\mathcal{G}}$ law. In Theorem 6.1.1 we showed that the $\mathrm{HN}_{\mathcal{G}} \times \mathrm{HIW}_{\mathcal{G}}$ law is strong hyper Markov. Therefore, Corollary 2.3.9 allows us to compute the posterior update for each clique locally. That is, the posterior distribution of $(\mu_{\mathbf{C}}, V_{\mathbf{C}})$ for any clique $\mathbf{C}$ depends only on $X_{1\mathbf{C}}, \ldots, X_{n\mathbf{C}}$. Since each clique is complete, we can apply the above analysis. For example, the posterior distribution of $\mu_{\mathbf{C}}|V_{\mathbf{C}}$ is a Normal distribution. Therefore, the posterior for μ given V is *hyper* Normal by definition. Similarly, the posterior distribution of V is hyper Inverse-Wishart. We see that the posterior distribution for (μ, V) is a $\mathrm{HN}_{\mathcal{G}} \times \mathrm{HIW}_{\mathcal{G}}$ law, which is to say that this family of laws is conjugate to sampling from the Normal distribution. Last but not least, since the prior law for (μ, V) is strong hyper Markov, we know that the marginal distribution of the observations is Markov by Corollary 2.3.10. We have shown that the marginal distribution of X is matrix T for each clique, whence the distribution of X is *hyper* matrix T.

6.1.3 Random Variable Generation

A major advantage to computing the density function of a (hyper) Markov distribution is that computations can be computed in blocks for each clique. In particular, each of the density calculations for Normal, Inverse Wishart, and T densities require inverting a $p \times p$ matrix. Despite the fact that positive definite matrices can be inverted via a relatively efficient Cholesky factorization, they still require on the order of p^3 operations. Thus, it is often faster to invert several smaller matrices than one large matrix, especially if p is large (where efficiency matters most) and the graph $\mathcal{G}$ is sparse. The (hyper) Markov versions of these distributions are given by the next three equations, where $\mathcal{G}$ is a decomposable graph with clique set $\mathscr{C}$ and separator set $\mathscr{S}$.

$$\frac{\partial}{\partial X}\mathrm{HN}_{\mathcal{G}}(X;\mu,V,\mathcal{G}) = \prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial X_{\mathbf{A}}}\mathrm{N}(X_{\mathbf{A}};\mu_{\mathbf{A}},V_{\mathbf{A}}) \tag{6.59}$$

$$\frac{\partial}{\partial V}\mathrm{HIW}_{\mathcal{G}}(V;d,D,\mathcal{G}) = \prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial V_{\mathbf{A}}}\mathrm{dIW}(V_{\mathbf{A}};d,D_{\mathbf{A}}) \tag{6.60}$$

$$\frac{\partial}{\partial X}\mathrm{HT}_{\mathcal{G}}(X;M,Q,D) = \prod_{\mathbf{A}\in\mathscr{C}\setminus\mathscr{S}} \frac{\partial}{\partial X_{\mathbf{A}}}\mathrm{dT}(X_{\mathbf{A}};M_{\mathbf{A}},Q_{\mathbf{A}},V_{\mathbf{A}}) \tag{6.61}$$

The last major computational hurdle of our Gibbs sampler is that we need to simulate μ from a hyper Normal distribution and V from a hyper Inverse Wishart distribution. Algorithms for generating Normal random variables are well-known, as are the conditional distributions of Normal vectors given some of its components. Therefore, hyper Normal random variables are relatively straightforward to generate. Let $\mathbf{C}_1,\ldots,\mathbf{C}_k$ be a perfect sequence of cliques with separators $\mathbf{S}_2,\ldots,\mathbf{S}_k$. We simulate a hyper Normal random vari-

able with distribution $\text{HN}_{\mathcal{G}}(m, V)$, by first generating the marginal value $\mu_{\mathbf{C}_1}$. Then for $i = 2, \ldots, k$, we generate the value of $\mu_{\mathbf{C}_i}$ given $\mu_{\mathbf{S}_i}$, which is a Normal random variable with mean $m_{\mathbf{C}_i} + V_{\mathbf{C}_i\mathbf{S}_i}V_{\mathbf{S}_i}^{-1}(\mu_{\mathbf{S}_i} - m_{\mathbf{S}_i})$ and covariance matrix $V_{\mathbf{C}_i} - V_{\mathbf{C}_i\mathbf{S}_i}V_{\mathbf{S}_i}^{-1}V_{\mathbf{S}_i\mathbf{C}_i}$.

Before specifying the simulation of $\text{HIW}_{\mathcal{G}}$ random variables, we first review a generative method for the Inverse-Wishart distribution. Let D be a $p \times p$ positive definite matrix $K^TK = D^{-1}$ and $d > 0$. It is well-known (Kshirsagar, 1959), that if L is a random $p \times p$ lower triangular matrix with independent entries such that $L_{i,i}^2 \sim \chi_{d+p-i}^2$, and for $j < i, L_{i,j} \sim N(0, 1)$, then $W = (KL)(KL)^T \sim W(d+p-1, D^{-1})$. Therefore, $V = W^{-1} \sim \text{IW}(d, D)$. Functionally, in order to invert W we first calculate its Cholesky square root, KL, which is triangular and easy to invert. Hence, we stop shy of computing W and work directly with KL.

Now that we understand how to simulate Inverse Wishart variables, we are ready to discuss generating $V \sim \text{HIW}_{\mathcal{G}}(d, D)$. We employ the method of Carvalho et al. (2007). Once again, let $\mathbf{C}_1, \ldots, \mathbf{C}_k$ be a perfect sequence of cliques with separators $\mathbf{S}_2, \ldots, \mathbf{S}_k$ and residuals $\mathbf{R}_2, \ldots, \mathbf{R}_k$. Let $\mathbf{A}_i = \cup_{j=1}^{i-1}\mathbf{C}_j \setminus \mathbf{S}_i$. We also define $V_{\mathbf{R}_i|\mathbf{S}_i} = V_{\mathbf{R}_i} - V_{\mathbf{R}_i\mathbf{S}_i}V_{\mathbf{S}_i}^{-1}V_{\mathbf{S}_i\mathbf{R}_i}$ and $D_{\mathbf{R}_i|\mathbf{S}_i} = D_{\mathbf{R}_i} - D_{\mathbf{R}_i\mathbf{S}_i}D_{\mathbf{S}_i}^{-1}D_{\mathbf{S}_i\mathbf{R}_i}$. We start by generating $V_{\mathbf{C}_1} \sim \text{IW}(d, D_{\mathbf{C}_1})$. Then for $i = 2, \ldots, k$ we generate $V_{\mathbf{C}_i}$ given $V_{\mathbf{S}_i}$ using the following algorithm.

1. Generate $V_{\mathbf{R}_i|\mathbf{S}_i} \sim \text{IW}(d + |\mathbf{R}_i|, D_{\mathbf{R}_i} - D_{\mathbf{R}_i\mathbf{S}_i}D_{\mathbf{S}_i}^{-1}D_{\mathbf{S}_i\mathbf{R}_i})$.

2. Generate $(V_{\mathbf{R}_i\mathbf{S}_i}V_{\mathbf{S}_i}^{-1}) \sim \text{MN}(D_{\mathbf{R}_i\mathbf{S}_i}D_{\mathbf{S}_i}^{-1}, V_{\mathbf{R}_i.\mathbf{S}_i})$

3. Set $V_{\mathbf{R}_i} = V_{\mathbf{R}_i|\mathbf{S}_i} + V_{\mathbf{R}_i\mathbf{S}_i}V_{\mathbf{S}_i\mathbf{R}_i}$.

4. (optional) Set $V_{\mathbf{R}_i\mathbf{A}_i} = V_{\mathbf{R}_i\mathbf{S}_i} V_{\mathbf{S}_i}^{-1} V_{\mathbf{S}_i\mathbf{A}_i}$.

Because we shall work with hyper Markov distributions, it is enough to generate $V_{\mathbf{C}_i}$ for $i = 1, \ldots, k$. Therefore, the fourth step in the above algorithm is not necessary unless we desire to fully specify the covariance matrix.

For the current work, we utilized the GNU Scientific Library (GSL) to generate chi-squared and standard Normal random variables. We also used the GSL for Cholesky factorizations. We invoked the C Basic Linear Algebra Subprograms (CBLAS) library for many other matrix operations. A few matrix operations were coded anew when suitably efficient routines were not available in CBLAS. For example, we were surprised to see that CBLAS had no routine for multiplying two triangular matrices! Thus, we needed our own routines to take advantage of the efficiency of this special class of matrices. Notably, triangular matrices can be multiplied in place which reduces memory overhead. In fact, we can even square a triangular matrix, $A^T A$ or AA^T, in place.

6.1.4 Gibbs Sampling Algorithm

Having found the necessary conditional distributions and methods for generating the requisite random variables, we are now ready to specify the Gibbs sampler. In the algorithm below, we denote the number of unique parameter values by k, and denote the j^{th} unique value by (μ_j^*, V_j^*). We denote the parameters for the i^{th} observation by (μ_i, V_i) and set $j^* = \{i : (\mu_i, V_i) = (\mu_j^*, V_j^*)\}$. The number of observations whose parameters are (μ_j^*, V_j^*) is n_j, while $n_j^{(-i)}$ is the number of such observations, excluding X_i. We denote by X_{j^*} the

$n_j \times p$ matrix whose rows are the observations, X_i such that $i \in j^*$. The mean matrix M_{j^*} is the $n_j \times p$ matrix whose rows are identically m and Q_{n_j} is the $n_j \times n_j$ matrix equal to $I_{n_j} + \tau J_{n_j n_j}$. Finally, n represents the total number of observations, and B is the total number of iterations for the Gibbs sampling loop, including burn-in.

1. For $i = 1, \ldots, N$: Set $w_i^0 = \frac{\partial}{\partial X_i} \mathrm{HT}_{\mathcal{G}}(X_i; d, (1+\tau), D)$.

2. For $i = 1, \ldots, N$

   (a) Generate $V_i \sim \mathrm{HIW}_{\mathcal{G}}(d+1, D + X_i X_i^T/(\tau+1))$.

   (b) Generate $\mu_i \sim \sum \mathrm{HN}_{\mathcal{G}}((\tau X_i + m)/(\tau+1), \tau V_i/(\tau+1))$.

3. For $b = 1, \ldots, B$:

   (a) For $i = 1, \ldots, n$:

      i. For $j = 1, \ldots, k$: Set $w^0 = w_i^0$ and $w^i = n_j^{(-i)} \cdot \frac{\partial}{\partial X_i} H_{\mathcal{G}}(X_i; \mu_j^*, V_j^*)$.

      ii. Generate $U \sim \mathrm{Unif}(0, \sum w^j)$

      iii. Set $h = \min\left\{h : U \leq \sum_{j=1}^{h} w_j\right\}$.

      iv. If $h = 0$, generate $V_i \sim \mathrm{HIW}_{\mathcal{G}}(d+1, D + X_i X_i^T/(\tau+1))$
      and $\mu_i \sim \sum \mathrm{HN}_{\mathcal{G}}((\tau X_i + m)/(\tau+1), \tau V_i/(\tau+1))$.

      v. If $h > 0$, set $V_i = V_h^*$ and $\mu_i = \mu_h^*$.

   (b) For $j = 1, \ldots, k$

      i. Generate $V_j^* \sim \mathrm{HIW}_{\mathcal{G}}(d + n_j, D + (X_{j^*} - M_{j^*})^T Q_{n_j^*}^{-1} (X_j^* - M_{j^*})$

      ii. Generate $\mu_j^* \sim \mathrm{HN}_{\mathcal{G}}(\tau \sum_{i \in j^*} X_i + m/(n_j \tau + 1), \tau V_j^*/(n_j \tau + 1))$.

6.2 Simulation Study 1: Bivariate Gaussian Data (2 Groups)

A mixture of two bivariate Gaussian data comprised the first set of simulation studies. Data were generated from two groups with means $(-5, 0)$ and $(5, 0)$ The covariance matrix for both groups was I_2, which satisfies the independence graph. There were $n_1 = 50$ observations from the first group, and $n_2 = 30$ from the second. Note that since both groups were centered on the x-axis and had the same covariance matrix, data set satisfied the independence model, $\mathbf{E} = \emptyset$, even when collapsed across groups.

We chose to center and scale the data so that the overall data have mean 0 and variance I_2, though the individual components do not. This was a logical first step to simplify analyses, with no knowledge of the underlying mixture. We repeated the Gibbs sampling process several times with varying parameter values to assess how these parameters affect inference. Table 6.1 shows the settings for each run, as well as the running time. This does not encompass all trials, merely the ones we will discuss below. For example, we ran several Gibbs samplers using a burn-in of only 1000 iterations, but early diagnostics suggested longer burn-ins would be safer.

As a convergence criterion, we ran several parallel samplers and found where the mean number of components leveled off. From Figure 6.1, we see that the mean number of components approximately reaches its limit by 1000 iterations, though the particular limit varies with the model parameters. It seems that 1000 may be enough iterations, but we chose a burn-in of 5000 iterations to be safe. Contour plots of the estimated density confirm this choice (Figure 6.2). By 5000 iterations, the posterior density is very stable. Unfortunately,

Runs	ν	τ	d	m	D	n	g	Burn Iter	Samp. Iter.	Time (s)
001 - 005	1	1	2	0	I_2	80	2	5k	10k	558
006 - 008	.01	1	2	0	I_2	80	2	5k	10k	524
009 - 011	10	1	2	0	I_2	80	2	5k	10k	644
012 - 014	1	10	2	0	I_2	80	2	5k	10k	551
015 - 017	1	.1	2	0	I_2	80	2	5k	10k	753
018 - 020	1	1	10	0	I_2	80	2	5k	10k	570
021 - 023	1	1	50	0	I_2	80	2	5k	10k	577

Table 6.1: Partial list of parameter settings and runtimes for the Gibbs sampler in the first simulation study.

autocorrelation plots showed significant autocorrelation in the number of components for around lags 2 to 5, depending on the parameter values. To get an approximately independent sample, we only take every 10^{th} observation, although side-by-side contour plots reveal that there is little difference in the posterior density (Figure 6.3). Taking every 10^{th} iteration leaves only 1000 samples with which to estimate the posterior density. In order to see if this is a large enough sample, we compared posterior density estimates from five runs of the sampler with $(\nu, \tau, d) = (1, 1, 2)$. Three of the estimates are shown in Figure 6.8 (first column). The other two estimates match those that are pictured. Seeing that all the densities are identical, we can conclude that 1000 is an adequate sample size.

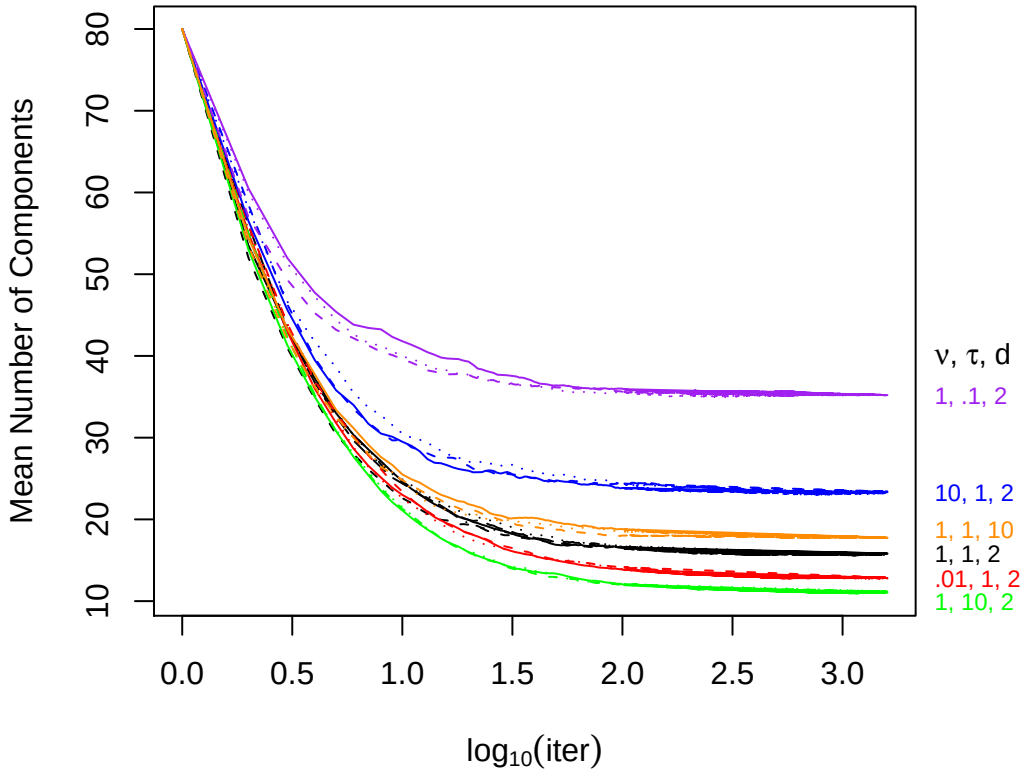

Figure 6.1: Convergence of mixture sizes for various parameter settings in the first simulation study. Not pictured: $(1, 1, 50)$ coincides with the group for $(1, 1, 10)$, indicating that both models behave similarly in this regard.

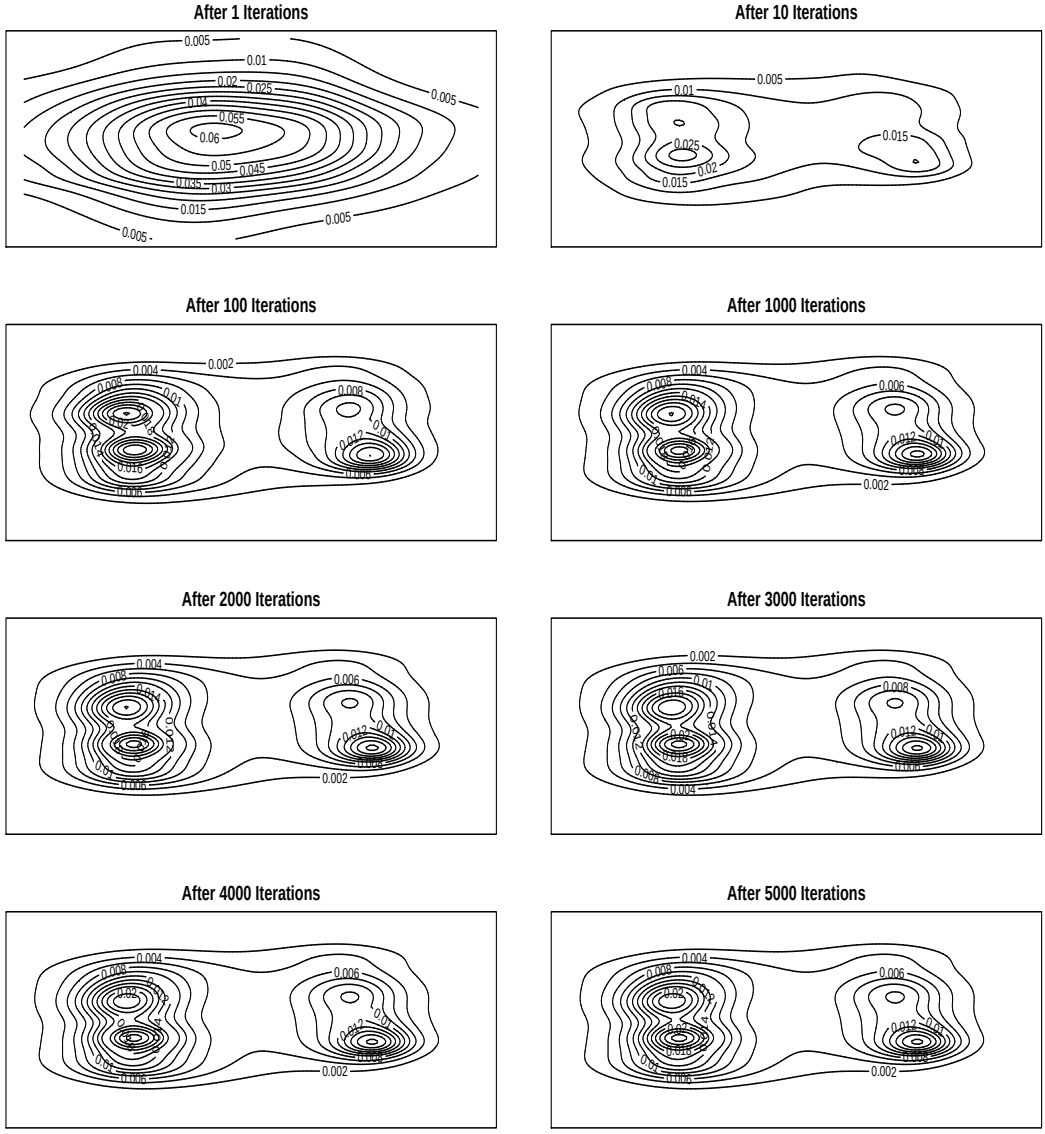

Figure 6.2: A sequence of estimates of the posterior distribution estimates in the first simulation study indicates that the Gibbs sampler has converged by around 1000 iterations.

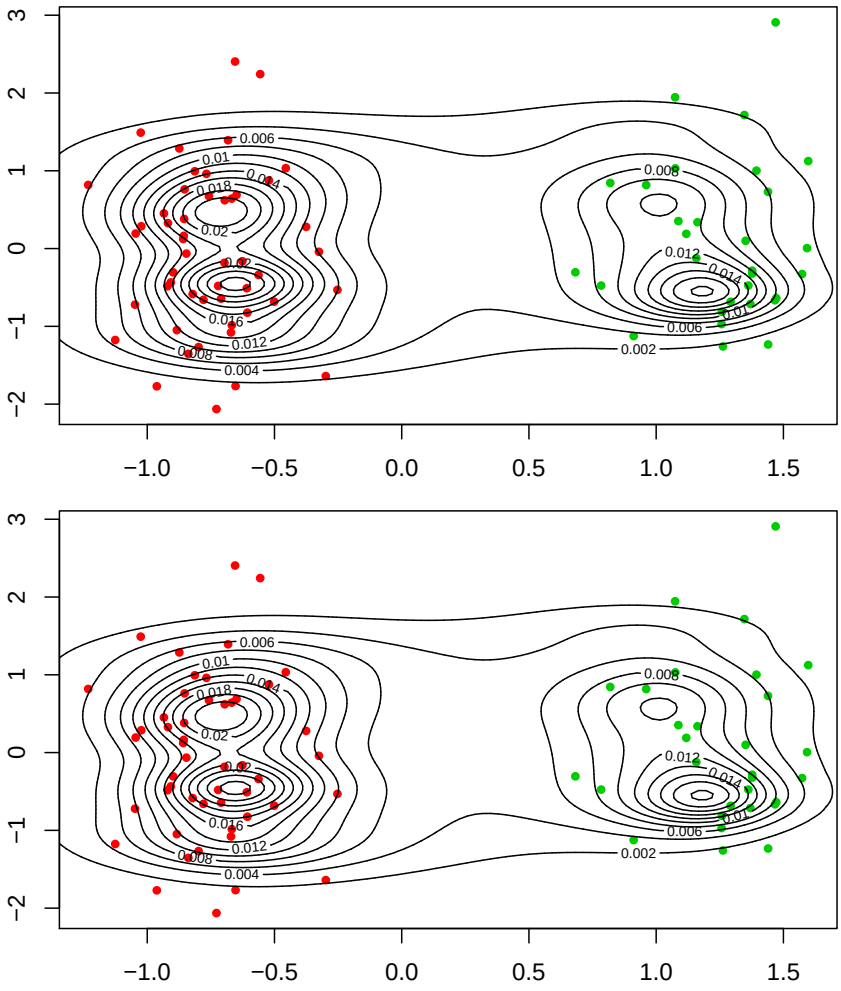

Figure 6.3: Posterior density estimates in the first simulation study for $(\nu, \tau, d) = (1, 1, 2)$ for 1000 approximately independent Gibbs samples taking every tenth iteration (top); and 10000 consecutive Gibbs samples with autocorrelation (bottom). Comparison reveals that both methods yield the same posterior.

6.2.1 Analysis of Run Time

A primary question of interest is how quickly we can generate samples from the posterior distribution. As we see in Table 6.1, the time required for 15,000 iterations was around 9-12 minutes, depending on the parameter values. Quite naturally, the main explanation for the difference in run time is the average number of components. When we sample the posterior distribution of $\theta_i|X_i$, we require a weight for each component in the mixture. After updating each θ_i, we then remix by updating the unique parameter values, one for each component. These two steps comprise the vast majority of the computational complexity within each iteration. Therefore, we expect an approximately linear relationship between the average number of components and the runtime. Figure 6.4 reveals that this is true in general. One significant deviation from the linear pattern are the points pertaining to $(\nu, \tau, d) = (1, 10, 2)$. The reason for this is unclear, but a good guess is that the workload for the computer was higher during those runs.

We can analyze the linear pattern further. The line in Figure 6.4 is the ordinary least squares regression line, which has an intercept of 415 seconds and a slope of 9.48 seconds per component. The intercept pertains roughly to the overhead involved in the Gibbs sampler, including reading in the data, graph, other parameters, and calculating the initial weights. The slope, of course, is the increase in runtime when the mixtures have one more component on average. By dividing by the total number of iterations, we approximate that the cost of one additional component is 0.6ms for a single iteration. A simple R^2 value for

the ordinary least squares regression shows that the linear relationship accounts for over 96% of the variation in runtime.

