Mixture models with a prior on the number of components

Jeffrey W. Miller*
Department of Statistical Science, Duke University
and
Matthew T. Harrison
Division of Applied Mathematics, Brown University

Abstract

A natural Bayesian approach for mixture models with an unknown number of components is to take the usual finite mixture model with symmetric Dirichlet weights, and put a prior on the number of components—that is, to use a mixture of finite mixtures (MFM). The most commonly-used method of inference for MFMs is reversible jump Markov chain Monte Carlo, but it can be nontrivial to design good reversible jump moves, especially in high-dimensional spaces. Meanwhile, there are samplers for Dirichlet process mixture (DPM) models that are relatively simple and are easily adapted to new applications. It turns out that, in fact, many of the essential properties of DPMs are also exhibited by MFMs—an exchangeable partition distribution, restaurant process, random measure representation, and stick-breaking representation—and crucially, the MFM analogues are simple enough that they can be used much like the corresponding DPM properties. Consequently, many of the powerful methods developed for inference in DPMs can be directly applied to MFMs as well; this simplifies the implementation of MFMs and can substantially improve mixing. We illustrate with real and simulated data, including high-dimensional gene expression data used to discriminate cancer subtypes.

Keywords: Bayesian, nonparametric, clustering, density estimation, model selection

*The authors gratefully acknowledge support from the National Science Foundation (NSF) grants DMS-1007593, DMS-1309004, and DMS-1045153, the National Institute of Mental Health (NIMH) grant R01MH102840, the Defense Advanced Research Projects Agency (DARPA) contract FA8650-11-1-715, and the National Institutes of Health (NIH) grant R01ES020619.

1 Introduction

Mixture models are used in a wide range of applications, including population structure (Pritchard et al., 2000), document modeling (Blei et al., 2003), speaker recognition (Reynolds et al., 2000), computer vision (Stauffer and Grimson, 1999), phylogenetics (Pagel and Meade, 2004), and gene expression profiling (Yeung et al., 2001), to name a few prominent examples. A common issue with finite mixtures is that it can be difficult to choose an appropriate number of mixture components, and many methods have been proposed for making this choice (e.g., Henna, 1985; Keribin, 2000; Leroux, 1992; Ishwaran et al., 2001; James et al., 2001).

From a Bayesian perspective, perhaps the most natural approach is to treat the number of components k like any other unknown parameter and put a prior on it. When the prior on the mixture weights is $\text{Dirichlet}_k(\gamma, \dots, \gamma)$ given k, we refer to such a model as a mixture of finite mixtures (MFM), for short. Several inference methods have been proposed for this type of model (Nobile, 1994; Phillips and Smith, 1996; Richardson and Green, 1997; Stephens, 2000; Nobile and Fearnside, 2007; McCullagh and Yang, 2008), the most commonly-used method being reversible jump Markov chain Monte Carlo (Green, 1995; Richardson and Green, 1997). Reversible jump is a very general technique, and has been successfully applied in many contexts, but it can be difficult to use since applying it to new situations requires one to design good reversible jump moves, which is often nontrivial, particularly in high-dimensional parameter spaces. Advances have been made toward general methods of constructing good reversible jump moves (Brooks et al., 2003; Hastie and Green, 2012), but these approaches add another level of complexity to the algorithm design process and/or the software implementation.

Meanwhile, infinite mixture models such as Dirichlet process mixtures (DPMs) have become popular, partly due to the existence of relatively simple and generic Markov chain Monte Carlo (MCMC) algorithms that can easily be adapted to new applications (Neal, 1992, 2000; MacEachern, 1994, 1998; MacEachern and Müller, 1998; Bush and MacEachern, 1996; West, 1992; West et al., 1994; Escobar and West, 1995; Liu, 1994; Dahl, 2003, 2005; Jain and Neal, 2004, 2007). These algorithms are made possible by the variety of computationally-tractable representations of the Dirichlet process, including its exchangeable partition distribution, the Blackwell–MacQueen urn process (a.k.a. the Chinese restaurant process), the random discrete measure formulation, and the Sethuraman–Tiwari stick-breaking representation (Ferguson, 1973; Antoniak, 1974; Blackwell and MacQueen, 1973; Aldous, 1985; Pitman, 1995, 1996; Sethuraman, 1994; Sethuraman and Tiwari, 1981).