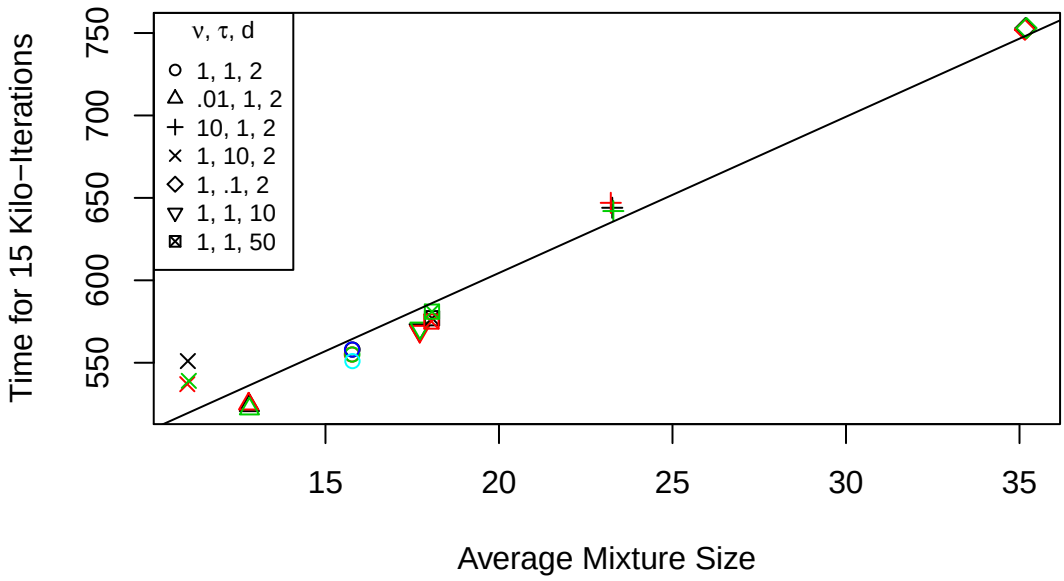

Figure 6.4: Runtime versus average number of components for various hyperparameters in the first simulation study.

6.2.2 Analysis of the Number of Mixture Components

We can see how the parameter values affect the number of clusters by comparing the posterior distribution for various settings of the hyperparameters. The analyses in this section are based on using every 10^{th} sample in order to minimize any correlation in the sample.

Figure 6.5 shows the distribution of the number of components at each iteration, with (τ, d) fixed at $(1, 2)$. As expected, the number of unique parameter values (and the spread) increases with ν. This follows the theory presented by Antoniak (1974) who showed that the number of components increases stochastically with ν. The prior expectation for the number of components is 1.05, 4.97, and 22.4 for $\nu = .01$, 1, and 10, respectively. The posterior means, based on the Gibbs samples are 12.8, 15.8, and 23.3. In other words, the posterior mean is weighted toward the prior mean for an ν around 10 to 11. We consider inference for ν more fully in Section 6.5.

The posterior distribution for the number of components at various levels of τ is shown in Figure 6.6. Clearly, the number of components increases with τ. The makes intuitive sense, because large values of τ imply that the marginal T distribution for X_i is flat with large variance. Hence, the probability of creating a new component, which is proportional to the marginal density evaluated at X_i, is small for large τ.

In contrast to ν and τ, the hyperparameter d does not seem to have much influence on the posterior number of components. There appears to be little difference between 2 and 10 degrees of freedom. There is even less difference when comparing 10 and 50 degrees of freedom. The difference in the posterior mean is around .4 components, and there appears to be a very slight decrease in the variance of the mixture sizes.

In general, we see that the number of mixture components is vastly overestimated. This is consistent with the results of Escobar and West (1995) in the univariate case.

6.2.3 Analysis of Posterior Densities

The hyperparameters affect the posterior distribution in logical ways. Contours of the posterior distribution are shown in Figure 6.8 for various levels of ν, τ, and d. We consider each of these parameters in turn. The plots in the first column are three independent runs with $\nu = \tau = 1$ and $d = 2$. We shall consider this a "baseline" and compare the other densities to them.

We can see the effect of changing ν in the first row of Figure 6.8. It is well-known that Dirichlet processes tend to result in a small number of sharp peaks when ν is small, and we see that in the first two plots. By contrast, in the third plot with $\nu = 10$, the posterior distribution is relatively flat.

The effect of different levels of τ is demonstrated in the second row. Interestingly, the second plot is the only plot in Figure 6.8 that is unimodal. This is explained by the very definition of τ, meaning that the distribution of group means (μ_i^*s) has very little variance. As we expected, this results in the location of each mixture component being near $m = 0$. As an added note, recall that $\tau = .1$ also led to a increased number of mixture components. This is because none of the observations are very close to the group means and therefore a new cluster is created for them. Therefore, the unimodal distribution for $\tau = .1$ is actually a large number of Gaussian components overlapping. For $\tau = 10$, the modes in the posterior are more peaked than the baseline $\tau = 1$. This is a result of the decreased number of components. As explained above, the large value of τ yields a rather uninformative prior for μ_i^* and hence small values for the marginal $f(X_i)$. This results in

a stronger chance of combining observations into a single components, which in turn gives a more precise measurement of the covariance for that component.

The effect of d, shown in the first row of Figure 6.8, is straightforward. Larger degrees of freedom for the inverse Wishart prior results in smaller determinants $|V_i|$. This means that the individual Gaussian components have less variance, and therefore there is a decreased chance that any two given components will overlap. The end result is a relatively large number of sharp peaks compared to when $d = 2$.

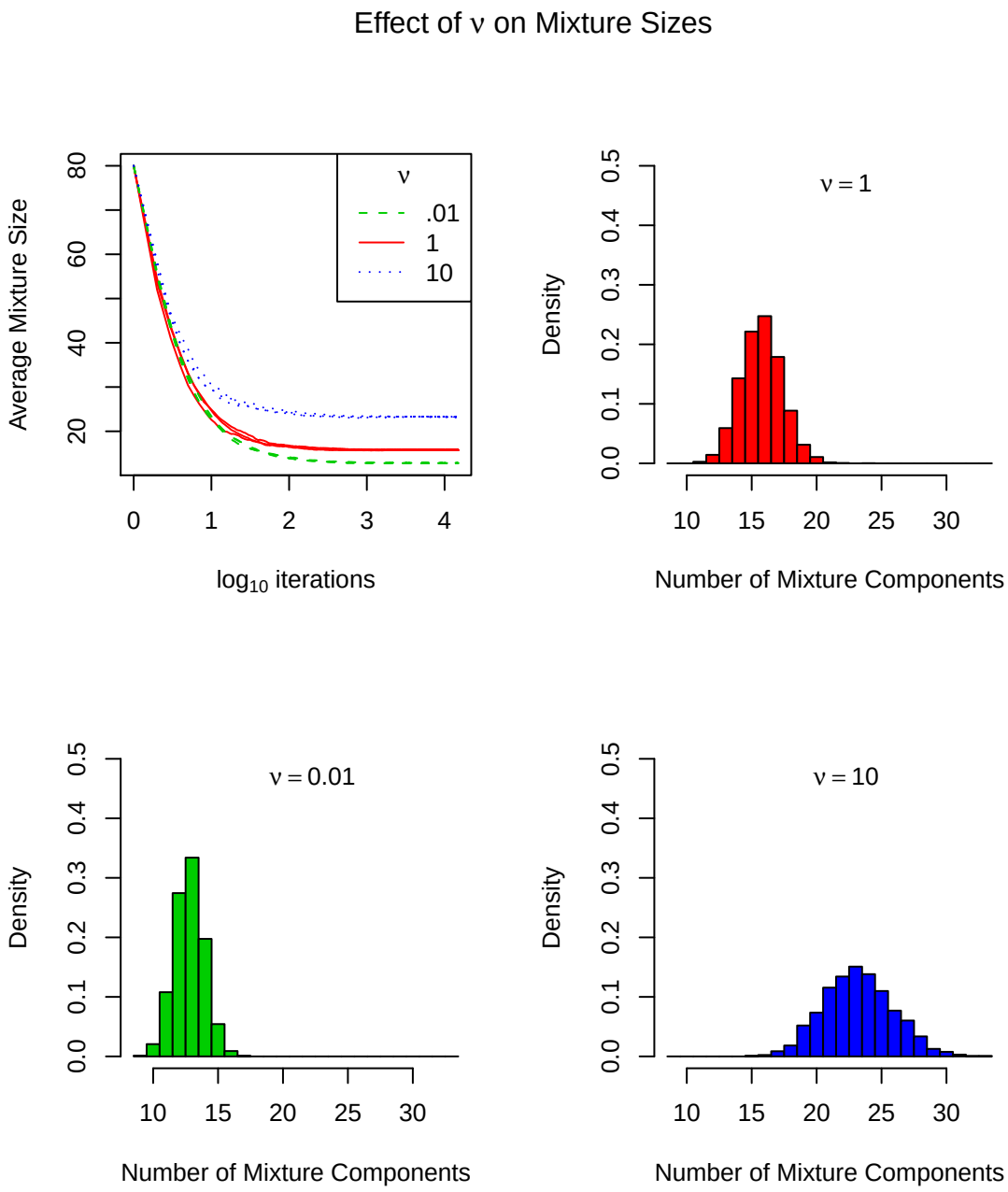

Figure 6.5: Posterior distribution of the number of components with $(\nu, \tau, d) = (\nu, 1, 2)$ in the first simulation study.

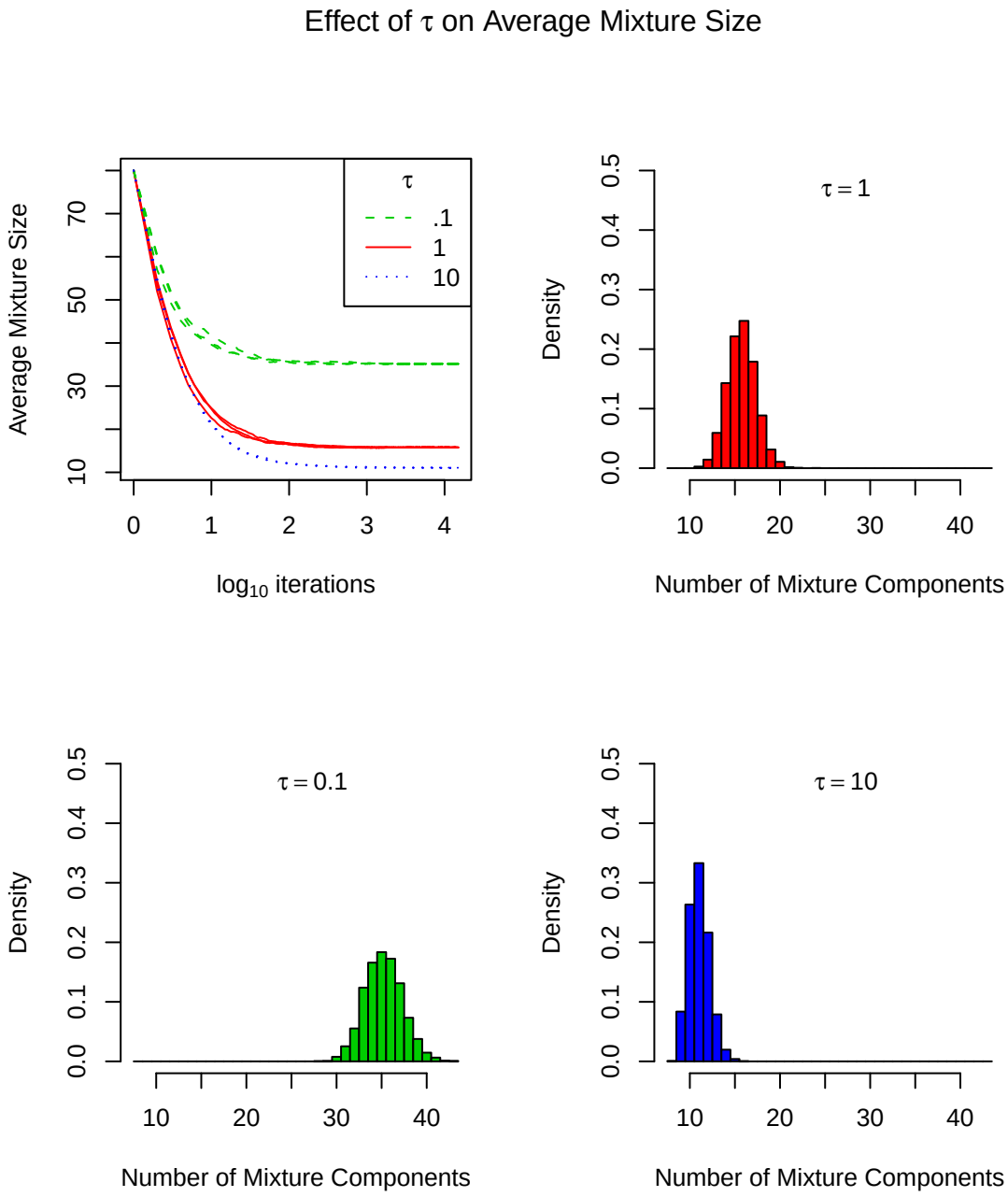

Figure 6.6: Posterior distribution of the number of components with $(\nu, \tau, d) = (1, \tau, 2)$ in the first simulation study.

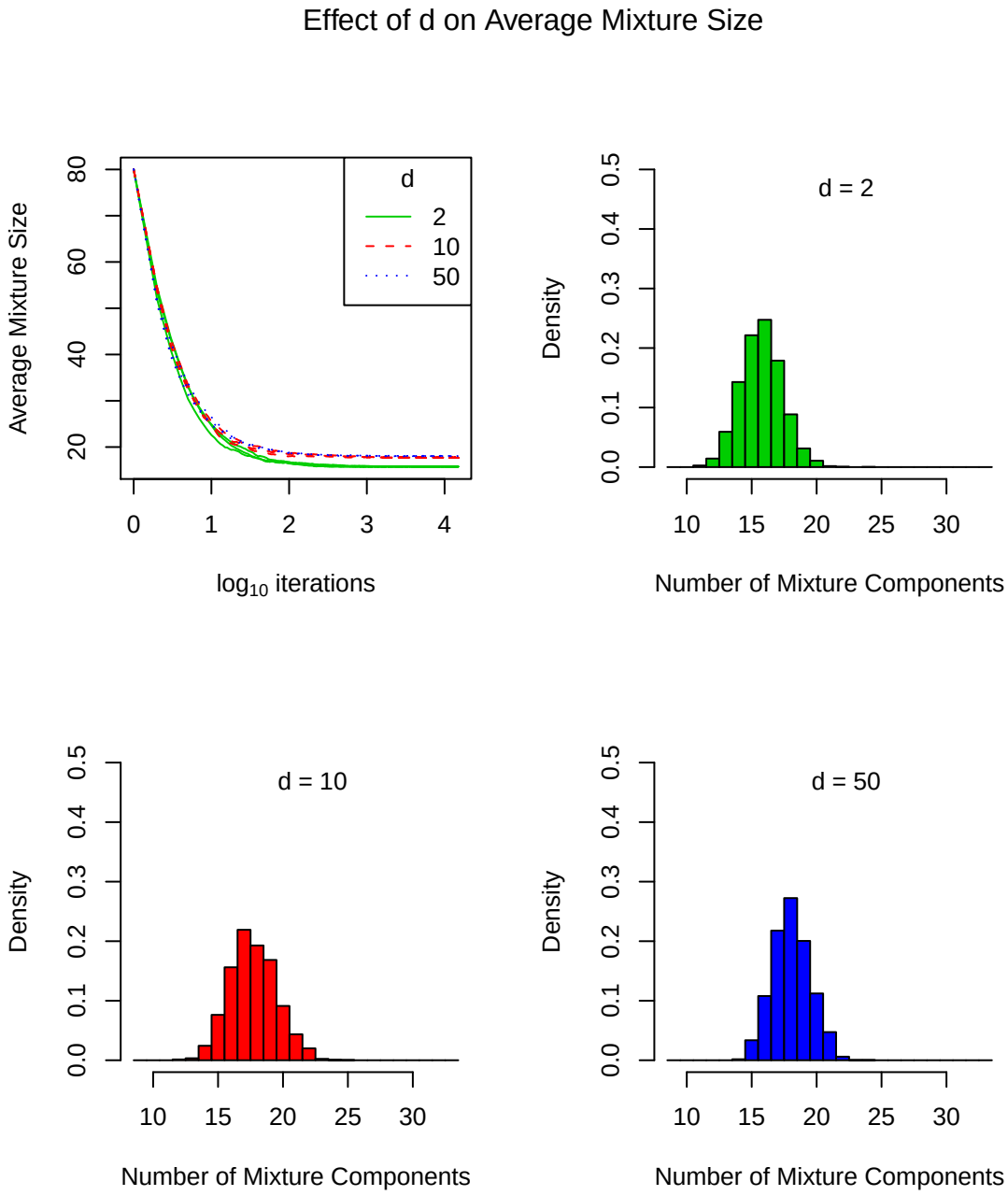

Figure 6.7: Posterior distribution of the number of components in the first simulation study with $(\nu, \tau, d) = (1, 1, d)$. The green lines pertaining to $d = 50$ are occluded by the blue lines for $d = 10$ because the posterior means are almost identical.

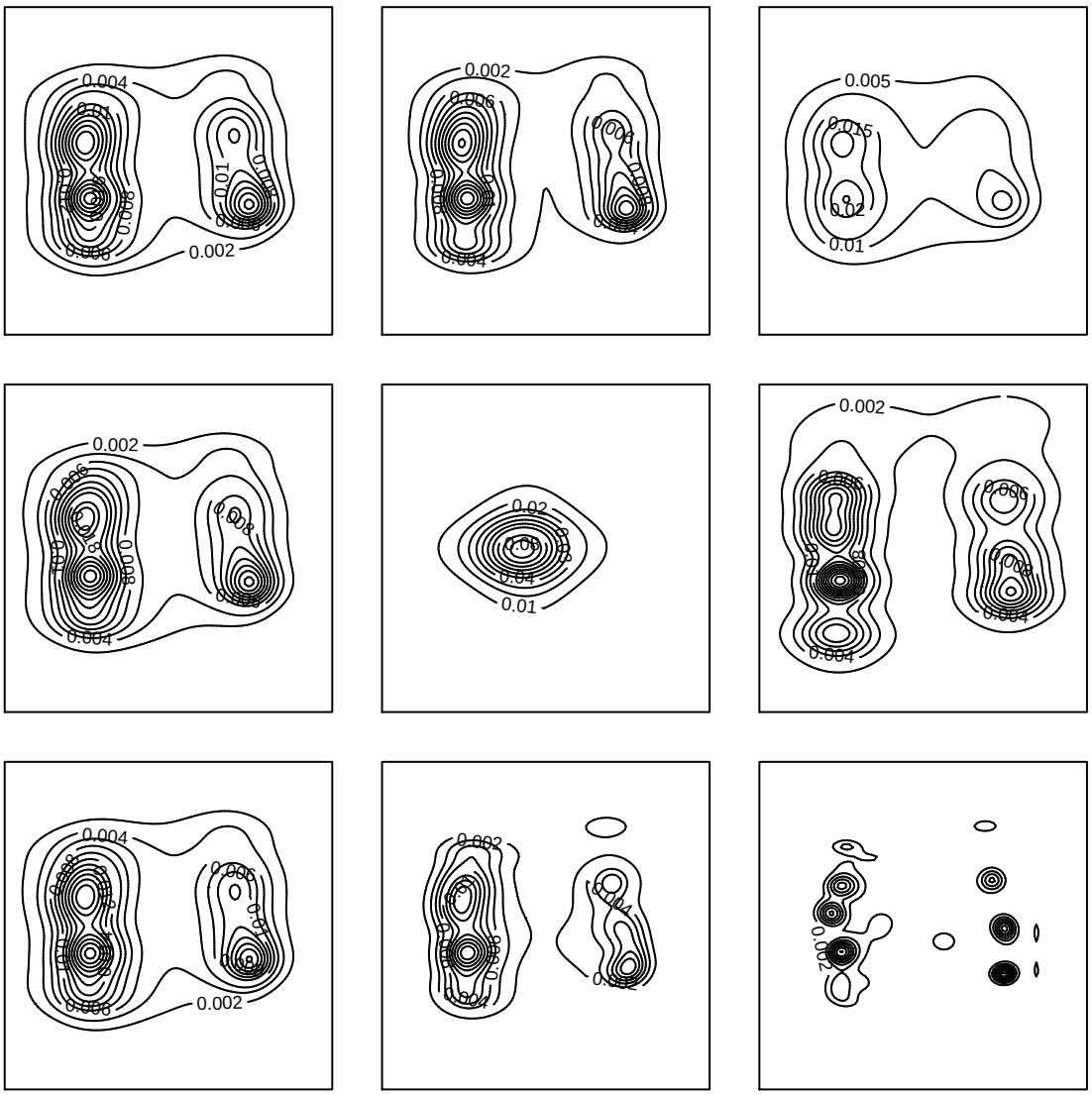

Figure 6.8: Comparison of posterior density estimates in the first simulation study. The top row compares models with ν varying from among $\{1, .01, 10\}$ (left to right) with $(\tau, d) = (1, 2)$. The middle row compares models with τ varying among $\{1, .1, 10\}$ with (ν, d) fixed at $(1, 2)$. The bottom row compares models with d varying among $\{2, 10, 50\}$ with $\nu = \tau = 1$.

6.3 Simulation Study 2: Bivariate Gaussian (3 Groups)

For further analysis, we combined data from the previous study with $n_3 = 40$ observations from a third Gaussian component with mean $\mu = (0, 5)$ and a diagonal variance matrix V with $V_{11} = .5$ and $V_{22} = 1$. Figure 6.9 reveals that a burn-in of 5000 iterations is still adequate. Once again, in order to minimize autocorrelation in the estimates, we used every 10^{th} iteration after the burn-in. This gives an effective sample size of 1000. As before, the posterior density estimate fits the data well, with some shrinkage toward the overall center (Figure 6.10).

In comparing the density of the number of components (Figures 6.11-6.13), we see that the components continue to have the same effect as in the smaller simulation. Increases in ν lead to a higher mean and larger spread in the number of mixture components, whereas increases in τ lead to fewer mixture components and less variance. The degrees of freedom, d, has little to no effect on the number of components.

Contour plots of the posterior density are shown in Figure 6.14. The first column shows three independent runs with the "baseline" parameters $(\nu, \tau, d) = (1, 1, 2)$. The similarity among these three plots shows that 1000 samples is sufficient for determining the posterior. Overall, the contours exhibit a relationship similar to this in Figure 6.8 from Simulation 1. Briefly, as ν increases, the posterior estimate flattens, whereas the posterior has relatively sharp peaks for ν equal to 1 or .01. Setting $\tau = .1$ once again constrains the means to be near zero, which leads to a large number of components but only one mode. Finally, as d

increases, the Gaussian components are more concentrated, leading to the "island" pattern in the bottom-right contour.

Run	ν	τ	d	m	D	n	g	Burn Iter	Samp. Iter.	Time (s)
101 - 105	1.00	1.0	2	0	I_3	110	3	5k	10k	1410
106 - 108	0.01	1.0	2	0	I_3	110	3	5k	10k	1365
109 - 111	10.00	1.0	2	0	I_3	110	3	5k	10k	1452
112 - 114	1.00	10.0	2	0	I_3	110	3	5k	10k	1393
115 - 117	1.00	0.1	2	0	I_3	110	3	5k	10k	1585
118 - 120	1.00	1.0	10	0	I_3	110	3	5k	10k	1425
121 - 123	1.00	1.0	50	0	I_3	110	3	5k	10k	1541

Table 6.2: Partial list of parameter settings and runtimes for the Gibbs sampler in the second simulation study.

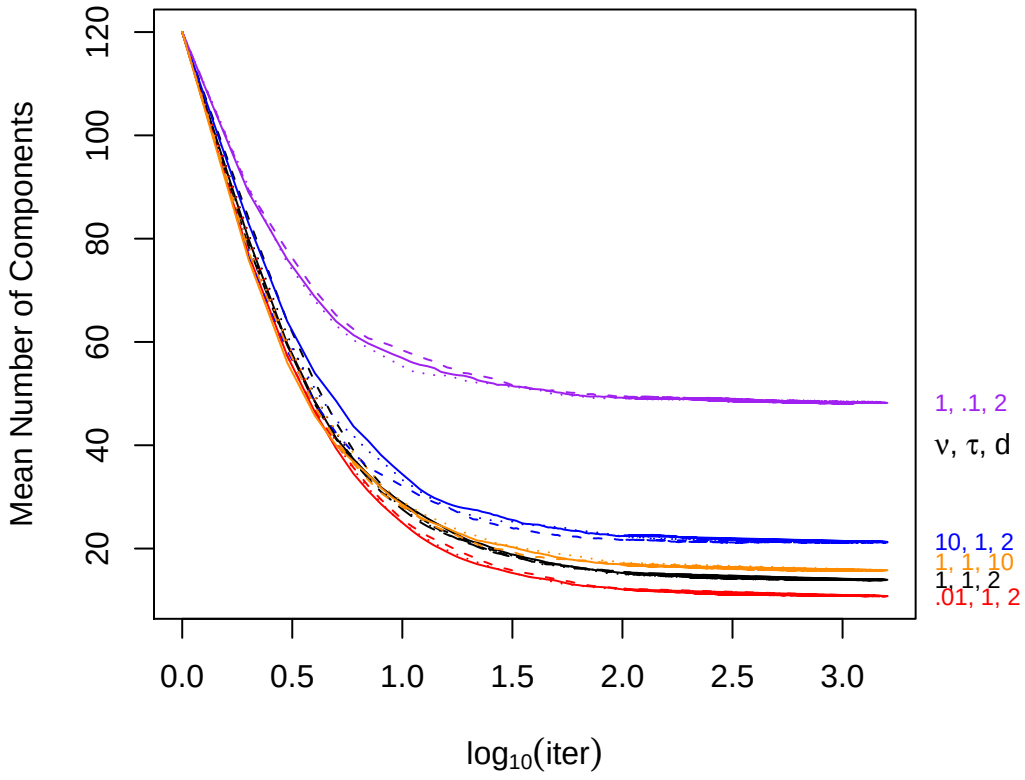

Figure 6.9: Convergence of the mixture sizes for data from a mixture of three Gaussians in the second simulation study. For clarity, two series have been removed that closely resemble one of the pictured series. $(1, 10, 2)$ overlaps $(.01, 1, 1)$, so the former is not shown. Likewise, $(1, 1, 50)$ is not shown because it coincides with $(1, 1, 10)$.

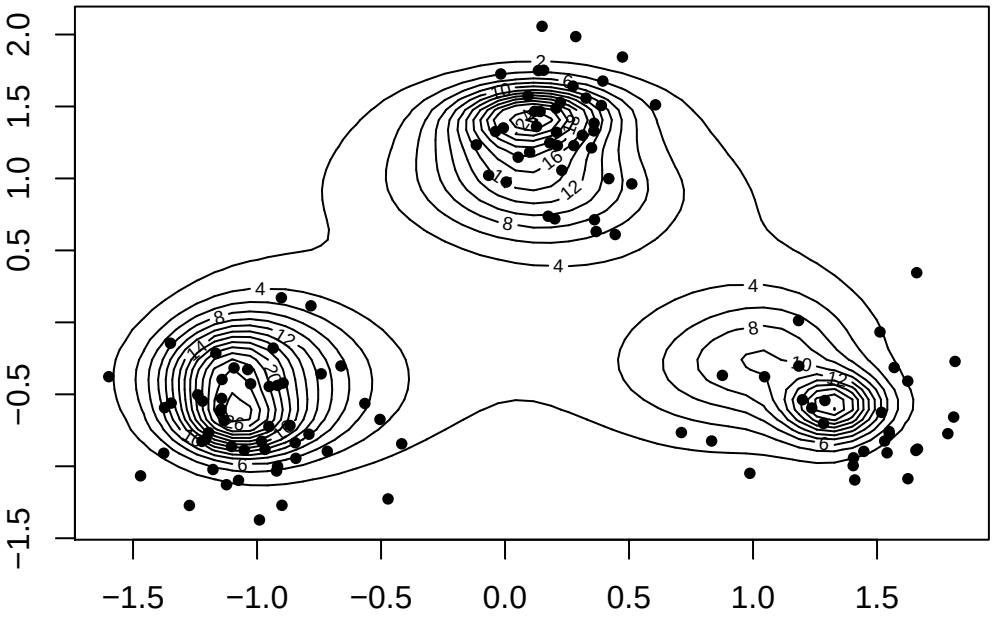

Figure 6.10: Contour plot of the posterior density estimation from a Gibbs sample with $(\nu, \tau, d) = (1, 1, 2)$ overlayed on the data for the second simulation study. As in the first simulation, component means exhibit shrinking toward the overall center.

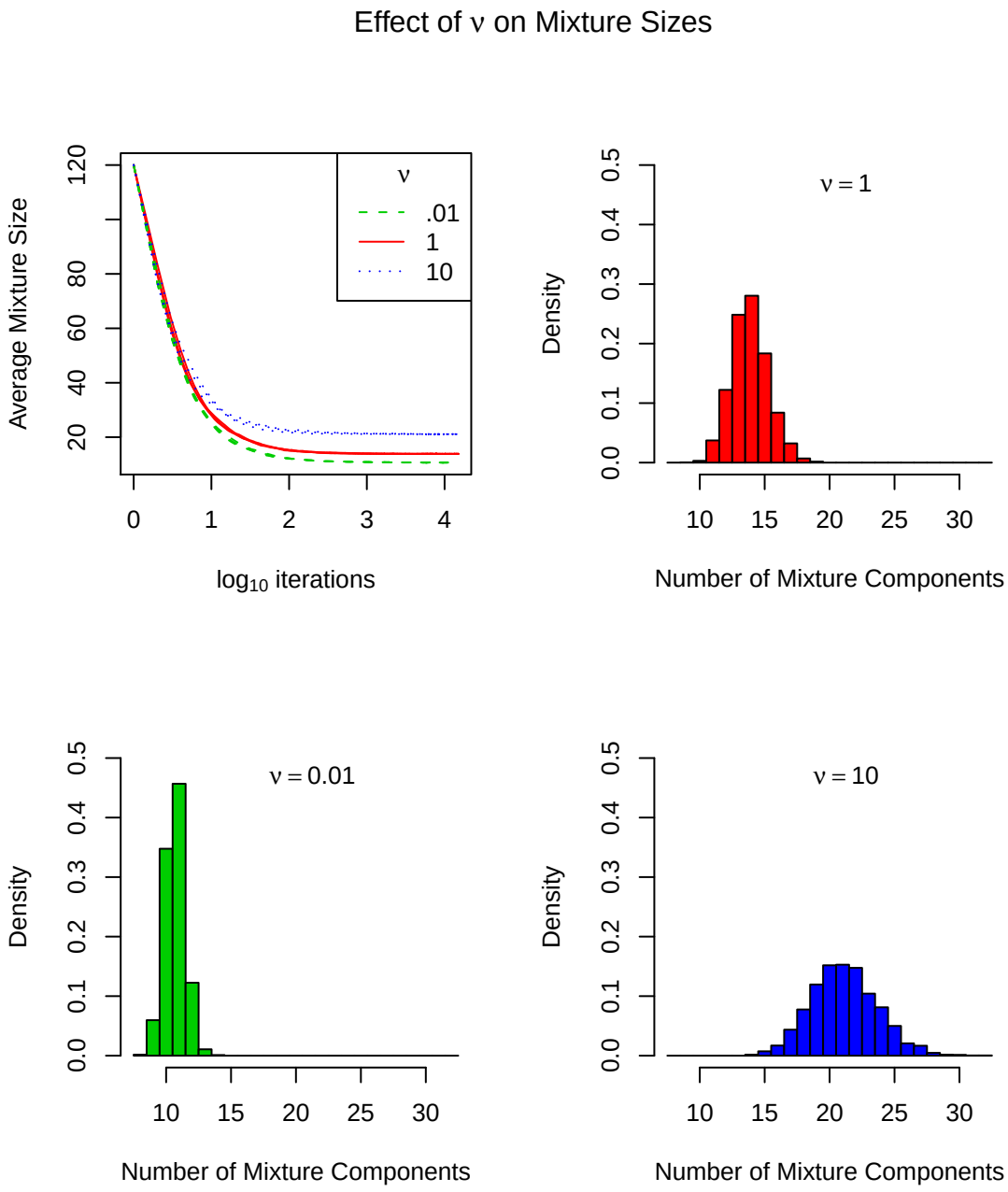

Figure 6.11: Posterior distribution of the number of components in the second simulation study with $(\nu, \tau, d) = (\nu, 1, 2)$.

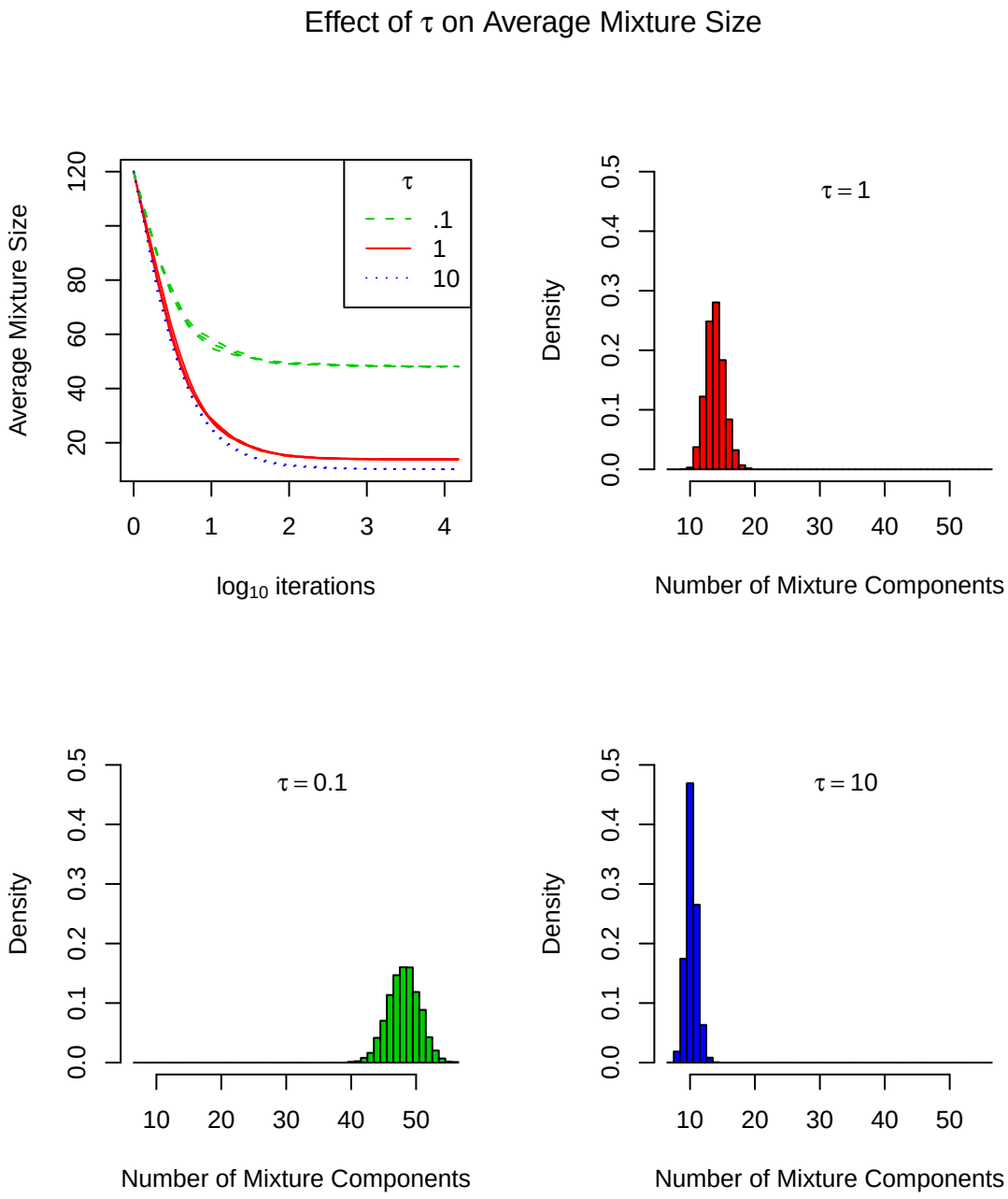

Figure 6.12: Posterior distribution of the number of components in the second simulation study with $(\nu, \tau, d) = (1, \tau, 2)$.

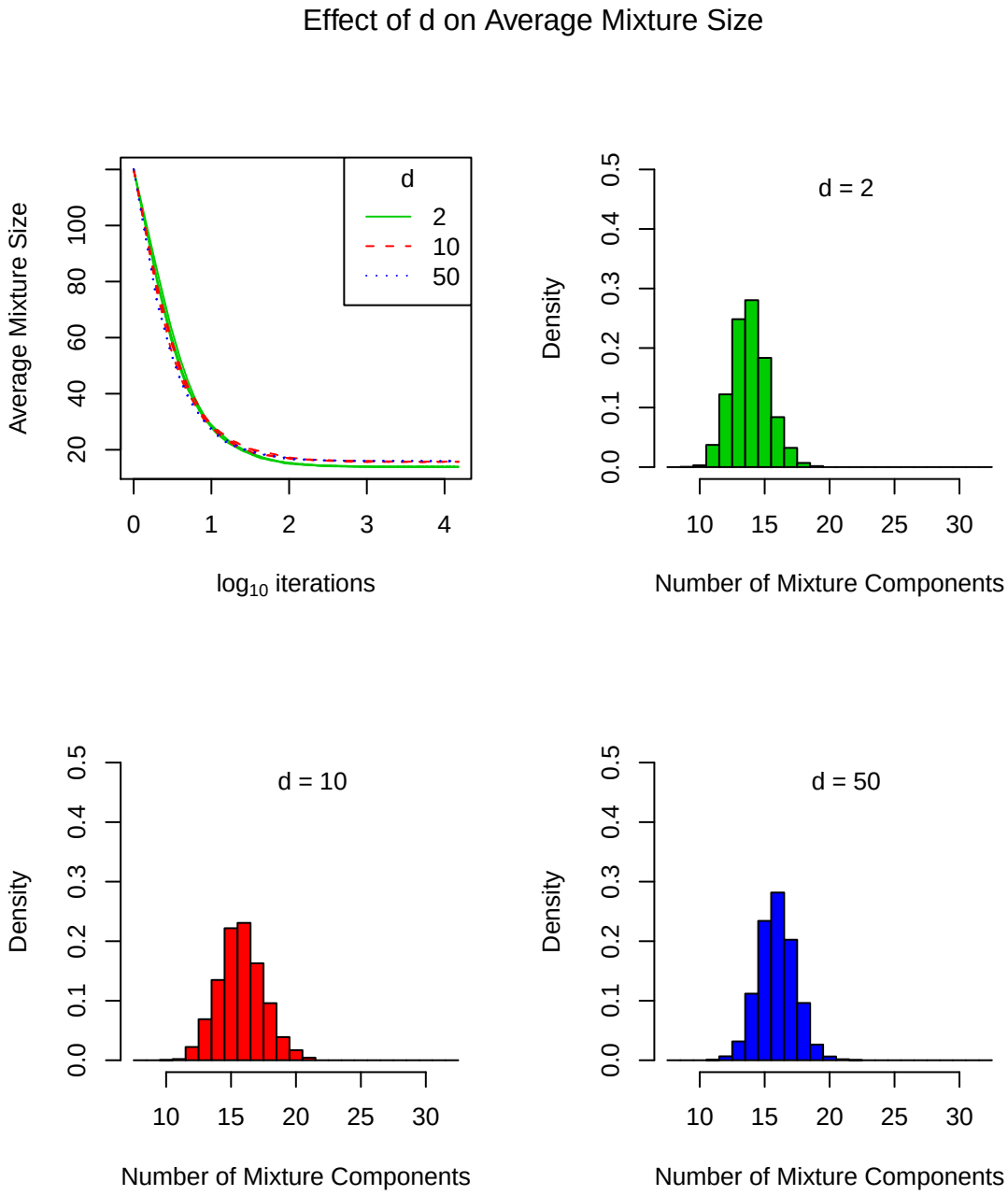

Figure 6.13: Posterior distribution of the number of components with $(\nu, \tau, d) = (1, 1, d)$ in the second simulation study. The green lines pertaining to $d = 50$ are occluded by the blue lines for $d = 10$ because the posterior means are almost identical.

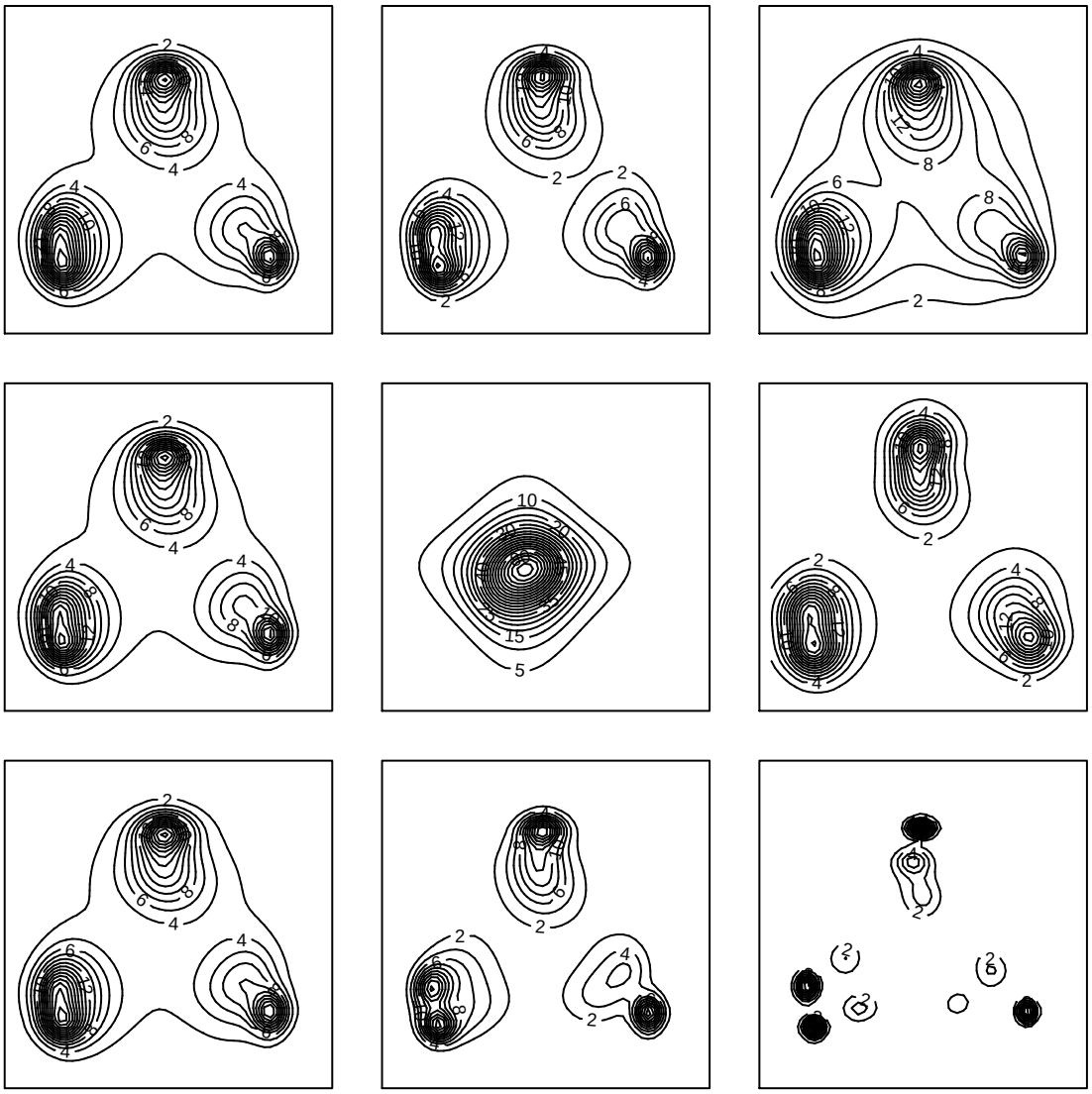

Figure 6.14: Comparison of posterior density estimates in the second simulation study. The top row compares models with ν varying from $1, .01, 10$ (left to right) with $(\tau, d) = (1, 2)$. The middle row compares models with τ varying from $1, .1, 10$ with (ν, d) fixed at $(1, 2)$. The bottom row compares models with d varying from $2, 10, 50$ with $\nu = \tau = 1$.

6.4 Comparison of Hyper Dirichlet Mixture and Kernel Density Estimation

To assess the benefits of using a hyper Dirichlet process, we compared posterior distributions from our model against kernel density estimates using an independent bivariate Gaussian kernel. As we see in this section, Dirichlet process mixtures are similar to using local bandwidth selection. As our goal here is simply to understand Dirichlet process mixtures using the terminology and parameters of kernel density estimates, we prefer to study the behavior of kernel density esimates using various bandwidths instead of more formal selections using cross-validation.

Figure 6.15 shows results from the first simulation study. The first three plots are density estimates from various bandwidths and the fourth plot is the posterior density from the hyper Dirichlet mixture model using our baseline parameters $\nu = \tau = 1$ and $d = 2$. An immediate observation is that the kernel density estimate in the topleft plot fits very poorly. This is because the bandwidth is the same in both dimensions. The observations have been scaled to have unit variance, but the resulting structure is two very non-spherical density components. Of course, we can fix this problem in the two-dimensional case by choosing separate bandwidths. In the bottomleft plot, we can get a very good fit by using a bandwidth of 0.2 for the x-axis and a bandwidth of 1 for the y-axis. For p dimensions, we would need to determine a good bandwidth for each dimension separately, for example by cross-validation. This process does not work well if the various components have different variances, as in the next two simulations.

Figure 6.16 compares kernel density estimates to they hyper Dirichlet mixture posterior for the second simulation study. For this data, using equal bandwidths for both dimensions works reasonably well. Here, the problem is that the components in the bottom corners are roughly spherical, but the group at the top is more oblong. Therefore, if equal bandwidths are chosen, the kernel density estimate yields ill-fitting circular contours for the topmost component. This can be seen in any of the three kernel density estimates for this data. By contrast, the hyper Dirichlet mixture fits differently shaped distributions in each of the three areas.

The effect of components with unequal variances can be seen even more clearly in Figure 6.17. This figure shows estimates from a third simulation of 120 observations generated with the following parameters:

i	μ_i	V_i	n_i
1	$\begin{pmatrix} -5 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$	50
2	$\begin{pmatrix} 5 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	30
3	$\begin{pmatrix} 0 \\ 5 \end{pmatrix}$	$\begin{pmatrix} .5 & 0 \\ 0 & 4 \end{pmatrix}$	40

As before, we scaled and centered the observations so that the overall data had mean 0 and variance I_2.

Figure 6.17 shows that the kernel density estimation simply cannot handle this type of mixture distribution well. The fit for the group in the lowerleft corner is reasonable, but the contours around the other two groups do not match the shape of the data. By contrast, the hyper Dirichlet mixture fits all three groups reasonably well. Of particular note is that the mixture model identified the small variance of the lowerright group, which is indicated by the relative proximity of successive contours compared to contours for the other groups.

The wisdom of Escobar and West (1995) seems to provide a good clue about the relationship between the kernel density estimate and a Dirichlet mixture. They state that in restricted Dirichlet mixtures, for which each group has variance $V_i = V$, the choice of τ relates to the choice of window-widths in traditional kernel density estimation. Logically, it follows that by allowing V_i to vary by component, we are essentially working with local bandwidth selection. This concept is illustrated nicely by our analysis of the second and especially the third simulation studies. The posterior estimate from the hyper Dirichlet mixture allows the spread of the Gaussian components to vary throughout the range of the observations. Of course, it is theoretically possible to calculate kernel density estimates with local bandwidths, but this is increasingly more complex as the dimension increases. By contrast, the hyper Dirichlet mixture that we have presented worked reasonably well with a $HIW(2, I_2)$ prior for the component covariance matrices.

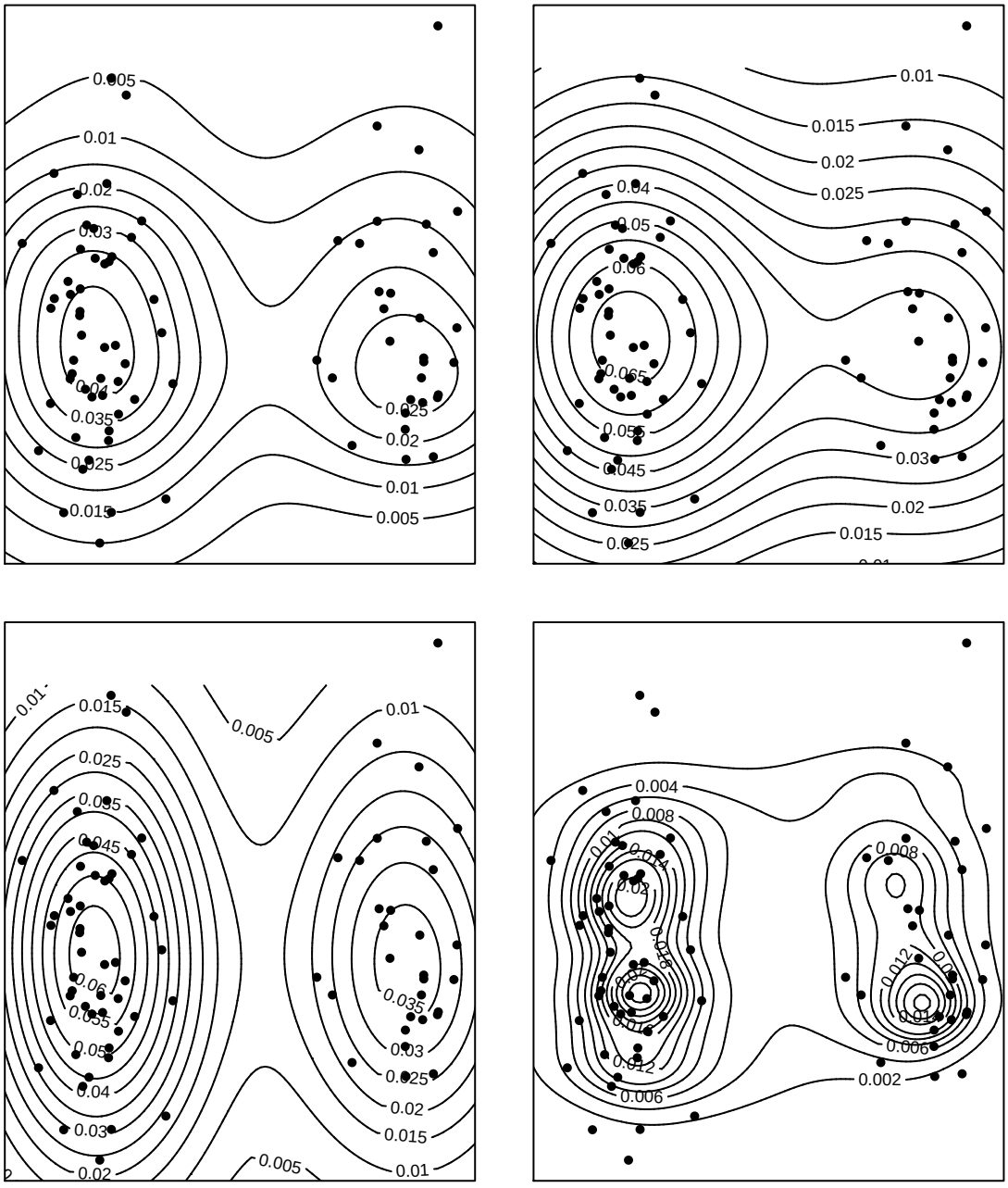

Figure 6.15: Data from Simulation 1 - Bivariate kernel density estimation with bandwidths of $(.3, .3)$ (topleft); $(.5, 1)$ (topright); and $(.2, 1)$ (bottomleft). On the bottomright is the posterior distribution from the hyper Dirichlet mixture model with $(\nu, \tau, d) = (1, 1, 2)$.