It turns out that there are MFM counterparts for each of these properties: an exchangeable partition distribution, an urn/restaurant process, a random discrete measure formulation, and in certain cases, an elegant stick-breaking representation. The main point of this paper is that the MFM versions of these representations are simple enough, and parallel the DPM versions closely enough, that many of the inference algorithms developed for DPMs can be directly applied to MFMs. Interestingly, the key properties of MFMs hold for any choice of prior distribution on the number of components.

There has been a large amount of research on efficient inference methods for DPMs, and an immediate consequence of the present work is that most of these methods can also be used for MFMs. Since several DPM sampling algorithms are designed to have good mixing performance across a wide range of applications—for instance, the Jain–Neal split-merge samplers (Jain and Neal, 2004, 2007), coupled with incremental Gibbs moves (MacEachern, 1994; Neal, 1992, 2000)—this greatly simplifies the implementation of MFMs for new applications. Further, these algorithms can also provide a major improvement in mixing performance compared to the usual reversible jump approaches, particularly in high dimensions.

This work resolves an open problem discussed by Green and Richardson (2001), who noted that it would be interesting to be able to apply DPM samplers to MFMs:

> *"In view of the intimate correspondence between DP and [MFM] models discussed above, it is interesting to examine the possibilities of using either class of MCMC methods for the other model class. We have been unsuccessful in our search for incremental Gibbs samplers for the [MFM] models, but it turns out to be reasonably straightforward to implement reversible jump split/merge methods for DP models."*

The paper is organized as follows. In Sections 2 and 3, we formally define the MFM and show that it gives rise to a simple exchangeable partition distribution closely paralleling that of the Dirichlet process. In Section 4, we describe the Pólya urn scheme (restaurant process), random discrete measure formulation, and stick-breaking representation for the MFM. In Section 5, we establish some asymptotic results for MFMs. In Section 6, we show how the properties in Sections 3 and 4 lead to efficient inference algorithms for the MFM. In Section 7, we first make an empirical comparison between MFMs and DPMs on a simulation example, then compare the mixing performance of reversible jump versus our proposed algorithms, and finally, apply the model to high-dimensional gene expression data.

2 Model

We consider the following well-known model:

$$\begin{aligned}
&K \sim p_K, \text{ where } p_K \text{ is a p.m.f. on } \{1, 2, \dots\} \\
&(\pi_1, \dots, \pi_k) \sim \text{Dirichlet}_k(\gamma, \dots, \gamma), \text{ given } K = k \\
&Z_1, \dots, Z_n \overset{\text{iid}}{\sim} \pi, \text{ given } \pi \qquad\qquad (2.1) \\
&\theta_1, \dots, \theta_k \overset{\text{iid}}{\sim} H, \text{ given } K = k \\
&X_j \sim f_{\theta_{Z_j}} \text{ independently for } j = 1, \dots, n, \text{ given } \theta_{1:K}, Z_{1:n}.
\end{aligned}$$

Here, H is a prior or "base measure" on $\Theta \subset \mathbb{R}^\ell$, and $\{f_\theta : \theta \in \Theta\}$ is a family of probability densities with respect to a sigma-finite measure ζ on $\mathcal{X} \subset \mathbb{R}^d$. (As usual, we give Θ and $\mathcal{X}$ the Borel sigma-algebra, and assume $(x, \theta) \mapsto f_\theta(x)$ is measurable.) We denote $x_{1:n} = (x_1, \dots, x_n)$. Typically, $X_1, \dots, X_n$ would be observed, and all other variables would be hidden/latent. We refer to this as a *mixture of finite mixtures* (MFM) model.