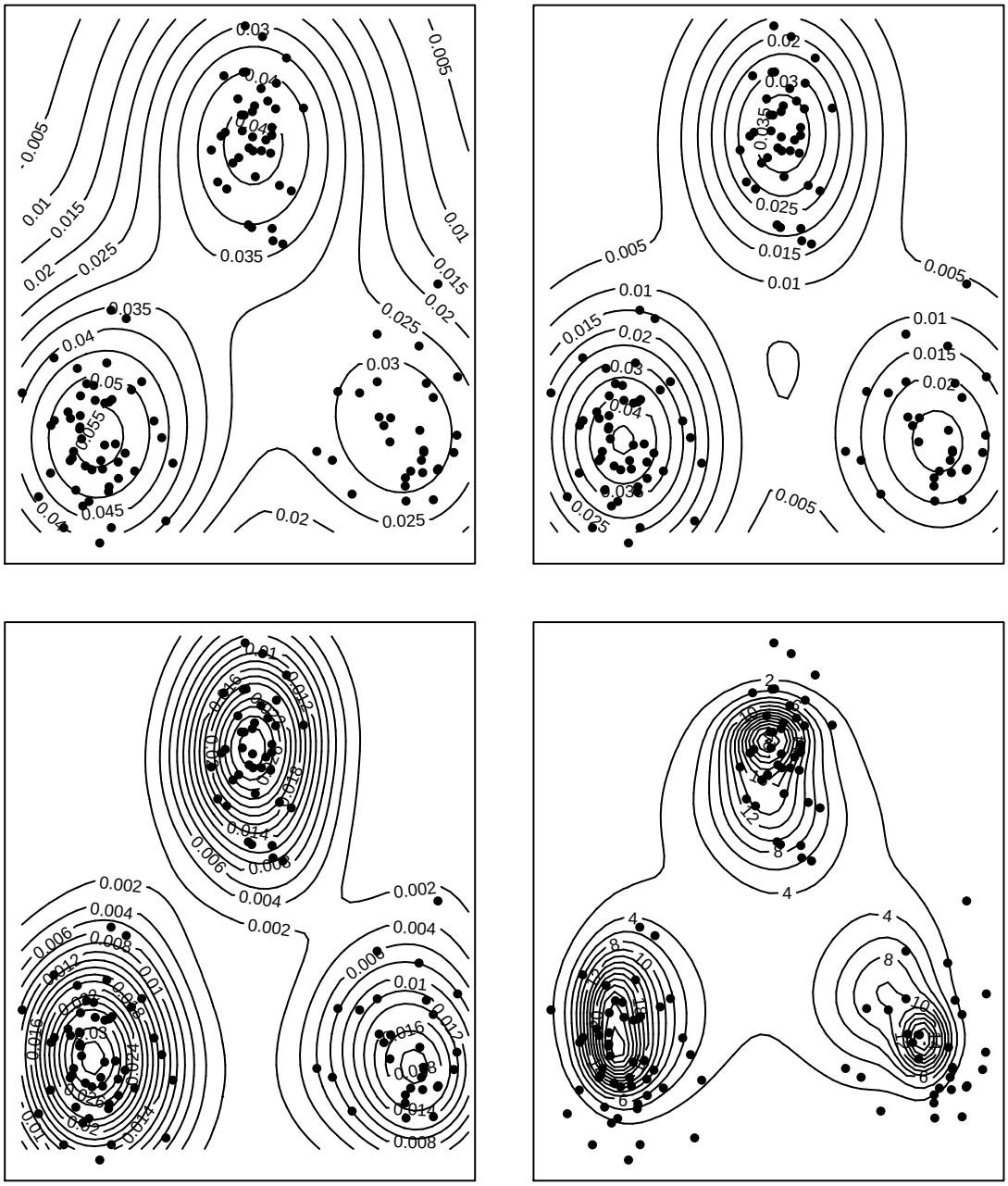

Figure 6.16: Data from Simulation 2 - Bivariate kernel density estimation with bandwidths of $(.5, .5)$ (topleft); $(.2, .2)$ (topright); and $(.1, .1)$ (bottomleft). On the bottomright is the posterior distribution from the hyper Dirichlet mixture model with $(\nu, \tau, d) = (1, 1, 2)$.

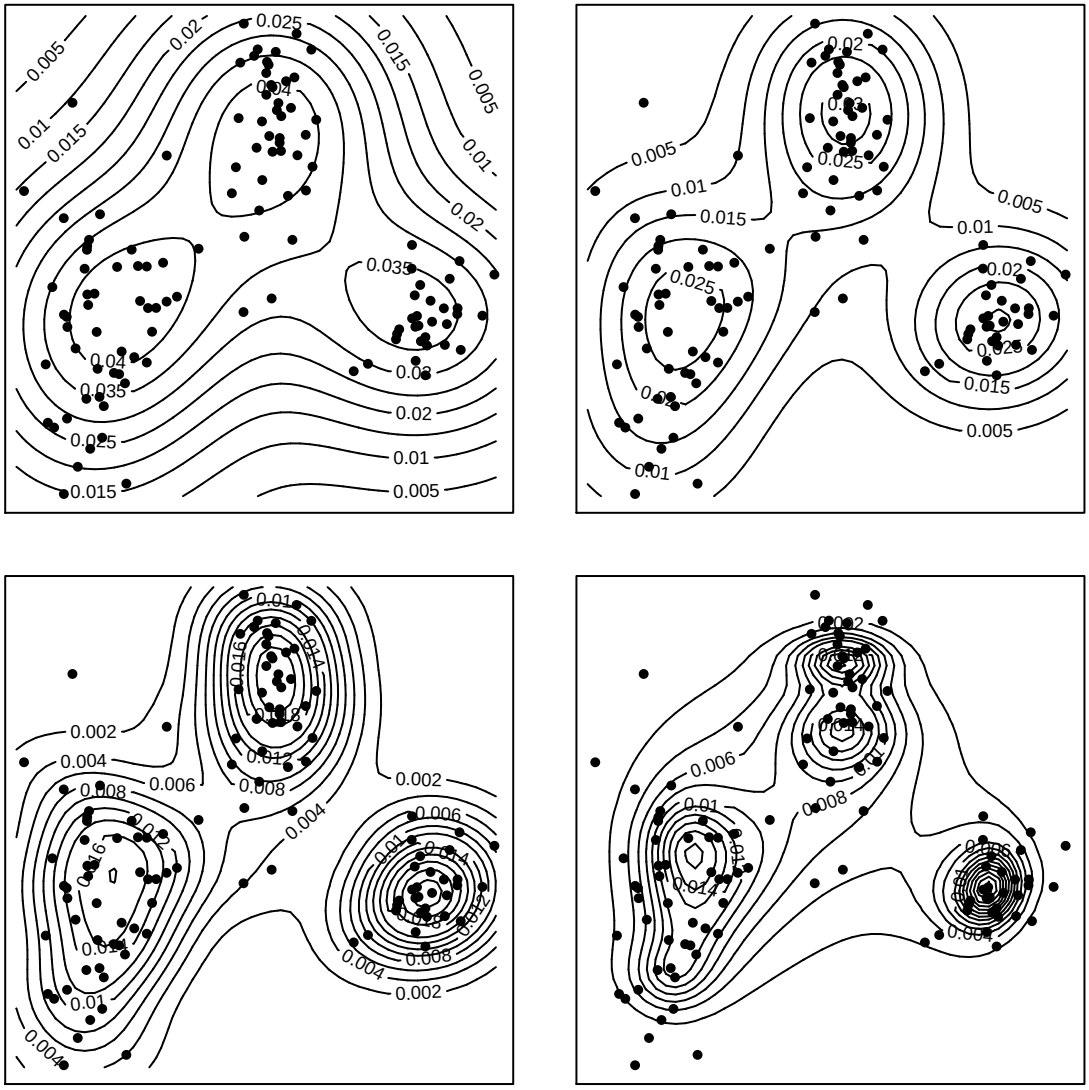

Figure 6.17: Data from Simulation 3 - Bivariate kernel density estimation with bandwidths of $(.5, .5)$ (topleft); $(.2, .2)$ (topright); and $(.1, .1)$ (bottomleft). On the bottomright is the posterior distribution from the hyper Dirichlet mixture model with $(\nu, \tau, d) = (1, 1, 2)$.

6.5 Inference for ν and τ

Sections 6.2 and 6.3 show some insight into how the hyperparameters affect inference. In particular, we saw that changing ν and τ induced differing numbers of components in the posterior as well as changes in the actual shape of the density. Therefore, we may desire to let these parameters vary by specifying a prior distribution. The inverse-Gamma distribution is a conjugate prior for τ; Escobar and West (1995) provide a clever method for updating ν when it is given a Gamma prior.

Of course, we could also consider setting priors for the other parameters, d, m, and D. For our purposes, we will not pursue this. Regarding the degrees of freedom parameter, d, we noted that it had very little effect on the number of mixture components. We also saw in Figures 6.8 and 6.14 that large values of d lead to very tight Gaussian components that seem to overfit the data in the simulation studies. Therefore, it is best to choose a small value for $d > 0$ and the results seem to be fairly stable unless we choose large d. Regarding m and D, we simply choose to center and scale the data, then set m to the zero vector and D to the identity matrix. For our simulations, the component groups are roughly equidistant and of similar size. Thus, the overall mean is close to the average of the group means. In more general settings, we could use a conjugate Normal prior for m, but we eschewed this addition in favor of simplicity. For the same reason, we decided not to introduce a prior for D. As noted in the comparison with the kernel density estimates, one of the benefits of the hyper Dirichlet mixture is a sort of automatic local bandwidth selection. In other words, the choice of $D = I$ seems to do reasonably well in our studies.

Suppose $\tau \sim IG(w, W)$ and that the data ($\vec{X}$), group means ($\vec{\mu}^*$), group covariance matrices ($\vec{V}^*$), group assignments ($\vec{t}$) and all the other parameters (ν, m, d, D) are given. Let n be the number of observations and k the number of unique parameter values.

$$\begin{aligned} & f(\vec{X}, \vec{\mu}^*, \vec{V}^*, \tau | \nu, \vec{t}, m, d, D, w, W) \\ &= \prod_{i=1}^{n} dHN_{\mathcal{G}}(X_i; \mu^*_{t_i}, V^*_{t_i}) \prod_{i=1}^{k} dHN_{\mathcal{G}}(\mu^*_i; m, \tau V^*_i) \\ &= \times \prod_{i=1}^{k} dHIW_{\mathcal{G}}(V^*_i; d, D) dIG(\tau; w, W), \end{aligned} \tag{6.62}$$

whence we see that $\tau | (w, W, m, \vec{V}^*)$ is conditionally independent of everything else. Furthermore,

$$\begin{aligned} & f(\tau | \vec{\mu}^*, \vec{V}^*, w, W) \\ &\propto \tau^{-|\mathbf{V}|k/2} \exp\left\{ -\left(\frac{1}{2} \sum_{i=1}^{k} (\mu^*_i - m)^T (V^*)^{-1}_i (\mu^*_i - m) \right) \tau^{-1} \right\} \\ &\quad \times \tau^{-(w+1)} \exp\{-W/\tau\} \end{aligned} \tag{6.63}$$

Therefore, the posterior distribution of τ is Inverse-Gamma with shape $|\mathbf{V}|k/2 + w$ and inverse-scale equal to $\frac{1}{2}R + W$, where $R = \sum_{i=1}^{k} (\mu^*_i - m)^T (V^*)^{-1}_i (\mu^*_i - m)$.

We next present the scheme for updating ν, which is due to Escobar and West (1995). Recall that ν implies an indirect prior for k, the number of distinct parameter values. From Antoniak (1974), we know that the prior distribution of k follows

$$p(k|\nu, n) = c_n(k) n! \nu^k \frac{\Gamma(\nu)}{\Gamma(\nu + n)}, \tag{6.64}$$

where $c_n(k) = p(k|\nu = 1, n)$. Note that

$$\frac{\Gamma(\nu)}{\Gamma(\nu + n)} = \frac{\Gamma(\nu)\Gamma(\nu + n + 1)B(\nu + 1, n)}{\Gamma(\nu + n)\Gamma(\nu + 1)\Gamma(n)}, \tag{6.65}$$

where $B(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1} = \Gamma(a)\Gamma(b)/\Gamma(a + b)$ is the beta function. Using the identity $\Gamma(a + 1) = a\Gamma(a)$, the left hand side reduces to $(\nu + n)B(\nu + 1, n)/(\nu\Gamma(n))$. Thus, given a prior $f(\nu)$ for ν, we see that

$$f(\nu|k, n) \propto f(\nu)\nu^{k-1}(\nu + n) \int_0^1 x^{\nu}(1 - x)^{n-1} dx, \tag{6.66}$$

where we have used the integral form of the beta function.

As Escobar and West (1995) point out, Equation 6.66 implies that the posterior distribution of ν can be calculated as the marginal from a joint distribution for $\nu > 0$ and $0 < \eta < 1$ where $f(\nu, \eta) = f(\nu)\nu^{k-1}(\nu + n)\eta^{\nu}(1 - \eta)^{n-1}$. From the form of Equation 6.66 we see that $f(\eta|\nu, k, n) \propto \eta^{\nu}(1-\eta)^{n-1}$, which is the Beta$(\nu+1, n)$ distribution as described in Chapter 3. Furthermore, under a Gamma(a, b) prior for ν, we see that

$$f(\nu|\eta, k, n) \propto \nu^{a-1}e^{-b\nu}(\nu + n)\nu^{k-1}\eta^{\nu} \tag{6.67}$$

$$\propto \nu^{a+k-1}e^{-(b-\log(\eta))\nu} + n\nu^{a+k-2}e^{-(b-\log(\eta))\nu}. \tag{6.68}$$

Equation 6.68 is clearly a mixture of two Gamma distributions. We simply need to fill in the appropriate normalization constants to find the mixing weights. This leads to

$$\begin{aligned} f(\nu|\eta,k,n) \quad \propto \quad & \frac{\Gamma(a+k)}{(b-\log(\eta))^{\nu+k}} \cdot dG(\nu; a+k, b-\log(\eta)) \\ & + \frac{\Gamma(a+k-1)}{(b-\log(\eta))^{\nu+k-1}} \cdot dG(\nu; a+k-1, b-\log(\eta)). \end{aligned} \tag{6.69}$$

The methods outlined in this section provide a simple method to extend our Gibbs sampler to include inference for τ and ν. We simply need to include an extra step at the end of each iteration in which we sample values for τ and ν from their respective conditional distributions. For τ, this is as simple as generating τ^{-1} from its conditional Gamma distribution given the unique parameter values and m. Sampling ν is marginally more complicated and requires a few extra steps. We first generate a random η from its conditional Beta distribution given ν and k. Secondly, we generate U from the uniform distribution on $(0,1)$. If $U \leq \Gamma(a+k)/(b-\log(\eta))$, then we generate ν from the Gamma$(a+k, b-\log(\eta))$ distribution; otherwise, we generate ν from the Gamma$(a+k-1, b-\log(\eta))$ distribution. After making this change to our Gibbs sampler, we can estimate the posterior distributions of τ and ν as normal using values from the Gibbs sampler after it has converged.

A simple way to detect convergence is to see when some of the summary statistics reach their limit. Figure 6.18 shows the mean number of components, the sample average of τ, and the sample average of ν plotted against the iteration number for all three simulations. It is immediately obvious from these plots that the variance in the number of components is large compared to our previous analyses with ν and τ fixed. This is not surprising, but

we see that a much longer burn-in is required. For all of the measures and all simulations, a burn-in of 10000 seems adequate, and we used 15000 to be conservative. As before, autocorrelation plots revealed significant correlation as far as lag 5 or 6. A second series of plots showed that taking every 10^{th} iteration is sufficient to get independent samples.

As Figure 6.19 reveals, the posterior estimate from this model fits the observed data quite well. In contrast to the earlier studies, there is little to no shrinkage toward the overall center. Each row of this figure pertains to one of the three simulated data sets. By comparing the plots in each row, we get a sense of whether or not the Gibbs sample is large enough. In particular, there are some small differences in the first and third row, which suggests that a slightly larger number of iterations would be useful. Histograms of sample draws for the mixture size, ν, and τ agree with this assessment.

Figures 6.20-6.22 show the estimated posterior distributions for the mixture size, ν, and τ. These are combined estimates using three runs of the Gibbs sampler for each simulation, so the effective sample size is 3000. The histograms reveal interactions between the various statistics. Notably, Simulation 2 contains fewer mixture components that either Simulation 1 or 3. This is odd at first glance because there are actually 50% more observations in Simulation 2, relative to Simulation 1. This is due to the fact that the within group variance is smaller relative to the between group variance in Simulation 2. In any case, this allows us to see an interesting interaction between the three statistics. We note that the smaller number of mixture components in Simulation 2 is paired with smaller estimates for the precision ν. This is to be expected since k increases stochastically with ν. Less obvious, though still somewhat intuitive, is that the standard error for estimating τ is large for

Simulation 2 compared to the other two simulations. With fewer unique parameter values, there is less information about τ. We also note that the estimates for τ are smallest in Simulation 3. This is to be expected since the three Gaussian components are more proximate than in Simulations 1 and 2. Hence, the variance for the unique μ_i^* is smaller. Finally, we note one more feature of the histograms. Namely, the posterior density of mixture size and τ are very similar between Simulations 1 and 3. This is odd since the two distributions are quite different. In fact, Simulation 3 is a lot closer to Simulation 2 in most, if not all, respects. For example, we compared sample variances for all three simulations, but the variances in Simulation 2 and 3 were very similar and different than the variance in Simulation 1. It seems therefore, that this similarity is mere coincidence, but it is not certain.

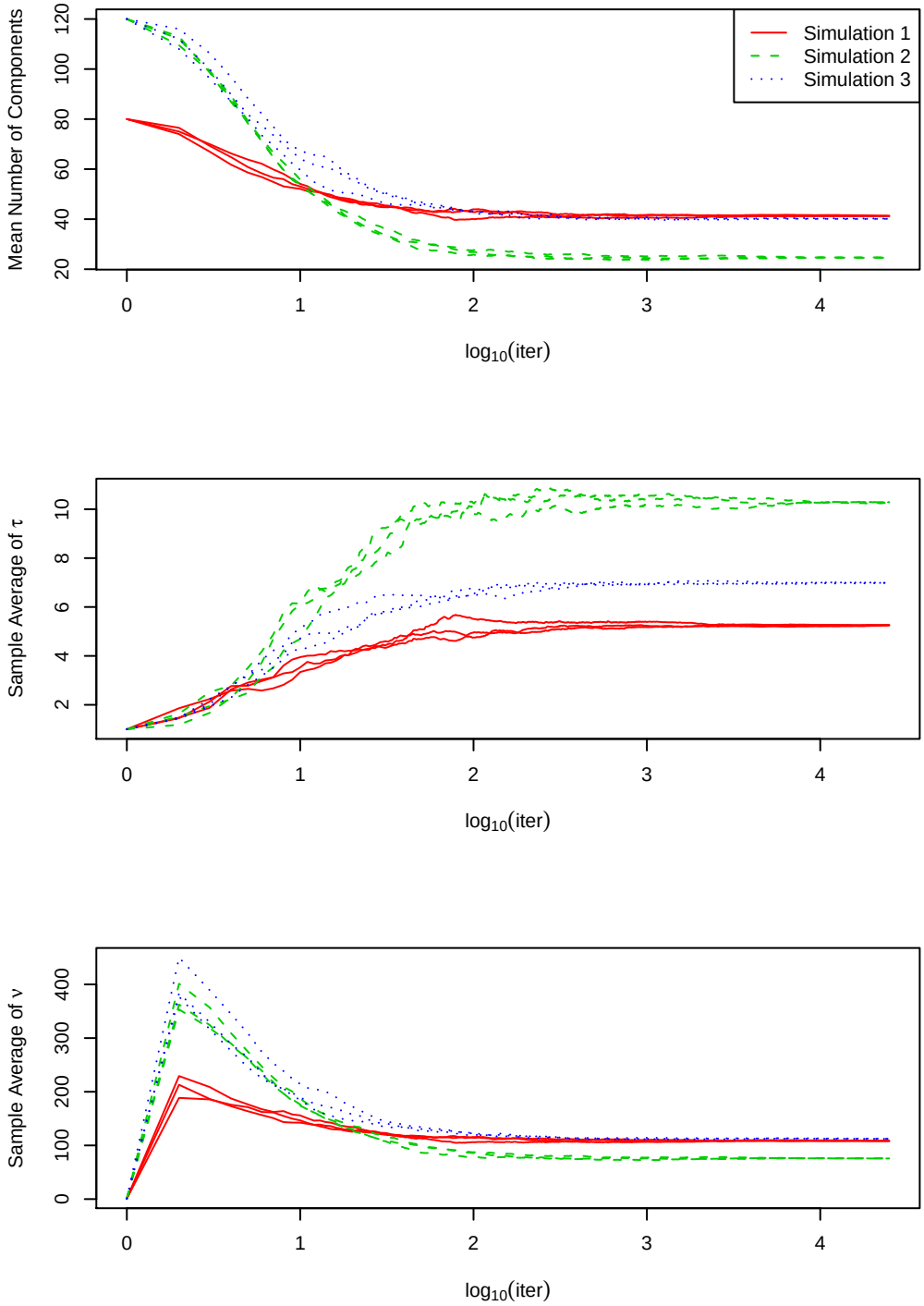

Figure 6.18: Converge of mixture sizes, ν, and τ for all three simulation studies with a prior for ν and τ.

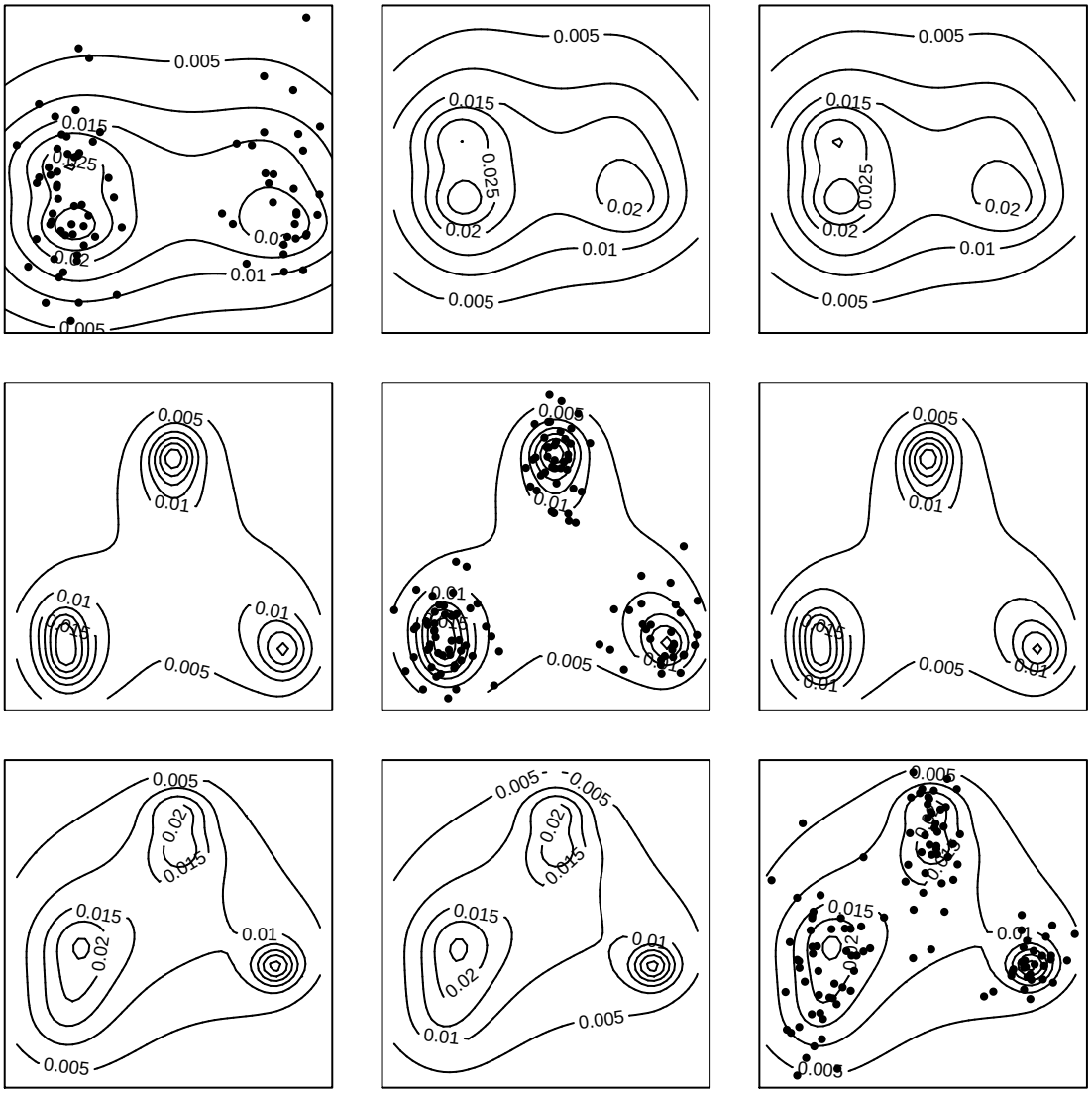

Figure 6.19: Comparison of posterior density estimates for independent Gibbs samples with a prior for ν and τ for Simulation studies 1 (top) to 3 (bottom).

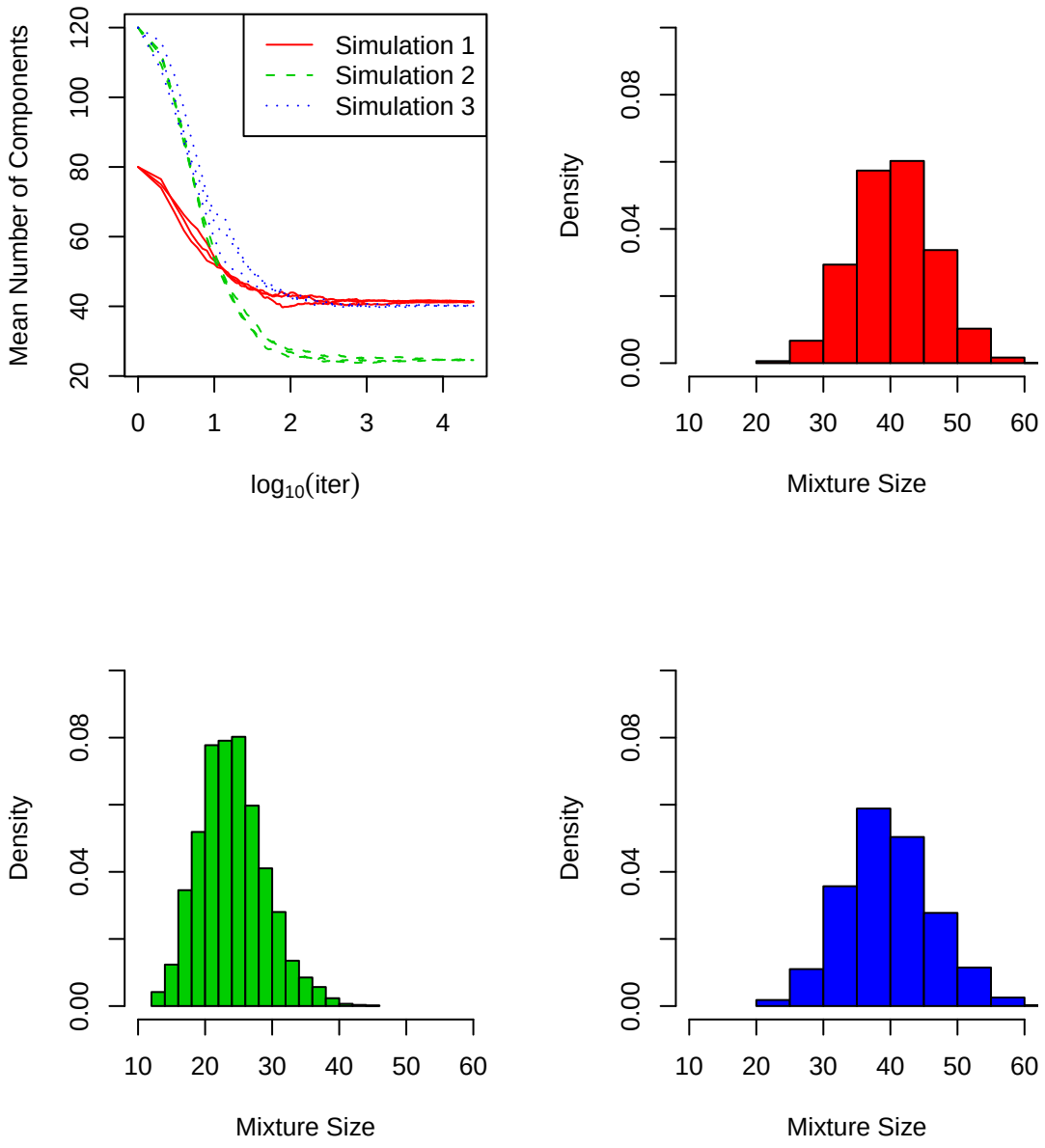

Figure 6.20: Posterior density of mixture sizes with a prior for ν and τ in Simulation 1 (topright), Simulation 2 (bottomleft), and Simulation 3 (bottomright).

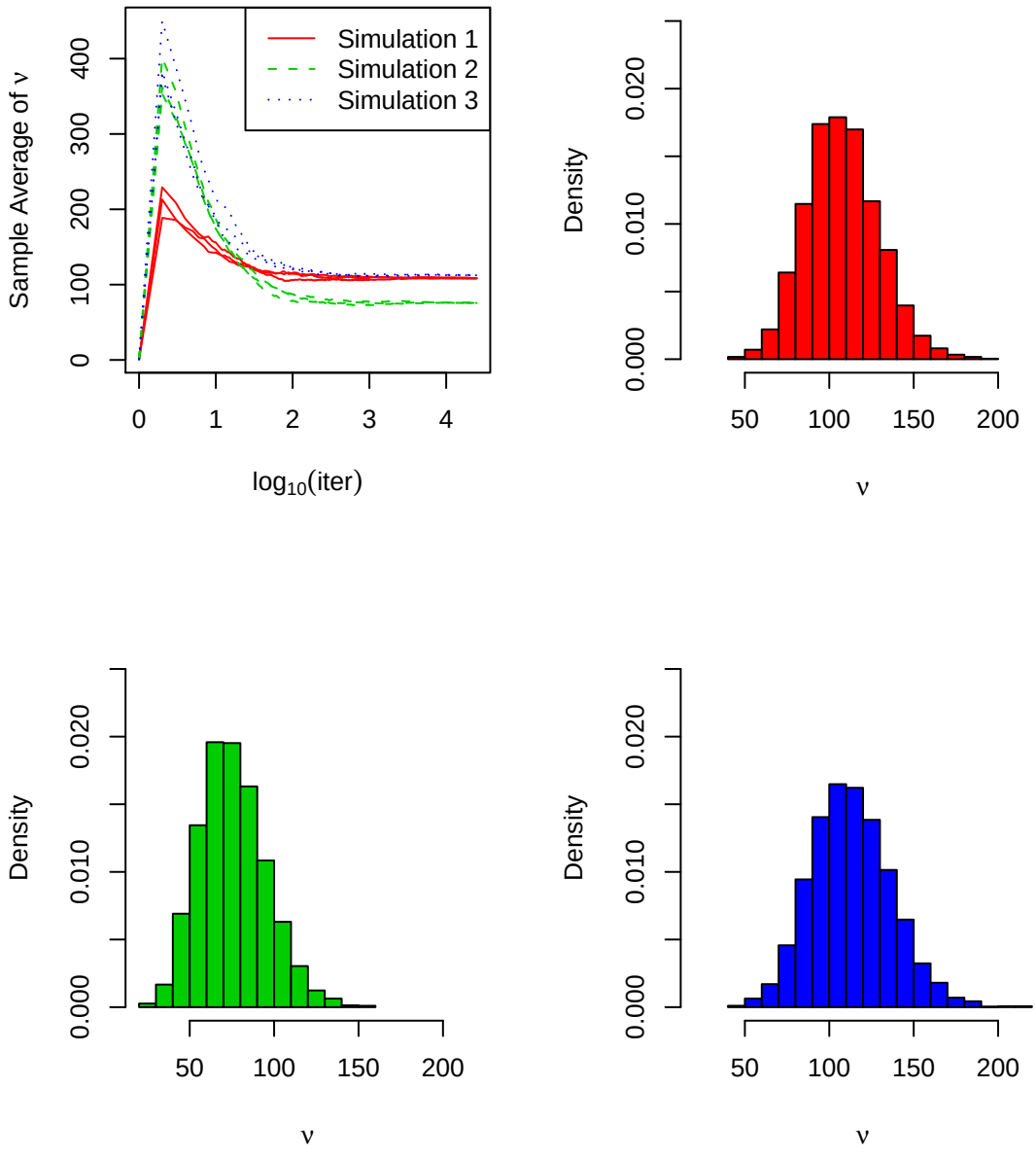

Figure 6.21: Posterior density of ν with a prior for ν and τ in Simulation 1 (topright), Simulation 2 (bottomleft), and Simulation 3 (bottomright).

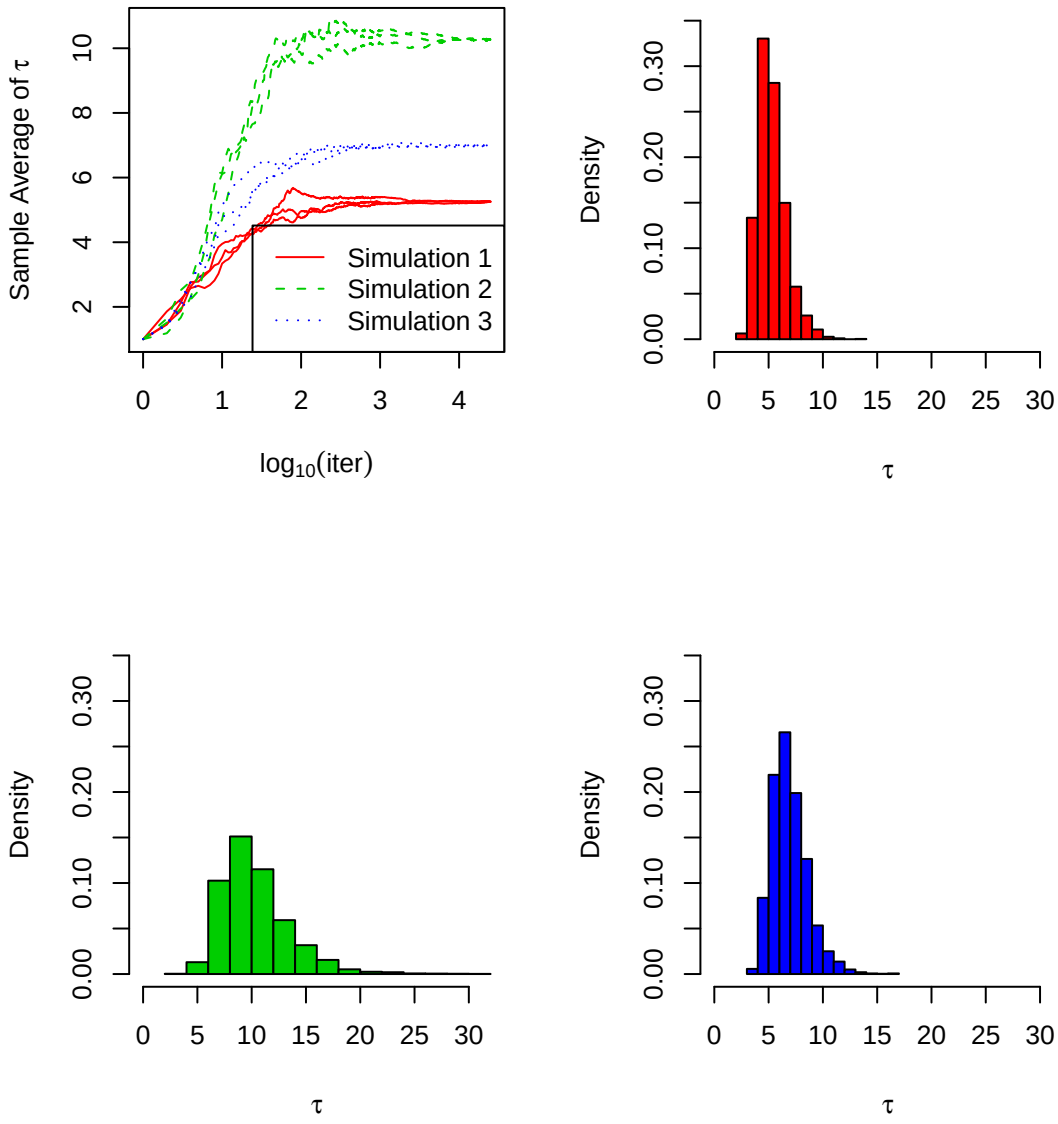

Figure 6.22: Posterior density estimates of τ with a prior for ν and τ in Simulation 1 (topright), Simulation 2 (bottomleft), and Simulation 3 (bottomright).

Chapter 7

Model Selection with Graphical Dirichlet Processes

We now show how to use a graphical Dirichlet Process to choose among competing models. Specifically, we will compare graphical Dirichlet process mixtures whose components are hyper Markov models with respect to various graphical models. In the Bayesian paradigm, we may choose the model with the highest posterior probability. Alternatively, we may select a set of competing models with similar posterior probabilities, either to establish a credible set for the model or to narrow the choice of models from which to choose using other criteria (e.g. parsimony). We shall see that choosing a good model is much easier than actually fitting a model. This is due to Theorem 4.5.3, which states that the parameter distribution is Markov given the cluster assignments (and a graph). In the terminology of the Chinese restaurant process, the distribution of the parameters (dishes) satisfies

the conditional independence once we account for each patron's table choice. Therefore, posterior calculations can be made rather easily by a Monte Carlo method in which we randomize table assignments, which allows us to compute the marginal distribution for each table analytically in the case of conjugate models. The benefit to this approach is that the graphical Dirichlet process allows us to fit a graphical model which takes into account a latent class variable with an unspecified number of levels. In this chapter, we focus on the Mode-Oriented Stochastic Search (Dobra and Massam, 2009) to select a graph for the graphical Dirichlet mixture model. We could also consider other stochastic searches such as the shotgun stochastic search of Jones et al. (2005) and Hans et al. (2007) or the MC^3 algorithm of Madigan and York (1995).

7.1 Monte Carlo Estimation of the Marginal Likelihood

Denote the set of competing models by $\mathscr{G}$, where $\mathcal{G} \in \mathscr{G}$ is a decomposable graphical model. In the absence of expert information, we may choose to use an uninformative prior. For a set of variables, $|V|$, there are $\binom{|V|}{k}$ possible k-way interactions. As a result, $|\mathscr{G}|$ is bounded by above by $\sum_{k=1}^{|V|}\binom{|V|}{k}$, so the prior $p(\mathcal{G}) = |\mathscr{G}|^{-1}$ is both proper and uninformative. On the other hand, if expert information is available, a more complicated prior may be used. For example, we may choose to put more prior weight on parsimonious models.

To calculate the posterior probability of $\mathcal{G} \in \mathscr{G}$, we simply weigh the data likelihood given $\mathcal{G}$ by the prior probability of $\mathcal{G}$. All that is required is a method for calculating $f(\vec{X}|\mathcal{G})$. Note that here, and throughout this chapter, we treat any parameters that describe the

distribution of $\vec{X}$ as nuisance parameters. For example, if the distribution of $\vec{X}$ given $\mathcal{G}$ is a non-parametric graphical Dirichlet mixture of Gaussians as in Chapter 6, then we wish to integrate out the parameters $\vec{\mu}$ and $\vec{V}$ as well as the random Dirichlet process measure P. If needed, we can use the methods of Chapter 6 to find posterior densities for parameters once we have narrowed down the candidate models to a more manageable size.

As a consequence of a Dirichlet process prior, it is difficult to integrate out the nuisance parameters analytically. On the other hand, the difficulty of an analytical solution lies in accounting for the unknown cluster assignments. This is to say that if the cluster assignments were known, then the marginal calculation may be carried out in the same way as any other Bayesian hierarchical model within each component. In the particular case of a conjugate model, we can calculate the marginal distribution of the data using a straightforward Monte Carlo integral. Alternatively, we could consider a more complicated Markov chain Monte Carlo estimation when the model is not conjugate.

For a particular graph, we have the following hierarchical model:

$$\begin{aligned} P|\mathcal{G} &\sim DP(\nu, H_{\mathcal{G}}) \\ \theta_i &\sim P \\ X_i|\theta_i &\sim F(X_i|\theta_i), \end{aligned}$$

where the base measure on $H_{\mathcal{G}}$ is hyper Markov with respect to the graphical model $\mathcal{G}$. Here we have made explicit the conditioning on $\mathcal{G}$. It is not necessary for P to be hyper Markov; a graphical Dirichlet process suffices. We prefer to use a graphical model in this

chapter for simplicity. In particular, we need do not need a separate Dirichlet process for each connected component as is required for a hyper Dirichlet process.

Using $\mathcal{P}$ to denote the support of P, the marginal likelihood for $X_1, \ldots, X_n$ is

$$f(\vec{X}|\mathcal{G}) = \int_{\Theta^n} \int_{\mathcal{P}} f(\vec{X}, \vec{\theta}, P|\mathcal{G}) dP d\vec{\theta} \quad (7.1)$$

$$= \int_{\Theta^n} \int_{\mathcal{P}} f(\vec{X}|\vec{\theta}, \mathcal{G}) f(\vec{\theta}, P|\mathcal{G}) dP d\vec{\theta}, \quad (7.2)$$

where we use the fact that $\vec{X}$ given $\vec{\theta}$ is conditionally independent of P. We note that $f(P|\mathcal{G})$ is a Dirichlet Process, and $\vec{\theta}|P, \mathcal{G}$ is a random sample of the random distribution P. Therefore, random deviates from the marginal $f(\vec{\theta}|\mathcal{G})$ can be simulated using a Chinese restaurant process. This bypasses the need to generate P, so we can integrate it out of the previous equation. The marginal for $\vec{X}$ is now

$$f(\vec{X}|\mathcal{G}) = \int_{\Theta^n} f(\vec{X}|\vec{\theta}) f(\vec{\theta}|\mathcal{G}) d\vec{\theta} \quad (7.3)$$

$$= E_{\vec{\theta}}(f(\vec{X}|\vec{\theta}), \quad (7.4)$$

where the expectation is with respect to the conditional distribution $f(\vec{\theta}|\mathcal{G})$. In this form, the Monte Carlo integration is evident. To approximate the marginal for $\vec{X}$ given $\mathcal{G}$, we generate a random sample of $\vec{\theta}$ from independent Chinese restaurant processes, then evaluate the sample average of $f(\vec{X}|\vec{\theta})$.

The estimation can be improved in some cases by considering the form of $\vec{\theta}$ in more detail. Recall that there is positive probability that some values of θ_i coincide. Let $\vec{\theta}^* =$

$(\theta_1^*, \ldots, \theta_k^*)$ be the vector of unique θ_is and define $\vec{t} = (t_1, \ldots t_n)$ where $t_i = j$ if $\theta_i = \theta_j^*$. In other words we are rewriting the vector of parameters in terms of the k clusters and the cluster assignments for each x_i.

$$f(\vec{X}|\vec{\theta}) \quad = \quad f(\vec{X}|\vec{\theta^*}, \vec{t}). \tag{7.5}$$

In place of Equation 7.3, we write

$$f(\vec{X}|\mathcal{G}) = \int_{\mathcal{T}} \int_{\Theta^n} \prod_{j=1} f(\vec{X}|\vec{\theta^*}, \vec{t}) f(\vec{\theta^*}, \vec{t}) d\vec{\theta^*} d\vec{t}, \tag{7.6}$$

where $\mathcal{T}$ is the set of all possible cluster assignments.

The importance of this specification, is that $\vec{t}$ and $\vec{\theta^*}$ are easy to specify using the Chinese restaurant process. $\vec{t}$ is the seating arrangements from the process, which depends only on ν; $\vec{\theta^*}|\vec{t}$ is a simple random sample from $H_{\mathcal{G}}$. Therefore, we may compute the integral over Θ separately for each θ_j^*. Denote the subset of observations with table assignment j by $\mathbf{X}_{j^*}$

$$f(\vec{X}|\mathcal{G}) = \int_{\mathcal{T}} \prod_{j=1}^{k} \left(\int_{\Theta} f(\mathbf{X}_{j^*}|\theta_j^*, \vec{t}, \mathcal{G}) f(\theta_j^*|\vec{t}) d\theta_j^* \right) f(\vec{t}) d\vec{t}, \tag{7.7}$$

where $f(\theta_j^*|\vec{t}) = dH_{\mathcal{G}}(\theta_j^*)$, and $f(\vec{t})$ is the distribution of table assignments induced by the Chinese restaurant process.

Consider the innermost integral of Equation 7.7, $\int_{\Theta} f(\mathbf{X}_{j^*}|\theta_j^*, \vec{t}) f(\theta_j^*|\vec{t})$. This integral is the calculation for the marginal data likelihood of a sample in a Bayesian graphical

model sans clustering. Therefore, we can compute the integral (i.e. $f(\mathbf{X}^{(j)}|\vec{t},\mathcal{G})$) using procedures that are already well-known. For example, in conjugate cases, we may compute this integral analytically. We calculate this density once for each component, then the full integral, which equals $f(\vec{X}|\vec{t},\mathcal{G})$, is simply the product of these component densities. After integrating out the unique parameters, we see

$$f(\vec{X}|\mathcal{G}) = \int_{\mathcal{T}} f(\vec{X}|\vec{t},\mathcal{G})d\vec{t} = \mathbb{E}[f(\vec{X}|\vec{t})], \tag{7.8}$$

with expectation relative to $f(\vec{t})$. Antoniak (1974) specified the distribution of $f(\vec{t})$, but it is difficult to integrate analytically. Fortunately, it is easy to generate random deviates using the Chinese restaurant process. In other words, we can approximate $f(\vec{X}|\mathcal{G})$ by randomly generating the cluster assignments, $\vec{t}$, and averaging the sample values of $f(\vec{X}|\vec{t},\mathcal{G})$. This is specified in the next algorithm.

Algorithm 7.1.1. *Monte Carlo Estimate of Data Distribution*

Set $f \leftarrow 0$.

For $b = 1, \ldots, B$*:*

1. *Set* $t_1 \leftarrow k \leftarrow 1$.
2. *For* $i = 2, \ldots, n$*:*
   (a) *Generate* $U \sim$ *Unif*$(0, \nu + i - 1)$.
   (b) *If* $U > i - 1$ *then set* $t_i \leftarrow k \leftarrow k + 1$,
   otherwise set $t_i \leftarrow j$, *where* $\sum_{l=1}^{j-1} < U \leq \sum_{l=1}^{j}$.

3. Set $f_b \leftarrow 1$.

4. For $j = 1, \ldots, k$*:*

(a) Set $f \rightarrow f \cdot f(X^{(j)}|\mathcal{G})$*, where* $X^{(j)}$ *is the subset of data such that* $t_i = j$ *and* $f(X^{(i)}|\mathcal{G})$ *is its marginal density for the non-mixture graphical model.*

5. Set $f \leftarrow f + f_b$.

Approximate $f(\vec{X}|\mathcal{G})$ *by* f/B.

Note that the only step in Algorithm 7.1.1 that depends on the specific model is the calculation of $f(X^{(j)}|\mathcal{G})$. Furthermore, this is one of the rudimentary calculations that Bayesians work with routinely. In particular, we have noted that we may compute this marginal easily for conjugate models. In more complex settings, this step may be a numerical integration or possibly a full Markov chain Monte Carlo integral. As examples of the algorithm, we compute this density for a graphical Dirichlet mixture of Gaussians, as well as a graphical Dirichlet mixture of multinomials.

7.1.1 Covariance Selection with a Dirichlet Mixture of Gaussians

Let $\mathcal{G}$ be a given decomposable graphical model. Recall that a graphical Dirichlet mixture of Gaussians, is a hierarchical model following the form:

$$\begin{aligned} P|\mathcal{G} &\sim DP(\nu, H_{\mathcal{G}}) \\ \mu_i, V_i|P &\sim P \\ X_i|\mu_i, V_i &\sim HN_{\mathcal{G}}(\mu_i, V_i) \end{aligned}$$

where the subscript $\mathcal{G}$ denotes a measure that is (hyper) Markov with respect to $\mathcal{G}$, $H_{\mathcal{G}}$ is a $HN_{\mathcal{G}} \times HIW_{\mathcal{G}}$ base measure, and $HN_{\mathcal{G}}(\mu_i, V_i)$ is the hyper Normal distribution with mean μ_i and covariance matrix V_i. Momentarily, we consider the smaller problem in which there exists a random sample $X_1, \ldots, X_n|\mathcal{G}, \mu, V \sim HN_{\mathcal{G}}(\mu, V)$, where $V|\mathcal{G} \sim HIW_{\mathcal{G}}(d, D)$ and $\mu|\mathcal{G}, V \sim HN_{\mathcal{G}}(m, \tau V)$. The matrix $\mathbf{X}$ is the $n \times p$ matrix whose i^{th} row is X_i. We showed in Section 6.1.2 that the marginal distribution of $f(\vec{X}|\mathcal{G})$ is a hyper matrix T distribution.

$$\begin{aligned} &f(X_1, \ldots, X_n|\mathcal{G}) \\ &= \prod_{\mathbf{A} \in \mathscr{C} \setminus \mathscr{S}} \pi^{-n|\mathbf{A}|/2} \frac{\Gamma_{|\mathbf{A}|}((d+n+|\mathbf{A}|-1)/2)}{\Gamma_{|\mathbf{A}|}((d+|\mathbf{A}|-1)/2)} \cdot \frac{|Q|^{-|\mathbf{A}|/2}|D_{\mathbf{A}}|^{(d+|\mathbf{A}|-1)/2}}{|D_{\mathbf{A}}+S_{\mathbf{A}}|^{-(d+|\mathbf{A}|+p-1)/2}}, \end{aligned} \quad (7.9)$$

where $Q = I_n + \tau J_{nn}$, and $S_{\mathbf{A}} = (X_{\mathbf{A}} - M_{\mathbf{A}})^T Q^{-1} (X_{\mathbf{A}} - M_{\mathbf{A}})$ is a $p \times p$ matrix.