It is important to note that we assume a symmetric Dirichlet with a single parameter γ not depending on k. This assumption is key to deriving a simple form for the partition distribution and the other resulting properties. Assuming symmetry in the distribution of π is quite natural, since the distribution of $X_1, \ldots, X_n$ under any asymmetric distribution on π would be the same as if this asymmetric distribution was replaced by its symmetrization, i.e., if the entries of π were uniformly permuted (although this would no longer necessarily be a Dirichlet distribution). Assuming the same γ for all k is a genuine restriction, albeit a fairly natural one, often made in such models even when not strictly necessary (Nobile, 1994; Phillips and Smith, 1996; Richardson and Green, 1997; Green and Richardson, 2001; Stephens, 2000; Nobile and Fearnside, 2007). Note that prior information about the relative sizes of the weights $\pi_1, \ldots, \pi_k$ can be introduced through γ—roughly speaking, small γ favors lower entropy π's, while large γ favors higher entropy π's.

Meanwhile, we put very few restrictions on p_K, the distribution of the number of components. For practical purposes, we need the infinite series $\sum_{k=1}^{\infty} p_K(k)$ to converge to 1 reasonably quickly, but any choice of p_K arising in practice should not be a problem. For certain theoretical purposes—in particular, consistency for the number of components—it is desirable to have $p_K(k) > 0$ for all $k \in \{1, 2, \ldots\}$.

For comparison, the Dirichlet process mixture (DPM) model with concentration parameter $\alpha > 0$ and base measure H is defined as follows, using the stick-breaking representation (Sethuraman, 1994):

$$\begin{aligned}
&B_1, B_2, \ldots \overset{\text{iid}}{\sim} \text{Beta}(1, \alpha)\\
&Z_1, \ldots, Z_n \overset{\text{iid}}{\sim} \pi, \ \text{ given } \pi = (\pi_1, \pi_2, \ldots) \text{ where } \pi_i = B_i \textstyle\prod_{j=1}^{i-1}(1 - B_j)\\
&\theta_1, \theta_2, \ldots \overset{\text{iid}}{\sim} H\\
&X_j \sim f_{\theta_{Z_j}} \text{ independently for } j = 1, \ldots, n, \text{ given } \theta_{1:\infty}, Z_{1:n}.
\end{aligned}$$

3 Exchangeable partition distribution

The primary observation on which our development relies is that the distribution on partitions induced by an MFM takes a form which is simple enough that it can be easily computed. Let $\mathcal{C}$ denote the partition of $[n] := \{1, \ldots, n\}$ induced by $Z_1, \ldots, Z_n$; in other words, $\mathcal{C} = \{E_i : |E_i| > 0\}$ where $E_i = \{j : Z_j = i\}$ for $i \in \{1, 2, \ldots\}$.

Theorem 3.1. *Under the MFM (Equation 2.1), the probability mass function of* $\mathcal{C}$ *is*

$$p(\mathcal{C}) = V_n(t) \prod_{c \in \mathcal{C}} \gamma^{(|c|)} \tag{3.1}$$

where $t = |\mathcal{C}|$ *is the number of parts/blocks in the partition, and*

$$V_n(t) = \sum_{k=1}^{\infty} \frac{k_{(t)}}{(\gamma k)^{(n)}} \, p_K(k). \tag{3.2}$$

Here, $x^{(m)} = x(x+1)\cdots(x+m-1)$ and $x_{(m)} = x(x-1)\cdots(x-m+1)$, with $x^{(0)} = 1$ and $x_{(0)} = 1$ by convention; note that $x_{(m)} = 0$ when x is a positive integer less than m. Theorem 3.1 can be derived from the well-known formula for $p(\mathcal{C}|k)$, which can found in, e.g., Green and Richardson (2001) or McCullagh and Yang (2008); we provide a proof for completeness. All proofs have been collected in Appendix A. We discuss computation of $V_n(t)$ in Section 3.2. For comparison, under the DPM, the partition distribution induced by $Z_1, \ldots, Z_n$ is $p_{\text{DP}}(\mathcal{C}|\alpha) = \frac{\alpha^t}{\alpha^{(n)}} \prod_{c\in\mathcal{C}} (|c|-1)!$ where $t = |\mathcal{C}|$ (Antoniak, 1974). When a prior is placed on α, as in the experiments in Section 7, the DPM partition distribution becomes

$$p_{\text{DP}}(\mathcal{C}) = V_n^{\text{DP}}(t) \prod_{c\in\mathcal{C}} (|c|-1)! \tag{3.3}$$

where $V_n^{\text{DP}}(t) = \int \frac{\alpha^t}{\alpha^{(n)}} p(\alpha) d\alpha$.