Given a vector of cluster labels $(\vec{t})$, let $t(i) = \{j : t_j = i\}$ be the set of observations from the i^{th} cluster. We will use $n_i = |t(i)|$ to denote the number of observations from this component. Let $\mathbf{X}^{(i)}$ be the $n_i \times p$ matrix with rows equal to X_j for $j \in t(i)$ and let $\mathbf{M}^{(i)}$ be the $n_i \times p$ whose rows are each $\vec{m}$. We can compute the overall density of the data by using Equation 7.9 within each cluster.

$$f(X_1, \ldots, X_n|\vec{t}, \mathcal{G})$$
$$= \prod_{i=1}^{k} \prod_{\mathbf{A} \in \mathscr{C} \setminus \mathscr{S}} \pi^{-n_i p/2} \frac{\Gamma_p((d+n_i+p-1)/2)}{\Gamma_p((d+p-1)/2)} |Q_i|^{-p/2} |D|^{(d+p-1)/2} |D+S_i|^{-(d+n_i+p-1)/2} \quad (7.10)$$

where $Q^{(i)} = I_{n_i} + \tau J_{n_i n_i}$ and $S_{\mathbf{A}}^{(i)} = (\mathbf{X}_{\mathbf{A}}^{(i)} - \mathbf{M}_{\mathbf{A}}^{(i)})^T (Q^{(i)})^{-1} (\mathbf{X}_{\mathbf{A}}^{(i)} - \mathbf{M}_{\mathbf{A}}^{(i)})$. As noted previously, this is enough to specify Algorithm 7.1.1 and numerically compute $f(\vec{X}|\mathcal{G})$.

7.1.2 Conditional Independence for Contingency Tables

Suppose that we have data that are cell counts from an n-way contingency table. A simple way to model this data would be to specify a multinomial model with a Dirichlet prior for cell probabilities. Alternatively, we could specify a mixture of multinomial models with k components. This would correspond to a latent variable having k levels that defines k clusters for the observations and a separate multinomial model within each cluster. As a further modification, we could model the counts using a Dirichlet mixture of multinomial models. This represents a similar mixture model, but one for which the latent variable has an unknown finite number of levels that may increase as more data are observed. Finally, as we discuss in this section, we may consider a *graphical* Dirichlet mixture. The interpretation is that some decomposable graphical model describes the conditional independence structure within the clusters. In other words, the various marginal values satisfy certain independence relations given the value of the latent variable.

Let $H_{\mathcal{G}}$ be a hyper Markov base measure over the $(p-1)$-dimensional simplex. Suppose $\theta_1, \ldots, \theta_n$ is a sample from the random measure P, where P is a graphical Dirichlet

process with law $DP(\nu H_{\mathcal{G}})$. Let $X_i|\theta_i$ be a multinomial draw with θ_i being the vector of probabilities. This is a graphical Dirichlet process mixture of multinomials, which we express mathematically by

$$\begin{aligned} P|\mathcal{G} &\sim DP(\nu, H_{\mathcal{G}}) \\ \theta_i|P &\sim P \\ x_i|\theta_i &\sim \text{Mult}(\theta_i) \end{aligned}$$

For our base measure $H_{\mathcal{G}}$, we choose to use the hyper Dirichlet law described by Dawid and Lauritzen (1993), which is conjugate to multinomial sampling. As in the Gaussian case, we wish to approximate the marginal data likelihood given $\mathcal{G}$ with all parameters integrated out. Once again, we begin by considering the simpler problem without clustering.

Let $X_1, \ldots, X_n$ be multinomial observations with probability vector θ, where $\theta \sim HD_{\mathcal{G}}(\lambda)$, and λ is an arbitrary table of positive numbers. Given $\mathcal{G}$, it is sufficient to specify the marginal tables $\lambda_{\mathbf{C}}$ for each clique $\mathbf{C} \in \mathscr{C}$. Dawid and Lauritzen show that a sufficient statistic for θ is the set of clique marginal tables, $\{X_{\mathbf{C}}\}$, and that the clique-specific posterior distribution of $\theta_{\mathbf{C}}|\vec{X}$ is $\text{Dir}(\lambda_{\mathbf{C}} + X_{\mathbf{C}})$. The overall law for $\theta|\vec{X}$ is the unique hyper Markov combination of the clique posteriors, which Dawid and Lauritzen (1993) call a hyper Dirichlet distribution. The authors also show that the marginal distribution for $X_{\mathbf{C}}|\mathcal{G}$, with θ integrated out is:

$$f(X_{\mathbf{C}}|\mathcal{G}) \sim \left(\frac{\Gamma(\sum_{i=1}^{p} \lambda_i)}{\Gamma(n + \sum_{i=1}^{p} \lambda_i)}\right) \prod_{i=1}^{p} \left(\frac{\Gamma(\lambda_i + n_{i\mathbf{C}})}{\Gamma(\lambda_i)}\right). \tag{7.11}$$

Returning to the larger setting of graphical Dirichlet mixtures, suppose we have a vector of cluster assignments, which we denote by $\vec{t}$. Within each cluster, the marginal distribution of the data follows Equation 7.11. Hence, we can analytically calculate $f(X|\mathcal{G}, \vec{t})$.

7.1.3 Efficiency of Monte Carlo Integration for 2-Way Tables

Calculations for 2^p-way contingency tables are very efficient if we take advantage of the binary nature of computers. Specifically, computer programmers keep track of several Boolean (i.e.- binary) variables through the use of *flags* and *masks*. Perhaps the best known example of this practice is the file permission flag system. In this system, the ability to execute a file is given a value of 1, whereas write permission is 2 and read permission is 4. These file settings are generally known as flags. It is clear that given a sum between 0 and 7, the values of all three flags are uniquely identified.

In general, we need only n bits to specify n Boolean variables. Thus a single byte stores up to 8 flags. For 2^p-way contingency tables, we simply need to treat each value as "on" or "off", and we can take advantage of this type of flag system. Taking a page out of the computer scientist's book, we treat an observation as a set of p flags, rather than a vector of p distinct values. This is why such tables are incredibly efficient with respect to memory. For the price of a single 4-byte integer, we can specify an observation's cell in a 2^{32}-way table.

Besides being memory efficient, 2^n-way tables are also quite fast due to a method known as *masking*. C++and similarly low-level programming languages have an operator called

the *bitwise-AND* operator (denoted &). Given two bit strings, say 1100 and 1010, the result of the bitwise-AND operation is a third string whose i^{th} bit is 1 if and only if the i^{th} bit of both arguments is 1. Thus, 1100 & 1010 = 1000. This operation turns out to be very useful for calculating a marginal value in a 2^n-way table. For example, suppose an observation occurs in the cell identified by the string 00110001. In other words, the first, fifth and sixth binary variables are "on", while the second, third, fourth, seventh and eighth variables are "off". To find the appropriate cell in some marginal table, we can use a special bit string called a *mask*. For example, consider the marginal table identified by the the 3^{rd} and 5^{th} variables. In this case, we only care about the third and fifth bits, so we use a mask (in this case 00010100) and the bitwise AND operation:

$$00110001 \;\&\; 00010100 = 00010000, \tag{7.12}$$

so we see that the fifth variable is "on" and the third variable is "off". We can see if two observations share a cell in the marginal table by comparing the results of this operation. For example,

$$00110001 \;\&\; 00010100 = 00010000 = 00010001 \;\&\; 00010100, \tag{7.13}$$

so 00110001 and 00010001 share the same cell in the marginal table, but 0011001 and 00010100 are in different cells. Essentially, we have zeroed out all the bits that don't matter. We then compare the resulting value, which depends only on the values of the 3^{rd} and 5^{th} variables. This is exactly what we require for a marginal table comparison.

We could go into much more detail about how the bitwise-AND and other bitwise operations increase the speed our calculations. Suffice it to say that bitwise operations are very efficient in compiled code. Therefore, there is a big advantage to thinking about our data from this unusual (for us statisticians) angle.

7.2 The Mode Oriented Stochastic Search

We now know how to score a graphical model, but we still need a way of searching through the enormous space of decomposable graphs to find the best one - or at least a very good one. The algorithm we chose to implement is the Mode Oriented Stochastic Search (MOSS) presented by Dobra and Massam (2009). The basic premise of this search is to *explore* models by scoring not just the graph, but also all of its neighbors. If a neighbor scores well, relative to the best model found so far, then we add it to the list of models we wish to explore. The overarching goal of the search is to find all graphs whose posterior probability is above a given threshold set relative to the posterior mode. As a practical matter, this strategy requires two threshold values. The lower threshold enables the search to escape local maxima by traveling through nearby bad models as long as they are not too bad. These moderately bad models are saved temporarily, but are discarded after each move with probability q. The MOSS procedure is specified by the next algorithm.

Algorithm 7.2.1 (MOSS(q, c, c^*))**.**

1. *Initialize a starting set of models* $\mathscr{S}$*.*

2. *For each graph* $\mathcal{G} \in \mathscr{S}$*, find the posterior probability* $\mathcal{G}(\vec{X})$ *and mark the graph "unexplored".*

3. *Set* $S \leftarrow \max\{f(\mathcal{G}|\vec{X})|\mathcal{G} \in \mathscr{S}\}$.

4. *While any unexplored models exist:*

   (a) *Let* $\mathscr{L} \subseteq \mathscr{S}$ *be the set of unexplored graphs.*

   (b) *Choose a random graph* $\mathcal{G}$ *from* $\mathscr{L}$ *with probabilities proportional to* $f(\mathcal{G}|\vec{X})$.

   (c) *For each neighbor* $\mathcal{G}^*$ *of* $\mathcal{G}$*:*

      i. *Calculate* $f(\mathcal{G}^*|\vec{X})$.

      ii. *If* $f(\mathcal{G}^*|\vec{X}) \geq c^*S$*, then mark* $\mathcal{G}^*$ *"unexplored" and add it to* $\mathscr{S}$.

      iii. *If* $f(\mathcal{G}^*|\vec{X}) > S$*, then set* $S \leftarrow f(\mathcal{G}^*|\vec{X})$ *and remove from* $\mathscr{S}$ *all models* $\mathcal{G}$ *such that* $f(\mathcal{G}|\vec{X}) \leq c^*S$.

   (d) *With probability* q*, remove from* $\mathscr{S}$ *all* $\mathcal{G}$ *such that* $f(\mathcal{G}|\vec{X}) \leq cS$.

5. *Remove from* $\mathscr{S}$ *all* $\mathcal{G}$ *such that* $f(\mathcal{G}|\vec{X}) \leq cS$.

7.2.1 Comparing Graphs for Equality

As part of the MOSS procedure, we need to be able to test graphs for equality. We do not want to insert a second copy of the same graph into our set of models, and in fact we do not want to waste time scoring the model again. We would like to make these comparisons efficiently, but we are not aware of any simple summary statistics that can uniquely identify a decomposable graph. Fortunately, we can greatly increase the speed of

these comparisons even with a non-unique identifier. What we really require is a statistic that is easily computed, easily compared, and as "unique as possible". Our solution to this problem is to score the edge $i \sim j$ by $(i+1)(j+1)$. We add 1 since the internal representation of nodes runs from 0 to $|\mathbf{V}| - 1$, which implies that $i \cdot j = 0$ for any edge with 0 as an endpoint. The ID code for a graph is simply the total value of all of its edges. For example, the null graph would have an ID code equal to 0, whereas the graph [012][123] would have an ID code of 25. Note that this ID may be slow to compute for larger graphs, but this is of limited importance. Each new graph differs from an old graph by exactly one edge, so we may compute its ID by adding or subtracting a single edge value.

If two graphs have different IDs, then they are obviously different. On the other hand, if two graphs have the same ID, we must resort to a clique-by-clique comparison because equality of IDs does not imply equality of graphs. We believe that the equivalence classes of IDs is quite small, so most inequalities can be decided quickly, but this is an exercise in number theory we have not investigated. At any rate, the non-unique ID method is certainly faster than a clique-wise comparison for every pair of graphs.

7.2.2 Visualizing Results

Recall that MOSS returns all models that score above some cut-off that is set relative to the best graph discovered. It is useful to have a way to summarize this information. We weight the set of models returned by MOSS by their posterior probabilities. The weight for an edge is simply the sum of weights for all graphs that contain the edge. Dobra and Massam (2009) summarize these graphs using the "median model" which is the model

containing all edges that have a weight of 0.5 or higher. Alternatively, the information can be visualized very nicely for graphical models that are not too large. We do so by using a typical graph display, but the color and thickness of each edge is proportional to its weight. For example, an edge with weight 1 is a relatively thick black line. For an edge with weight .5, the edge is a 50% gray-scale line that is half as thick. As the weight approaches 0, the line becomes lighter and thinner. The overall effect is that edges that have low probability are gossamer, but edges with high weight are very vivid. Additionally, we use broken lines for weights under .5 and solid lines for weights that are at least .5. This enables us to quickly read the median model from the weighted graph.

7.3 Simulation Studies for Multinomial Mixtures

We concentrate our analysis on the graphical Dirichlet mixture of multinomials, because this model is much more efficient than the Gaussian mixture model with the aid of bitwise operations. We tested the MOSS procedure for graphical Dirichlet mixtures of multinomial data by examining simulations of various sizes from various mixtures of decomposable graphical models. In this setting, the data form a contingency table of observations. Each X_i is taken to be a vector $(X_{i1}, \ldots, X_{ip})$. For simplicity and efficiency, each X_{ij} was a binary decision. As discussed in Section 7.1.3, we represent X_i by $\sum_{j=1}^{p} X_{ij} 2^{j-1}$, which is easily seen to be an invertible mapping of the vector representation.

Models for our simulation study included a 5-star graph, [01][02][03][04][05], in which the X_i's are independent given X_1; an autocorrelation graph, [01][12][23][34]; and three

other graphs: [01][02][34], [0][12][34], [012][34]. The last three graphs are very similar, and are an interesting test to see how the procedure differentiates them.

Each data set consisted of a mixture of three components. Marginal probabilities varied by component, but were equal for all X_{ij} within a given component. Likewise, the degree of correlation varied by component. For consistency, model parameters were held as constant as possible between the simulations. The general model was:

$$X_{ij} = \alpha_g + \beta_g \sum_{\{k:j\sim k\}} X_{ik}, \tag{7.14}$$

where α_g is the marginal probability for the g^{th} component and β_g controls the degree of correlation between neighbors on the graph. The parameters were set as in Table 7.1, and we generated data sets of size 250, 1250, and 2500.

g	α_g	β_g	$n_g^{(1)}$	$n_g^{(2)}$	$n_g^{(3)}$
1	0.4	0.2	100	500	1000
2	0.5	0.25	80	400	800
3	0.6	0.2	70	350	700

Table 7.1: Model parameters for the first three multinomial mixture simulation studies. $n_g^{(i)}$ is the group size for the i^{th} simulation.

Each of our graphs have $p = 5$ nodes, and hence the data form a contingency table with $2^5 = 32$ cells. The model has only two hyper parameters: $\vec{\lambda}$ and ν. $\vec{\lambda}$ is the parameter of the Dirichlet base measure, and ν is the precision of the Dirichlet process prior over

Dirichlet laws. To signify lack of prior knowledge, we use a flat Dirichlet base measure, $\lambda_i = \lambda/32$ for all 32 cells. Continuing to be non-informative, we choose λ to be small.

Initial analyses showed that for around $\lambda = 1$, spurious edges would appear in the mean graph. This is explained by the clustering property of the Dirichlet process. Setting $\lambda_i = 1/32$ quite naturally corresponds to a prior count of $1/32$ observations in each cell, *per cluster*. To counter this, we proceeded to use $\lambda_i = 1/3200$, so that the prior would not unduly favor more heavily connected graphs.

Figures 7.1 and 7.2 show the median graphs selected by the MOSS procedure. The graphs depicted are for the smallest simulation ($N = 250$). Results did not vary between the different sample sizes. In most cases, the MOSS procedure with a graphical Dirichlet mixture model detects the correct graph. The only problem is for the graph [012][34], which did not find the edge between nodes 1 and 2. The reason for this is unclear.

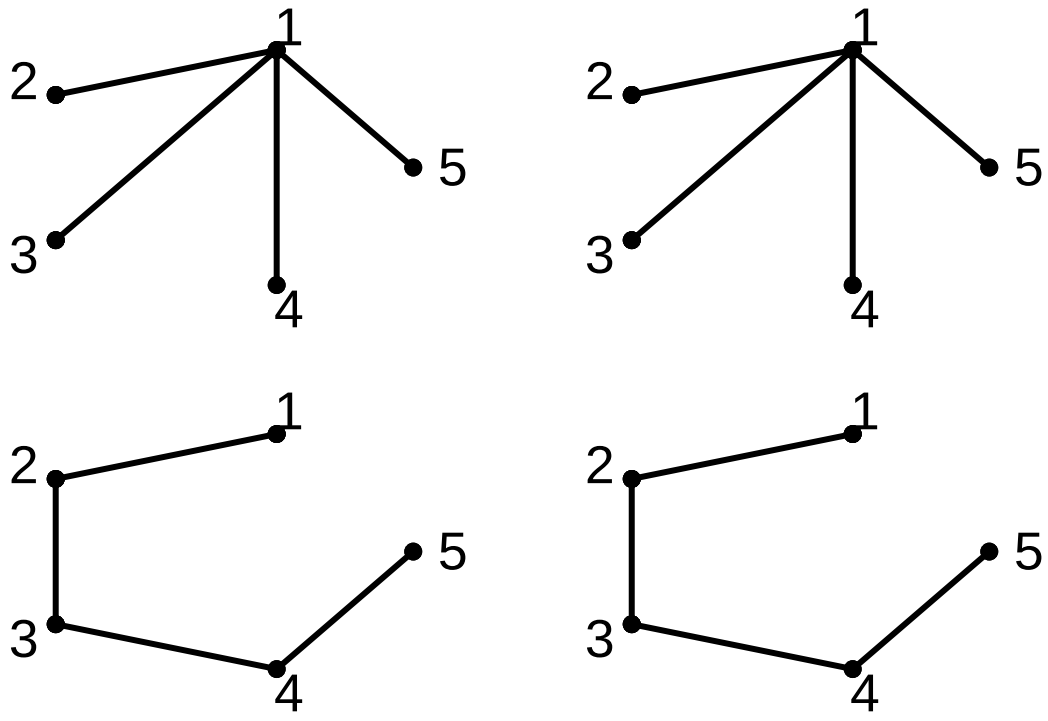

Figure 7.1: Fitted model (left) and true model (right) for star graph (top) and autocorrelation graph (bottom).

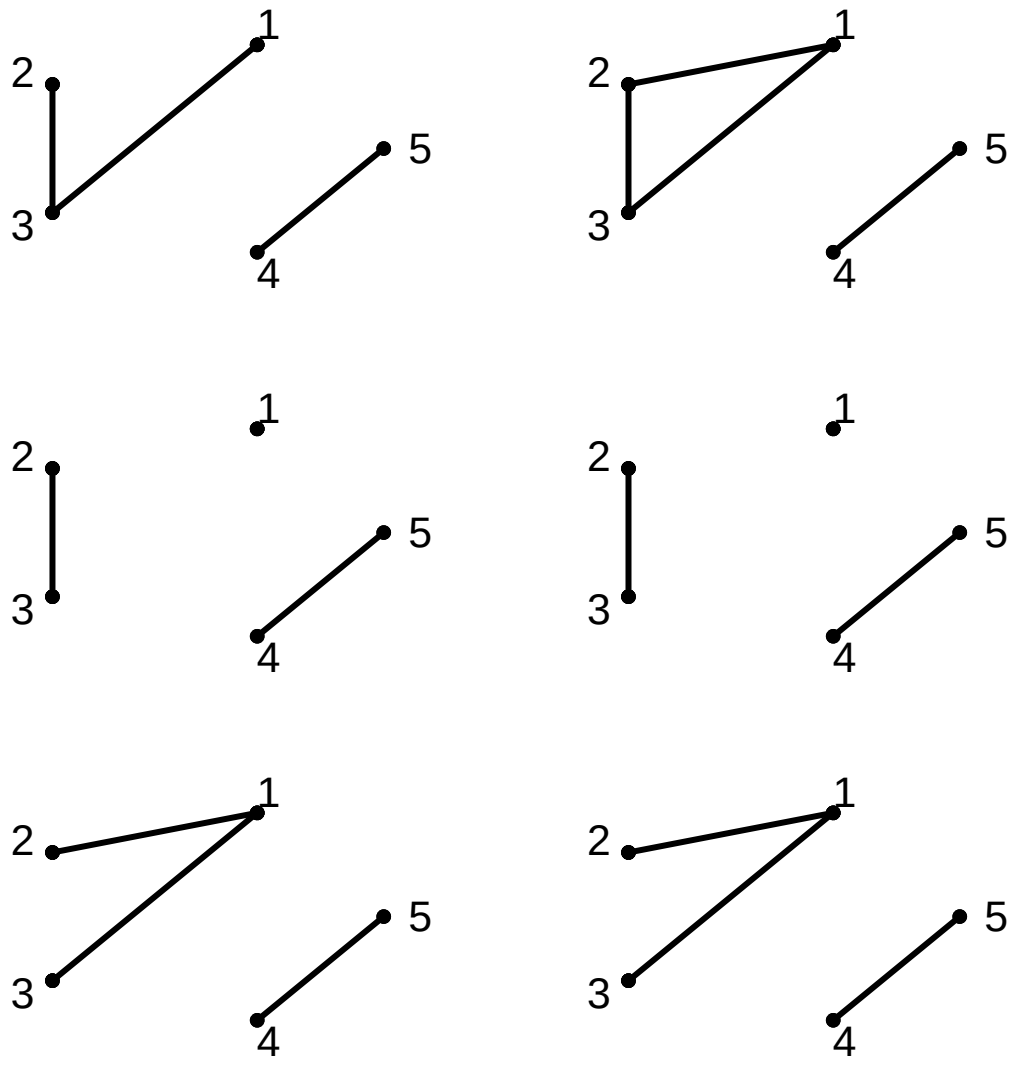

Figure 7.2: Fitted model (left) and truth (right) for graphs [123][45] (top), [1][23][45] (middle), and [12][13][45] (bottom).

7.3.1 Comparison to Non-Mixing Model

We were concerned that if the data arose from a non-mixture distribution, the Dirichlet process might find spurious interactions due to its clustering effect. To resolve this issue, we simulated data from the 5 graphs discussed above, but used parameter values for only one group, which results in non-mixture data. In all cases, the MOSS procedure chose the correct model, indicating that the graphical Dirichlet mixture model works at least as well as the simple Bayesian hierarchical model when the data do not come from a mixture.

We were also curious how well the simple Bayesian model sans Dirichlet process mixing would fit the mixture data from the previous simulations. As it turns out, the MOSS procedure with the simple model also tended to choose the correct graph. A likely explanation is that the various mixture components were not sufficiently different and so did not add or remove interactions when the data are collapsed across the latent variable. It is important to note that with model selection, we are not trying to find a good distribution for the observed data; we seek simply to enumerate the interactions. Because of this, the simple model can determine the correct graph unless collapsing across mixture components induces edges (positively or negatively). It is easy to see that cases do exist. We present one example now.

Suppose that Z is a latent variable taking levels 1 with probability w and 2 with probability $1 - w$. Let X_1, X_2, X_3 be binary random variables such that X_2 and X_3 are conditionally independent and identically distribution given Z, with a distribution specified by the following table:

	$Z=1$	$Z=2$
$\mathbb{P}(X_1=1)$	0.5	0.5
$\mathbb{P}(X_2=1), \mathbb{P}(X_3=1)$	0.6	0.4
$\mathbb{P}(X_2=1\|X_1=1), \mathbb{P}(X_3=1\|X_1=1)$	0.3	0.7
$\mathbb{P}(X_2=1\|X_1=0), \mathbb{P}(X_3=1\|X_1=0)$	0.9	0.1

Summed across both levels of the latent Z,

$$\mathbb{P}(X_3|X_1, X_2) = \frac{.045w + .245(1-w)}{.15w + .35(1-w)}, \tag{7.15}$$

and

$$\mathbb{P}(X_3|X_1) = \frac{.15w + .35(1-w)}{.5}, \tag{7.16}$$

whence we see that $X_2 \not\perp\!\!\!\perp X_3$ in the overall mixture distribution.

We simulated more binary data according to this distribution. The resulting graphs selected by the MOSS procedure are shown in Figure 7.3. In the first set of data, there were 1250 observations in each group, implying $w = .5$. In a second simulation, there were 2000 observations from group 1 and 500 observations from group 2, implying that $w = .8$. As expected, the simple Bayesian model is unable to detect the presence of the latent variable mixing in either case. More interesting is that the graphical Dirichlet mixture worked, but only sometimes. In the balanced data, the MOSS procedure with graphical Dirichlet mixing detected that no three-way interaction term is present, but the wrong

two-way interactions are concluded. This is probably because the Monte Carlo approach randomizes cluster assignments based on the prior distribution. As such, a given cluster is actually slightly more likely to contain observations from both groups instead of only one group. In other words, a given cluster may not be fit well by the true graph because the clusters may be composed of data from multiple component distributions. Thus, there is a need for an improved method for assigning clusters. A possibility is to randomly generate the number of clusters based on the distribution presented by Antoniak (1974), then choose the cluster assignments conditional on this and the observed data. When the group sizes are imbalanced, then the graphical Dirichlet mixture does quite well. Naturally, in the imbalanced case, the probability that a cluster contains a majority of points from a single component distribution is increased.

Admittedly, there is room to improve the model selection, but we have seen that the graphical Dirichlet mixing does at least as well as the simpler Bayesian model, in some cases even outperforming it. When the data are not from a mixture, then the graphical Dirichlet model does as well as the simpler Bayesian model. For certain mixtures, if the components just happen to exhibit the same conditional independence structure in the overall distribution as in the cluster distributions, then both procedures are again comparable. On the other hand, there are definitely mixture models that the simpler Bayesian model cannot possibly determine. In these cases, the graphical Dirichlet mixture performed better when the component distributions had unequal probabilities.

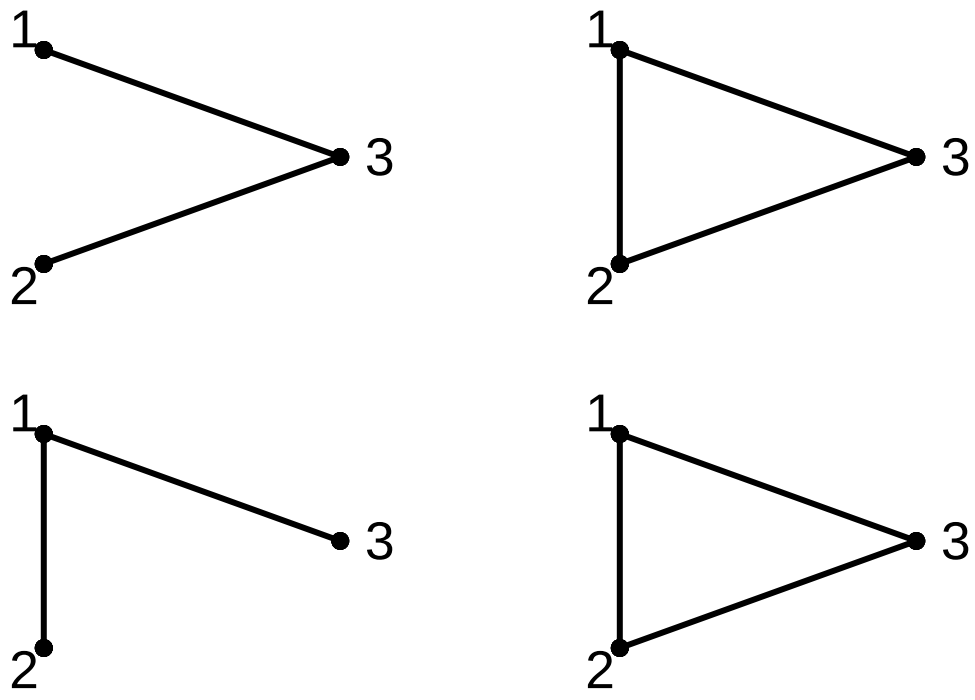

Figure 7.3: Fitted models using graphical Dirichlet mixing (left) and a single Dirichlet-Multinomial law (right) for the graphs [12][13]. The top row is from a simulation using balanced group sizes. The bottom row shows results for imbalanced groups. The correct graph is the bottom-left, indicating that the graphical Dirichlet mixture succeeds for imbalanced groups and the simple model never succeeds.

7.4 Czech Autoworkers Data

As an application of this method to real world data, we analyzed the Czech autoworkers data that Dobra and Massam (2009) used when they presented the MOSS procedure. Edwards and Havranek (1985) first examined this data, which classifies $N = 1841$ workers from a Czechoslovakian car factory by the presence or absence of $p = 6$ risk factors for coronary thrombosis. We enumerate these 6 risk factors as follows: (1) the worker smokes, (2) strenuous mental work, (3) strenuous physical work, (4) systolic blood pressure ≥ 140, (5) ratio of β to α lipoproteins ≥ 3, and (6) family history of coronary heart disease. In all cases, we treat the presence of a risk factor as 1, and the absence by 0. As mentioned in Section 7.1.3, this 2^6-way table is incredibly memory efficient, requiring only 1841 bytes for the entire contingency table without resorting to any special data types. Additionally we can use bit shift operators and bitwise comparisons to efficiently compute marginal tables on demand.

We ran the MOSS algorithm multiple times, varying λ in $\{.01, .1, 1, 10\}$. These correspond to a prior count of $1/3200, 1/320, 1/32, 1/3.2$ observations in each cell per component. We chose the same MOSS settings as Dobra and Massam (2009): $c = .1,\ c' = .001,\ q = .1$. Table 7.3 shows the resulting model selection and their relative probabilities. Also included are the median graphs, which contain all edges with a posterior weight of 0.5 or more. These tables are summarized graphically in Figure 7.4. Our results echo those of Dobra and Massam: as λ increases, the posterior favors higher-level interaction terms and places more weight on already present interactions.

It is interesting to compare the results of the MOSS procedure with the graphical Dirichlet mixing versus those of Dobra and Massam (2009), who used a non-mixed Dirichlet-multinomial model. Essentially, the two models are the same, except that the mixture model allows for the possibility of clustering. Specifically, if the posterior graphs differ, it suggests the presence of a latent variable which explains some of the relationships among the measured variables. If the graphs are similar, then it suggests no such latent variable exists.

Figure 7.5 shows the posterior graph inference for both types of models for $\lambda \in \{1, 2, 3\}$. Notably, the two median models (solid lines) agree in all cases, except that for $\lambda = 1$, the Dirichlet mixture model gives the edge $(1, 4)$ a weight just less than .5. In most cases, the two models even approximately agree about the relative weighting of the edges. In general, because the two processes agree so well, it does not seem like there are any important latent variables present which influence the conclusions about the independence structure. We emphasize that this conclusion cannot be reached using the simpler model alone. While the graphical Dirichlet process did not change our conclusions, it did provide a way to verify them.

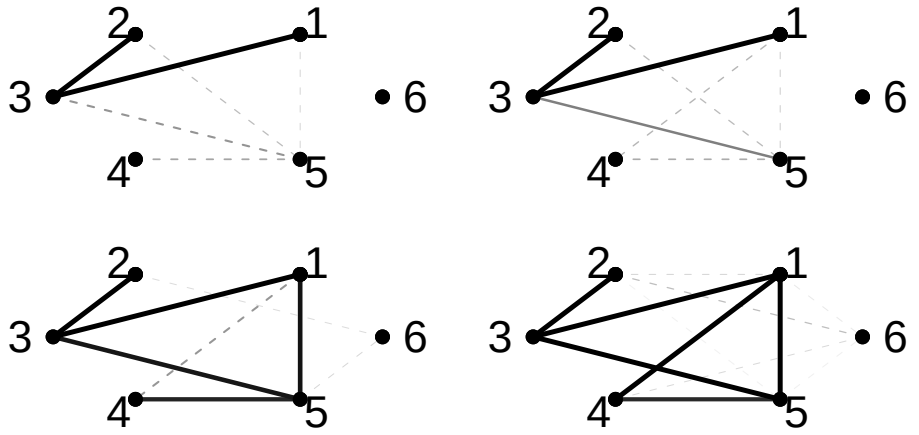

Figure 7.4: Visual representation of the median graphs for the Czech auto workers data set. Thicker darker lines indicate higher posterior probability of an edge. Dashed lines indicate that the edge has weight $< .5$ and is not in the median graph. From left to right and top to bottom: $\lambda = .01$, $\lambda = .1$, $\lambda = 1$, $\lambda = 10$

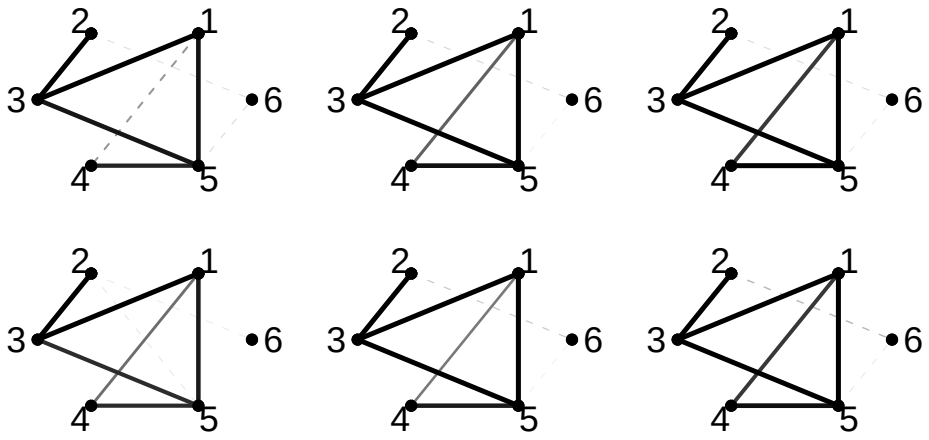

Figure 7.5: Comparison of posterior edge weights for MOSS procedure using graphical Dirichlet mixtures (top) and the simple Dirichlet-multinomial model (bottom). From left to right, columns correspond to $\lambda = 1,\ 2,\ 3$.

$\lambda = .01$		$\lambda = .1$		$\lambda = 10$	
model	P	model	P	model	P
[13][23][45][6]	.279	[13][14][23][35][6]	.229	[135][145][23][6]	.539
[13][23][25][45][6]	.215	[13][23][35][4][6]	.174	[135][145][16][23]	.174
[13][23][4][5][6]	.164	[13][23][25][45][6]	.131	[135][145][23][26]	.129
[13][23][35][4][6]	.140	[13][23][25][4][6]	.108	[135][145][23][6]	.104
[13][23][15][4][6]	.127	[13][23][45][6]	.070	[135][145][23][56]	.054
[13][23][25][4][6]	.044	[13][15][23][45][6]	.069		
[13][23][35][45][6]	.030	[13][15][23][4][5]	.063		
		[13][23][35][45][6]	.062		
		[135][23][4][6]	.037		
		[13][14][15][23][6]	.030		
		[13][14][23][25][6]	.027		
[13][23][4][5][6]	(med)	[13][23][35][4][6]	(med)	[135][145][23][6]	(med)

Table 7.2: MOSS output - models with high posterior probability for the autoworkers data set for $\lambda \in \{.01, 1, 10\}$. P is the relative posterior probability within each set.

$\lambda = 1$		$\lambda = 2$		$\lambda = 3$	
model	P	model	P	model	P
[135][23][45][6]	.334	[135][145][23][6]	.410	[135][145][23][6]	.538
[135][145][23][56]	.155	[135][23][45][6]	.286	[135][23][45][6]	.223
[135][145][23][6]	.116	[135][23][45][26]	.094	[135][145][23][26]	.151
[135][14][23][6]	.104	[135][145][23][26]	.091	[135][145][23][56]	.087
[135][23][45][26]	.101	[135][14][23][6]	.062		
[13][15][23][45][6]	.085	[135][145][23][56]	.057		
[13][23][35][45][6]	.070				
[135][14][23][26]	.035				
[135][145][23][6]	(med)	[135][145][23][6]	(med)	[13][145][23][6]	(med)

Table 7.3: MOSS output - models with high posterior probability for the autoworkers data set for $\lambda \in \{1, 2, 3\}$. P is the relative posterior probability within each set.

Chapter 8

Further Work and Extensions

The main purpose of this dissertation is to extend knowledge about the Dirichlet process to incorporate hyper Markov priors. Key results included determining when a hyper Dirichlet process is also a simple Dirichlet process. To put it conversely, when is the distribution of a Dirichlet process, as defined by Ferguson (1973), a hyper Markov prior law? In Theorem 4.2.7, we provided the necessary and sufficient conditions for this property. While these conditions were shown independently by Asci et al. (2006), we provide a second proof, as well as advance in the resulting theory. On the theoretical front, Section 4.5 explores the noisy observations of a random sample of a Dirichlet process, and the resulting *hyper Dirichlet process mixture*. In this section, we explore what conditional independence structures persist in various marginal and conditional distributions for the random Dirichlet process measure P, a random sample $\theta_1, \ldots, \theta_n \sim P$, and noisy observations $X_i \sim F(\cdot; \theta_i)$. The key result is Theorem 4.5.3, which states that if we condition on cluster memberships,

then the distributions of θ_i and X_i are respectively a hyper Markov prior law and a Markov measure. Most importantly, this result does not require our "hyper" Dirichlet process to actually be hyper Markov! Instead, we are guaranteed the same properties if we use a (non-hyper) Dirichlet process as long as its base measure is hyper Markov. We call such a Dirichlet process a *graphical Dirichlet process* to avoid confusion with the usual modifier "hyper". In fact, since the graphical Dirichlet process is by definition a Dirichlet process, a lot of current knowledge, and algorithms translate directly. For this reason, it is safe to say that we have accomplished our goal of joining the fields of graphical models and non-parametric inference. In general, once we account for the clustering effect of the Dirichlet process, we can apply the multitude of theorems that Dawid and Lauritzen (1993) and others have presented for hyper Markov laws. In particular, if the base measure is *strong* hyper Markov, the marginal distribution for each cluster of observations is Markov, and posterior calculations can be computed locally by clique and component.

Several open questions remain about the current work, which also provide opportunities for further study. An important problem is the issue of clustering for model selection as discussed in Chapter 7. We calculated the likelihood of the data by generating random table assignments from the Chinese restaurant process (i.e.- the prior distribution). Because we used the prior distribution, clustering was not done "smartly" by taking values near each other. More work is needed in this area before the hyper Dirichlet process reaches it full potential for model selection. At the very least, we have seen that it works when the collapsed data are close to a non-mixture distribution, and that it also works for mixtures in which the components are unequally weighted, which is the benefit to using

graphical Dirichlet mixtures rather than simpler Bayesian models. Our approach to the Czech autoworkers data was to use both types of models. This allowed us to determine that there were no important latent variables that helped explain the independence structure in the data.

Another important issue is that of consistency. For Gaussian mixtures, Escobar and West (1995) show that their univariate Gibbs sampler converges to the correct posterior distribution. The same logic applies in the multivariate setting. Of course, we also want the posterior distribution of the Dirichlet mixture to converge to the correct distribution! Ghosal et al. (1999) provide some conditions for weak and L_1 convergence for location-scale mixtures of Normals. These results are a good place to begin to show when convergence occurs in the hyper Normal case.

Consistency is also a concern when using graphical Dirichlet processes for model selection. That is to say, we would like to show that the posterior probability of the correct graph converges toward one as the sample size increases. It is beneficial to know when this consistency occurs and how many observations are needed for the posterior probability of the correct graph to be greater than, say $1 - \epsilon$, with large probability. A related question is what kind of errors can we expect when the sample size is smaller. This question is particularly important, because convergence is often slow or impossible with latent variable models. For example, Fienberg et al. (2010) show that the posterior distribution in latent variable models is typically multimodal. In these cases, finding the global posterior maximum is particularly hard.

In Section 6.4, we saw that non-parametric mixture models using the Dirichlet processes are similar in effect to local bandwidth selection for kernel density estimators. As such, it would be enlightening to compare Dirichlet mixtures to other methods for local bandwidth selection. We note that using cross-validation to choose local bandwidths for each dimension is complex for even moderate dimensions. A simpler method is to choose whatever bandwidth includes the k nearest neighbors for each point and use cross-validation to choose k. Further investigation is required to understand how these two methods compare to Dirichlet process mixtures in terms of both results and computational complexity.

In Chapter 7, we used graphical Dirichlet mixtures and the MOSS procedure (Dobra and Massam, 2009) to select a conditional independence model. We compared our results to those of Dobra and Massam who used MOSS with non-mixture models, but there are a variety of other methods we could use for comparison. These include other stochastic searches that could work with graphical Dirichlet mixtures or non-mixture models such as the shotgun stochastic search of Jones et al. (2005) and Hans et al. (2007) or the MC^3 algorithm of Madigan and York (1995). There are also a variety of methods including the PC algorithm of Spirtes et al. (2001) and Kalisch and Buhlman (2007); and the Min-Max Hill-Climbing method of Tsamardinos et al. (2006). Tsamardinos et al. provide a list of many other model selection methods for comparison.

Clearly, the next step is to expand this work to more observations in larger dimensions. The issue in these more complex settings is two-fold. As a practical matter, we need to investigate how computation time scales as the sample size and dimension increase. In the Gaussian mixtures, it is well-known that the computational complexity of inverting

a $p \times p$ matrix is on the order of p^3. Fortunately, we have already seen in Chapter 6 that the hyper Normal distribution is not so severe because we can take advantage of the conditional independence structure. To recall, we can invert the submatrix pertaining to each clique and separator individually. Nonetheless, it is possible that the straightforward Gibbs sampling algorithm may not be useful in these settings. In this case, we may need to turn to variational inference or other approximations to reduce the complexity of the problem.

The second issue in large-dimensional settings is overcoming sparsity, the problem of having little to no information for much of $\mathcal{X}$, the space of observations. In the case of contingency tables, this corresponds to having many empty cells with no observations. How well does the model selection procedure of Chapter 7 work when various marginal tables place all weight into one cell? To answer this question, we need to investigate some more complex data sets. For example, Dobra and Massam (2009) cite two studies that provide 2^8- and 2^{16}-way contingency tables. The first is a study of households in Rochdale that concerns eight binary variables related to women's economic activity and their husbands' unemployment. The sample of 665 observations results in zeros for 165 of the possible 256 cells. The second study is the National Long-Term Care Survey, whose $21,574$ observations result in $62,384$ zeros, which is 95.19% of the available cells in a 2^{16}-way table.

We should also like to extend our applications to non-decomposable graphs. Once again, we rely on Theorem 4.5.3. Liu and Massam (2006) provide a general conjugate prior for hierarchical log-linear models. We need only use this class of measures as our Dirichlet process base measure. Once we account for cluster memberships, the distribution within

each component is the typical Bayesian model, $\theta_i^* \sim H$ and $X_1, \ldots, X_{n_i} | \theta_i^* \sim F(\cdot; \theta_i^*)$. Therefore, posterior and marginal calculations proceed as in the simple non-clustered case, but repeated for each component. Even though we can extend the theory in this way to incorporate non-decomposable graphs, implementation is still a concern. This is because θ_i^* can not be marginalized out of the posterior laws of Liu and Massam unless the graph is decomposable. Dobra and Massam (2009) provide methods for numerical calculations, but the computational complexity is a concern.

As we have noted throughout this work, the graphical Dirichlet process is convenient because it is a special case of the Dirichlet process, so it allows immediately use of existing theory. For example, we discuss Dirichlet mixtures of Gaussians in Chapter 6. Another application that we could explore is the infinite hidden Markov models of Beal et al. (2002). In these models, the Dirichlet process enables us to consider a hidden Markov model in which the number of hidden states is finite, but unknown. Similar to Dirichlet process mixtures, the number of hidden state spaces may grow as more data are observed.

Another way to extend the theory that of Chapter 4 is to consider what other non-parametric processes may have interesting hyper Markov generalizations. We saw immediately that Theorem 4.5.3 applies to all stick-breaking processes as long as the base measure is hyper Markov. Therefore, our work immediately allows us to incorporate any current or future application of a stick-breaking process as long as we are willing to condition on the cluster memberships as in the Gibbs sampling procedures of MacEachern (1994) and Ishwaran and James (2001).

Hyper Markov versions of some of the extended Dirichlet process family seem rather straightforward. This is again thanks to Theorem 4.5.3. For example, the hierarchical Dirichlet process can be used in graphical settings by using a graphical Dirichlet process at the highest level. Thus, any draws from this process will be hyper Markov distributions, which will trickle down through the various layers in the hierarchy. At the bottommost layer, we will again have a graphical Dirichlet process, so we may again apply Theorem 4.5.3.

Particularly exciting is the idea of a hyper Markov Beta process. It would not be so simple to directly apply my work to describe a hyper Markov version of a Beta process, but the interpretation is extremely interesting. Although the Beta process has a stick-breaking representation (Teh et al., 2007), it is *not* a stick-breaking process. This is easily seen since the random weights in the Beta process do not sum to one. The reason behind this is that a given observation can be a combination of several atoms, and thus the random atoms are no longer mutually exclusive. This process leads to fuzzy clustering or feature selection, as opposed to the hard clustering of the Dirichlet process. Herein lies the interesting interpretation of a hyper Beta process: the various features can be selected according to some independence model (in theory.) To extend the example of Teh et al. regarding films, are the features "action" and "comedy" independent given that the movie "stars Jackie Chan"? This type of question has applications for trying to understand customer preferences, which is certainly a hot topic at present. Other examples of soft clustering include mixed-membership models (Erosheva and Fienberg, 2005) and the related latent Dirichlet allocation model of Blei et al. (2003).

A major extension of this work is to allow for the graphical models to vary between the components. In this case, it would no longer make sense to use the type of hyper or graphical Dirichlet processes we have discussed in this dissertation. These models constrain the base measure to be hyper Markov with respect to some graph, which then becomes the graphical model within each component. There are a few potential candidates to replace this idea in the non-identical graph situation. One idea is to use a base measure for the full graphical model that gives positive mass to some proper subspaces. This would lead to a positive probability of choosing a graph besides the saturated graph, and hence the graphical model would be permitted to differ between components. Another possibility is to put a prior over the space of graphs, with a Dirichlet process for each permissible model. An interesting twist on this idea would be to use a model similar to the hierarchical model which could share information between the various graphs. For example, if a certain interaction term is present, we may be able to estimate it by combining information from all graphs which contain the appropriate edge.

Generally speaking, there are a variety of interesting ways to extend the work presented here.

Appendix A

Matrix Algebra Proof

Lemma A.0.1. *For real numbers, $x_1, \ldots, x_n$, set $s = \sum_{i=1}^n x_i$ and define the matrices*

$$A_n(x_1 \ldots x_n) = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \frac{s-x_1}{s^2} & \frac{-x_1}{s^2} & \ldots & \frac{-x_1}{s^2} & \frac{-x_1}{s^2} \\ \frac{-x_2}{s^2} & \frac{s-x_2}{s^2} & \cdots & \frac{-x_2}{s^2} & \frac{-x_2}{s^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-x_{n-1}}{s^2} & \frac{-x_{n-1}}{s^2} & \ldots & \frac{s-x_{n-1}}{s^2} & \frac{-x_{n-1}}{s^2} \end{pmatrix} \tag{A.0.1}$$

$$B_{n-1}(x_1 \ldots x_n) = \frac{1}{s^2} \begin{pmatrix} s - x_1 & -x_1 \cdots & -x_1 \\ -x_2 & s - x_2 \cdots & -x_2 \\ -x_{n-1} & -x_{n-1} & \cdots s - x_{n-1} \end{pmatrix}. \tag{A.0.2}$$

These matrices have the following determinants

$$|A_n(x_1 \cdots x_n)| = \frac{(-1)^{n-1}}{s^{n-1}} \tag{A.0.3}$$

$$|B_{n-1}(x_1 \cdots x_n)| = \frac{x_n}{s^n} \tag{A.0.4}$$

We first make two remarks.

Remark A.0.1. *Let $a = (a_1, \ldots, a_d)$ and $b = (b_1, \ldots, b_n)$ be in $\mathbb{R}^d$ and consider the $d \times d$ matrix $C(a, b)$, where $c_{ij} = a_i b_j$. For any $\lambda \in \mathbb{R}$, $|\lambda I_d - C(a, b)| = \lambda^{d-1}(\lambda - \sum_{i=1}^d a_i b_i)$. This is true because the columns of $C(a, b)$ are scalar multiples of each other, meaning that the matrix has rank 0 or 1. Therefore, $C(a, b)$ has 0 as an eigenvalue of multiplicity d or $d-1$. Furthermore, a^T is an eigenvector of $C(a, b)$ with eigenvalue $\sum_{i=1}^d a_i b_i$. This implies that the characteristic polynomial of $C(a, b)$ is $\lambda^{d-1}(\lambda - \sum_{i=1}^d a_i b_i)$.*

Remark A.0.2. *If A and D are square matrices of order p and q such that D^{-1} exists, then*

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I_p & BD^{-1} \\ 0 & I_q \end{pmatrix} \begin{pmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} I_p & 0 \\ D^{-1}C & I_q \end{pmatrix}, \tag{A.0.5}$$

so that

$$\left| \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right| = |D| \cdot \left| A - BD^{-1}C \right|. \tag{A.0.6}$$

Proof of Lemma A.0.1. Consider two vectors $a = (x_1, \ldots, x_{n-1})$ and $b = (1, \ldots, 1)$ in $\mathbb{R}^{d-1}$. We note that $B_{n-1}(x_1, \ldots, x_n) = s^{-2}(sI_{n-1} - C(a, b)$. We apply Remark A.0.1 with $d = n - 1$ and λ to obtain

$$|B_{n-1}(x_1, \ldots, x_n)| = \left| s^{-2(n-1)}(sI_{n-1} - C(a, b) \right| = \frac{1}{s^{2n-2}} s^{n-2} \left(s - \sum_{i=1}^{n-1} x_i \right) = \frac{x_n}{s^n}. \tag{A.0.7}$$

For the determinant of $A_n(x_1, \ldots, x_n)$, we consider

$$A_n^* = \begin{pmatrix} s - x_1 & -x_1 & \ldots & -x_1 & -x_1 \\ -x_2 & s - x_2 & \ldots & -x_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_{n-1} & -x_{n-1} & \ldots & -x_{n-1} & -x_{n-1} \\ s^2 & s^2 & \ldots & s^2 & s^2 \end{pmatrix}, \tag{A.0.8}$$

which is obtained by a circular permutation of the rows of $s^2 A_n(x_1, \ldots, x_n)$. Since the signature of such a permutation is $(-1)^{n-1}$, we have $|A_n(x_1, \ldots, x_n)| = s^{-2n}(-1)^{n-1} |A_n^*|$. Note that we can write A_n^* as the block matrix

$$A_n^* = \begin{pmatrix} s^2 B_{n-1}(x_1, \ldots, x_n) & -a^T \\ s^2 b & s^2 \end{pmatrix}. \tag{A.0.9}$$

We apply Remark A.0.5 with $p = n - 1$, $q = 1$, $A = s^2 B_{n-1}(x_1, \ldots, x_n) = sI_{n-1} - C(a, b)$, $B = -a^T$, $C = s^2 b$, and $D = s^2$. Note that $BD^{-1}C = -C(a, b)$ and $A - BD^{-1}C = sI_{n-1}$. Hence, $|A_n^*| = (-s)^{n+1}$, which implies that $|A_n(x_1, \ldots, x_n)| = (-1)^{n-1}/s^{n-1}$. □

Appendix B

Description of Novel Application Programs

B.1 Gibbs Sampler for Dirichlet Mixtures

The Gibbs sampling program is designed to be extended to other mixture models via class inheritance. New models require a class for reading and accessing hyper parameters and options that inherit from the Base classes. In addition, the functions must be defined to calculate the density of an observation for an existing component (i.e.- cluster), calculate the marginal density of an observation to weight the probability of a instating a new component, and optionally update hyper parameters. All code is available upon request.

B.2 MOSS Procedure for Model Selection

The implementation of the MOSS Procedure is general and not restricted to any specific family of graphical models. Adaptation to a new model requires a function for scoring models, locally scoring models based on neighboring graphs, reading in an appropriately

gibbs.cpp	Main Program
BaseHyperPar.cc	Virtual class for model hyper parameters
BaseOptions.cc	Virtual class for storing and accessing options
Data.cc	Simple class to read in data matrix without knowing size
Exception.h	Exception handling class
gauss.cc	Density calculations for hyper Normal and hyper T distribution random variable generation for hyper Normal and hyper Inverse-Wishart
GaussComponent.cc	Class containing covariance matrix, mean vector, cluster members, etc.
GaussHyperPar.cc	Hyper parameter class for Gaussian mixture models
GaussOptions.cc	Options class for Gaussian mixture models
GenMat.cc	General Matrix class
gibbs_gauss.cc	Routines specific to Gaussian mixtures (density calculation, etc.)
Graph.cc	Graph class using algorithms in Chapter 5
matrix.cc	Matrix calculations and partial CBLAS interface (necessary for Gaussian model calculations)
Observation.cc	Class for a single vector-valued observation or parameter
SpdMat.cc	Symmetric Positive Definite Matrix class
TriMat.cc	Triangular Matrix class

Table B.1: Program components for Gibbs sampler

formatted data file, and reading in options and/or parameters. All code is available upon request.

moss.cpp	Main Program
DirMix.cc	Function for randomizing the Chinese restaurant process
Exception.h	Exception handling class
GenMat.cc	General Matrix class
Graph.cc	Graph class using algorithms in Chapter 5
matrix.cc	Matrix calculations and partial CBLAS interface (necessary for Gaussian model calculations)
Model.cc	Extended Graph class to calculate an ID (Section 7.2.1) and track posterior probability
modelDir.cc	Density calculations for multinomial mixtures
modelGauss.cc	Density calculations for Gaussian mixtures
ModelSet.cc	Container class for Models that allows easy sorting, inserting, and removing models
SpdMat.cc	Symmetric Positive Definite Matrix class
TriMat.cc	Triangular Matrix class

Table B.2: Program components for model selection with MOSS

Appendix C

Gaussian Mixture Data

C.1 First Simulation (2 Groups, N = 80)

C.1.1 Group 1

-0.96252	-1.7709
-0.3757681	0.279161
-0.649096	0.6896103
-0.7577688	0.6744051
-0.854872	0.3806457
-0.8463264	-0.06467736
-0.856552	0.1632653
-0.9185322	0.3274943
-1.125838	-1.175224
-1.023435	0.2870387
-0.5561287	2.24312
-0.6087618	-0.509932
-0.7675085	0.9609931
-0.7982847	-1.265946
-0.6955114	0.620941
-0.7281424	-2.062863
-0.7789618	-0.6583404
-0.8837022	-1.048387
-0.898037	-0.3055856
-0.7096705	-0.6438723
-0.6548463	2.40493
-1.045544	0.1936726
-0.8533627	0.7614752
-0.6823466	1.392846
-0.8215265	-0.5845036

-0.5219728	0.8755979
-1.025816	1.490722
-0.562838	-0.3390962
-0.8132794	0.9939994
-0.6065489	-0.8239922
-0.6529929	-1.767877
-0.6262193	-0.1626263
-0.8587752	0.1232434
-0.4556338	1.033686
-0.6710562	-1.078418
-0.6966584	-0.1826885
-1.232063	0.8179208
-0.3252679	-0.04129751
-0.8384499	-1.353489
-0.7214294	-0.4798759
-0.6676924	-0.9843735
-0.9075145	-0.4347082
-0.8740164	1.287443
-0.2982996	-1.639237
-0.2526371	-0.5308874
-0.9344895	0.4519744
-0.6680409	0.6427215
-1.048147	-0.720952
-0.501346	-0.682268
-0.9206802	-0.483624

C.1.2 Group 2

1.59758	1.125016
1.257029	-0.9700945
1.118404	0.1892425
1.262351	-1.25831
0.9114809	-1.124827
1.469418	2.908482
1.346539	1.717096
1.439406	0.7294625
1.471952	-0.637986
1.377825	-0.3210632
1.466749	-0.6646276
0.8195599	0.8443221
1.292487	-0.6839993
0.9607036	0.8171132
1.592922	0.005989548

1.362352	-0.4750972
1.370255	-0.7129749
1.394153	1.002271
1.351645	0.100199
1.573454	-0.327239
1.377118	-0.2839906
1.074378	1.945562
1.087292	0.3537159
0.7836717	-0.4752664
1.157849	-0.1229258
1.258187	-0.8128672
1.439963	-1.233051
1.075225	1.030824
1.162404	0.3388
0.682555	-0.3050413

C.2 Second Simulation (3 Groups, N=120)

-1.272848	-1.272156
-0.5652529	-0.5636246
-0.8948735	-0.4217673
-1.025928	-0.4270224
-1.14303	-0.52855
-1.132724	-0.6824602
-1.145056	-0.6036798
-1.219801	-0.5469199
-1.469802	-1.066282
-1.346309	-0.560902
-0.7827593	0.1151486
-0.8462324	-0.8363467
-1.037674	-0.3279734
-1.074788	-1.097636
-0.9508483	-0.4455004
-0.9901998	-1.373062
-1.051486	-0.8876389
-1.177798	-1.022445
-1.195085	-0.7657216
-0.9679236	-0.8826385
-0.9018082	0.1710726
-1.372971	-0.5931706
-1.14121	-0.3969297
-0.9349723	-0.1787185

-1.102817	-0.8621198
-0.7415689	-0.3574872
-1.34918	-0.1448912
-0.7908505	-0.7773034
-1.092871	-0.3165659
-0.8435638	-0.9448906
-0.899573	-1.271111
-0.8672853	-0.7163128
-1.147737	-0.617512
-0.6615672	-0.3028495
-0.9213566	-1.032824
-0.9522316	-0.7232466
-1.597905	-0.3774213
-0.504352	-0.6743798
-1.123226	-1.127892
-0.9821042	-0.8259589
-0.9172999	-1.000321
-1.206514	-0.8103483
-1.166117	-0.2151474
-0.4718295	-1.226651
-0.4167627	-0.8435892
-1.239045	-0.5038977
-0.9177202	-0.4379727
-1.37611	-0.9092784
-0.716694	-0.8959086
-1.222391	-0.8272543

C.2.1 Group 2

1.814513	-0.2712846
1.403824	-0.9953857
1.236649	-0.5947017
1.410242	-1.094997
0.9871092	-1.048863
1.659955	0.3451074
1.51177	-0.06665302
1.623762	-0.4079938
1.663012	-0.8806041
1.549498	-0.7710709
1.656737	-0.8898118
0.8762568	-0.3682966
1.446585	-0.896507
1.04647	-0.3777004

1.808895	-0.6580367
1.530839	-0.8243073
1.540369	-0.9065214
1.569189	-0.3137073
1.517927	-0.6254765
1.785417	-0.7732053
1.548646	-0.758258
1.183556	0.01230825
1.199129	-0.5378573
0.8329773	-0.8243658
1.284218	-0.7025917
1.40522	-0.9410456
1.624433	-1.086267
1.184577	-0.3038388
1.28971	-0.5430125
0.7110353	-0.7655335

C.2.2 Group 3

0.005056816	0.9752276
0.1346278	1.749739
0.4183162	0.9987685
0.4441468	0.6102641
0.3264747	1.558365
-0.06595759	1.021703
0.2773774	1.228761
0.3588781	1.383504
0.155543	1.751748
0.05294463	1.147564
0.3487071	1.212559
0.5110467	0.9623233
-0.1158666	1.233439
-0.03860848	1.325969
0.1740918	0.7374434
0.09227962	1.573248
0.1492266	2.057101
0.2850054	1.985774
0.3942652	1.676832
0.4733483	1.843865
0.3118394	1.300251
0.6066388	1.511629
0.2229918	1.526994
0.1802363	1.247405

-0.01730635	1.726108
0.2289823	1.05661
0.2007447	0.7193761
0.273882	1.641478
0.2075747	1.320678
0.3580804	1.330512
0.09909286	1.182937
0.3605706	0.7130042
0.3673858	0.6312578
0.1269241	1.360492
0.3882338	1.507009
0.2109494	1.228037
0.1192166	1.466044
0.2059637	1.488257
0.1415698	1.465119
-0.006944648	1.351143

C.3 Third Simulation (3 Groups, N=120)

C.3.1 Group 1

-1.07232632698244	0.365416568060968
-0.68146117103353	-0.661843740806224
1.37044977155614	-0.70923945226658
-1.25587260372295	-0.807957535366028
0.223000661414888	1.47672922816427
-1.14583705419235	-1.68561655314086
0.277012945182815	1.03216173499748
1.27012635229262	0.0969505576368773
-1.16868726005606	0.0882601975216972
0.181748240545161	1.69844487476056
-1.3221529914788	-0.531569195419415
1.29747211319929	-0.916326722388026
0.473949943917318	0.573605631877731
-0.494562786985223	-0.355244791700483
1.40919714147866	-0.390828050707983
1.60995346396902	-0.507447045263485
0.176286888806879	1.41040121599526
-1.31722297258992	-0.620348468341057
1.26782487439168	-0.109854961528088
0.246991447135737	1.78575152089344
0.514137733932112	1.80560093127146
-1.23932547989891	-1.8427657256247

0.286187678161086	0.919024376808327
-1.34463539603662	0.331231948072771
1.79852188352525	-0.521229502940919
0.472189404752624	1.06030099737809
0.255400301598607	1.2752771357267
-1.34379748494324	-2.08102492585022
0.324521763482157	1.52897381492056
-1.43041179914401	-0.271317349526568
-1.09756851440387	-0.663586759722812
-1.04372092182988	-1.31261570557139
-1.64280281807315	0.568441836069828
1.29042671600037	-0.343301849074761
1.55295560344453	-0.198732246363775
0.222877736154015	0.91230421218286
-1.17482970389165	-1.24862129530304
-1.17046168724993	0.0548115865036639
1.33136978333496	-0.252979627369859
-0.571506403539276	0.878867334181586

C.3.2 Group 2

-1.16746441908748	-0.335962910514146
-1.34294013060903	-0.512400008273499
-0.721264031173109	-0.091574921418631
-0.88478886212507	-1.11271966484283
1.29510536156124	-0.618435732922602
-0.771271815148217	-0.392950775415555
1.17395383348143	-0.640719078349997
0.0048785180954753	-0.489892765175499
-0.928552125250176	-1.03290026894456
-0.0522179474361349	0.776695272924482
-0.0825968539139523	0.549140706393836
1.33670880468296	-0.745034190973435
-0.910051187188799	-0.83492114946746
0.276076544815898	0.991027680894188
1.27076867364182	-0.544861273246977
-1.07615872863243	-1.23185957098382
0.409782480919684	0.881288458045092
-0.65098342808144	-0.453505604254917
1.14990647950309	-0.728237045627949
0.288538435418871	1.2235094802462
1.37062152018357	-1.0426814825097
-1.41688671046778	-1.50009028685562

0.111361833817917	1.80764469594734
0.0848929167797318	1.75095139318827
-0.332431851227817	0.0640866948053936
-1.16117291040544	-0.427771578598902
-0.721054784866919	-0.93111793410897
1.38258675171664	-0.782805372208657
-0.943799561010877	-0.0950729459554145
0.0112723815525064	0.168753132003603

C.3.3 Group 3

-0.815262841951742	-0.884225822754394
-0.787154628243534	-0.0871836020654103
0.832213645786596	-1.00712663642927
1.63592847639642	-0.820860619292854
-1.09062496275221	-0.986389956297472
0.193430406267649	1.67304479613017
0.387118004299425	1.56231714589633
-1.47997817525861	-0.947185824896642
-1.05576786126791	-1.58829022780494
1.61208897992135	-0.45624269508864
0.212375702006937	1.49626406684974
1.88861838957372	-0.162120477449412
1.53176686880408	-0.5952350875607
0.175611051013942	1.60067110924502
0.216165746556758	-0.370976642546026
-0.572428337209138	-0.398420676117426
-0.0190175489517496	1.69155188596933
0.403442086374965	1.94442122892745
-1.11394388960048	-0.328749198124284
-1.18790713509424	-0.11107107285966
-1.27928481183383	1.34091575388402
1.42081170027514	-0.574192587932637
1.50918308163331	-0.457264894535138
-0.598775780285499	-0.0584205871452388
-0.71195164467903	-0.454685660343099
0.361426307667132	1.29622088774886
0.123088492156529	0.398868667273172
0.338849742271531	0.526620607714772
0.198509175207646	1.0497079858967
0.372058103225975	0.141850275902466
0.939537094687078	-0.94235594440351
-0.834753302091619	0.0560169161441602

-0.0316969955317014	1.20628636297068
1.31865899610251	-0.608941629448621
0.145850115540845	0.664762610158703
-0.874567041187382	-1.99045808660134
0.103245160342108	0.947237208375102
-1.31990220488434	-1.4213049819749
0.143799223979596	1.1773321315882
1.5167437247145	-0.777071193232019
1.62692718175579	-0.0438037851328646
-1.15477816737323	0.141781780453283
1.30729974557565	-0.524105037030218
-1.46372223711287	-1.45746043166993
0.523077921244384	0.781436363020525
1.15792264994112	-0.681781243942502
0.267064533246091	1.33811156287878
-0.96791589220077	-1.02191689682453
0.55104054211344	1.19198666951934
0.0073423470574356	2.03274833040974

Appendix D

Multinomial Mixture Data

All tables are for 2^p-way tables. Cell values are calculated by

$$\text{Cell} = \sum_{i=1}^{p} (i-1)^{X_i}$$

D.1 First Simulation: (2 Groups, N=2500, p=5)

$\mathcal{G}$	5 Star		5 ACF		[012][34]		[0][12][34]		[01][02][34]	
Cell	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2
0	533	74	587	78	542	65	435	27	497	61
1	43	3	101	27	92	15	113	46	60	7
2	68	42	41	18	45	15	75	22	60	22
3	22	7	53	47	35	7	30	27	34	25
4	82	48	42	27	1	17	58	17	75	28
5	26	8	6	11	30	24	15	25	46	22
6	12	19	21	13	19	23	54	48	7	18
7	17	17	20	52	53	141	24	71	29	94
8	68	48	52	18	114	26	77	16	102	21
9	28	9	7	12	18	7	34	21	18	4
10	12	25	7	5	10	5	17	7	18	20
11	15	17	3	10	5	2	4	10	8	14

$\mathcal{G}$	5 Star		5 ACF		[123][45]		[1][23][45]		[12][12][45]	
Cell	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2	Grp 1	Grp 2
12	15	18	25	16	1	13	8	4	13	17
13	15	19	6	10	7	8	4	11	9	14
14	3	25	11	32	7	19	20	29	4	9
15	29	72	19	80	9	73	10	40	15	67
16	82	40	80	27	66	29	59	12	70	21
17	20	5	10	19	14	20	17	24	5	6
18	13	31	8	12	7	11	16	10	8	17
19	9	17	9	33	3	3	5	15	3	22
20	9	23	5	12	0	12	8	17	9	17
21	12	19	2	5	5	14	4	30	7	25
22	2	25	4	17	4	14	13	32	2	14
23	15	70	10	38	12	81	5	51	11	95
24	11	19	41	51	80	105	43	38	54	80
25	10	17	7	27	16	26	20	83	3	18
26	4	19	1	12	6	39	13	27	16	45
27	21	86	8	38	8	3	8	50	13	50
28	1	21	17	65	2	27	9	33	20	48
29	22	65	7	29	11	43	6	82	11	63
30	0	28	12	97	6	58	24	103	4	48
31	31	314	28	312	22	305	22	222	19	238

D.2 Second Simulation (2 Groups, N=2500, p=4)

$\mathcal{G} = [12][13]$

	Mixture 1		Mixture 2	
Cell	Grp 1	Grp 2	Grp 1	Grp 2
0	9	496	2	201
1	315	56	296	194
2	53	62	65	70
3	121	133	121	129
4	49	62	44	56
5	138	132	141	138
6	519	6	531	304
7	46	303	50	158
Total	1250	1250	2000	500

D.3 Czech Autoworkers Data (N = 1841, p = 6)

Cell	n	Cell	n	Cell	n	Cell	n
0	44	16	23	32	5	48	7
1	40	17	32	33	7	49	3
2	112	18	70	34	21	50	14
3	67	19	66	35	9	51	14
4	129	20	50	36	9	52	9
5	145	21	80	37	17	53	16
6	12	22	7	38	1	54	2
7	23	23	13	39	4	55	3
8	35	24	24	40	4	56	4
9	12	25	25	41	3	57	0
10	80	26	73	42	11	58	13
11	33	27	57	43	8	59	11
12	109	28	51	44	14	60	5
13	67	29	63	45	17	61	14
14	7	30	7	46	5	62	4
15	9	31	16	47	2	63	4

from Dobra and Massam (2009)

Bibliography

Allcroft, D. J. and Glasbey, C. A. (2003). A latent Gaussian Markov random-field model for spatiotemporal rainfall disaggregation. *Applied Statistics*, 52(4):487–498.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to nonparametric problems. *The Annals of Statistics*, 2:1152–1174.

Asci, C., Napp, G., and Piccioni, M. (2006). The hyper-Dirichlet process and its discrete approximations: The butterfly model. *Journal of Multivariate Analysis*, 97(4):895–924.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). The infinite hidden markov model. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *NIPS*, pages 577–584. MIT Press.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2002). The infinite hidden Markov model. In *Machine Learning*, pages 29–245. MIT Press.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355.

Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Carvalho, C., Massam, H., and West, M. (2007). Simulation of hyper-inverse Wishart distributions in graphical models. *Biometrika*, 94:647–659.

Connor, R. J. and Mossiman, J. E. (1969). Concepts of independence of proportions with a generalization of the Dirichlet distribtution. *Journal of the American Statistical Association*, 64(325):194–206.

Dawid, A. P. (1981). Some matrix-variate distribution theory: Notational considerations and a bayesian application. *Biometrika*, 68(1):265–274.

Dawid, A. P. and Lauritzen, S. L. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21(3):1272–1317.

Dickey, J. M. (1967). Matricvariate generalizations of the multivariate t distribution and the inverted multivariate t distribution. *The Annals of Mathematical Statistics*, 38(2):511–518.

Dobra, A. and Massam, H. (2009). The mode oriented stochastic search MOSS algorithm for log-linear models with conjugate priors. *(submitted)*.

Doksum, K. (1974). Tailfree and neutral random probabilities and their posterior distributions. *The Annals of Probability*, 2(2):183–201.

Duan, J. A., Guindani, M., and Gelfand, A. E. (2007). Generalized spatial Dirichlet process models. *Biometrika*, 94(4):809–825.

Edwards, D. and Havranek, T. (1985). A fast procedure for model search in multidimensional contingency tables. *Biometrika*, 72(2):339–351.

Erosheva, E. A. and Fienberg, S. E. (2005). Bayesian mixed membership models for soft clustering and classification. In *ClassificationThe Ubiquitous Challenge*, pages 11–26. Springer-Verlag.

Escobar, M. D. (1994). Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Fabius, J. (1964). Asymptotic behaviorof Bayes' estimates. *The Annals of Mathematical Statistics*, 35(2):846–856.

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

Ferguson, T. S. (1974). Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4):615–629.

Fienberg, S. E., Hersh, P., Rinaldo, A., and Zhou, Y. (2010). Maximum likelihood estimation in latent class models for contingency table data. In Gibilisco, P., Riccomagno, E., Rogantin, M. P., and Wynn, H. P., editors, *Algebraic and Geometric Methods in Statistics*, pages 27–62. Cambridge University Press, Cambridge, UK.

Freedman, D. A. (1962). On the asymptotic behavior of Bayes' estimates in the discrete. *The Annals of Mathematical Statistics*, 34(4):1386–1403.

Frydenberg, M. and Lauritzen, S. L. (1989). Decomposition of maximum likelihood in mixed graphical interaction models. *Biometrika*, 76(3):539–555.

Gelfand, A. E., Kottas, A., and MacEachern, S. N. (2005). Bayesian nonparametric spatial modeling with Dirichlet processes mixing. *Journal of the American Statistical Association*, 100(471):1021–1035.

Ghosal, S., Ghosh, J., and Ramamoorthi, R. V. (1999). Posterior consistency of Dirichlet mixtures in density estimation. *The Annals of Statistics*, 27(1):143–158.

Giudici, P. and Green, P. (1999). Decomposable graphical gaussian model determination. *Biometrika*, 86:785–801.

Goldwater, S., Griffiths, T., and Johnson, M. (2006). Interpolating between types and tokens by estimating power-law generators. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 459–466. MIT Press, Cambridge, MA.

Griffin, J. E. and Steel, M. F. J. (2006). Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101(473).

Griffiths, T. and Ghahramani, Z. (2006). Infinite latent feature models and the indian buffet process. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 475–482. MIT Press, Cambridge, MA.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102:507–516.

Heinz, D. (2009). Building hyper dirichlet processes for graphical models. *Electronic Journal of Statistics*, 3:290–315.

Hjort, N. (1990). Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, 18(3):1259–1294.

Ishwaran, H. and James, L. (2001). Gibbs sampling measures for stick-breaking priors. *Journal of the American Statistical Association*, 96(452):161–173.

Ishwaran, H. and James, L. (2003). Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13:1211–1235.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). Experiments in stochastic computation for high-dimensional graphical models. *Statist. Sci*, 20:388–400.

Kalisch, M. and Buhlman, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Machine Learning Research*, 8:616–636.

Kshirsagar, A. M. (1959). Bartlett decomposition and wishart distribution. *The Annals of Mathematical Statistics*, 30(1):239–241.

Lavine, M. (1992). Some aspects of Pólya tree distributions for statistical modelling. *The Annals of Statistics*, 20(3):1222–1235.

Letac, G. and Massam, H. (2007). Wishart distributions for decomposable graphs. *The Annals of Statistics*, 35(3):1278–1323.

Liu, J. and Massam, H. (2006). The conjugate prior for discrete hierarchical log-linear models.

Lo, A. Y. (1984). On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357.

MacEachern, S. E. (1994). Estimating normal means with a conjugate style dirichlet process prior. *Communications in Statistics - Simulation and Computation*, 23(3):727–741.

MacEachern, S. N. (2000). Dependent dirichlet processes. *Unpublished Manuscript, Statistics Department, The Ohio State University*.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232.

McAuliffe, J. D., Blei, D. M., and Jordan, M. I. (2006). Nonparametric empirical Bayes for the Dirichlet process mixture model. *Statistics and Computing*, 16(1):5–14.

Mossiman, J. E. (1962). On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions. *Biometrika*, 49(1/2):65–82.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Statistics*, 25(2):855–900.

Rasmussen, C. E. (1999). The infinite gaussian mixture model. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *NIPS*, pages 554–560. The MIT Press.

Schervish, M. J. (1995). *Theory of Statistics*. Springer-Verlag, New York.

Sebastiani, M. R. (2003). Markov random-field models for estimating local labour markets. *Applied Statistics*, 52(2):201–211.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650.

Speed, T. and Kiiveri, H. (1986). Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150.

Spirtes, P., Glymour, C., and Scheines, R. (2001). *Causation, Prediction, and Search*. MIT Press, United States.

Sudderth, E. B. and Jordan, M. I. (2008). Shared segmentation of natural scenes using dependent pitman-yor processes. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *NIPS*, pages 1585–1592. MIT Press.

Teh, Y. W. (2006). A hierarchical Bayesian language model based on Pitman-Yor processes. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992, Morristown, NJ, USA. Association for Computational Linguistics.

Teh, Y. W., Görür, D., and Ghahramani, Z. (2007). Stick-breaking construction for the Indian buffet process. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101.

Thibaux, R. and Jordan, M. I. (2007). Hierarchical beta processes and the indian buffet process. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The Max-Min Hill-Climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

van Belle, G., Fisher, L. D., Heagerty, P. J., and Lumley, T. S. (2004). *Biostatistics: A Methodology for the Health Sciences*. John Wiley & Sons, Inc., Hoboken, NJ, 2nd edition.

Wallach, H., Sutton, C., and McCallum, A. (2008). Bayesian modeling of dependency trees using hierarchical Pitman-Yor priors. In *Proceedings of the Workshop on Prior Knowledge for Text and Language (held in conjunction with ICML/UAI/COLT)*, pages 15–20, Helsinki, Finland.

Xu, T., Zhang, Z. M., Yu, P. S., and Long, B. (2008). Evolutionary clustering by hierarchical dirichlet process with hidden markov state. *Data Mining, IEEE International Conference on*, 0:658–667.

Zachary, S. and Ziedins, I. (1999). Loss networks and Markov random fields. *Journal of Applied Probability*, 36:403–414.