Viewed as a function of the block sizes $(|c| : c \in \mathcal{C})$, Equation 3.1 is an *exchangeable partition probability function* (EPPF) in the terminology of Pitman (1995, 2006), since it is a symmetric function of the block sizes. Consequently, $\mathcal{C}$ is an *exchangeable random partition* of $[n]$; that is, its distribution is invariant under permutations of $[n]$ (alternatively, this can be seen directly from the definition of the model, since $Z_1, \ldots, Z_n$ are exchangeable). In fact, Equation 3.1 induces an exchangeable random partition of the positive integers; see Section 3.3.

Further, Equation 3.1 is a member of the family of Gibbs partition distributions (Pitman, 2006); this is also implied by the results of Gnedin and Pitman (2006) characterizing the extreme points of the space of Gibbs partition distributions. Results on Gibbs partitions are provided by Ho et al. (2007), Lijoi et al. (2008), Cerquetti (2011), Cerquetti et al. (2013), Gnedin (2010), and Lijoi and Prünster (2010). However, the utility of this representation for inference in mixture models with a prior on the number of components does not seem to have been previously explored in the literature.

Due to Theorem 3.1, we have the following equivalent representation of the model:

$$\begin{aligned}
&\mathcal{C} \sim p(\mathcal{C}), \text{ with } p(\mathcal{C}) \text{ as in Equation 3.1} \\
&\phi_c \overset{\text{iid}}{\sim} H \text{ for } c \in \mathcal{C}, \text{ given } \mathcal{C} \\
&X_j \sim f_{\phi_c} \text{ independently for } j \in c,\ c \in \mathcal{C}, \text{ given } \phi, \mathcal{C},
\end{aligned} \tag{3.4}$$

where $\phi = (\phi_c : c \in \mathcal{C})$ is a tuple of $t = |\mathcal{C}|$ parameters $\phi_c \in \Theta$, one for each block $c \in \mathcal{C}$.

This representation is particularly useful for doing inference, since one does not have to deal with component labels or empty components. (If component-specific inferences are desired, a method such as that provided by Papastamoulis and Iliopoulos (2010) may be useful.) The formulation of models starting from a partition distribution has been a fruitful approach, exemplified by the development of product partition models (Hartigan, 1990; Barry and Hartigan, 1992; Quintana and Iglesias, 2003; Dahl, 2009; Park and Dunson, 2010; Müller and Quintana, 2010; Müller et al., 2011).

3.1 Basic properties

We list here some basic properties of the MFM model. See Appendix A for proofs. Denoting $x_c = (x_j : j \in c)$ and $m(x_c) = \int_\Theta \big[\prod_{j\in c} f_\theta(x_j)\big] H(d\theta)$ (with the convention that $m(x_\varnothing) = 1$), we have

$$p(x_{1:n}|\mathcal{C}) = \prod_{c\in\mathcal{C}} m(x_c). \tag{3.5}$$

While some authors use the terms "cluster" and "component" interchangeably, we use *cluster* to refer to a block c in a partition $\mathcal{C}$, and *component* to refer to a probability distribution f_{θ_i} in a mixture $\sum_{i=1}^k \pi_i f_{\theta_i}(x)$. The number of components K and the number of clusters $T = |\mathcal{C}|$ are related by

$$p(t|k) = \frac{k_{(t)}}{(\gamma k)^{(n)}} \sum_{\mathcal{C}:|\mathcal{C}|=t} \prod_{c\in\mathcal{C}} \gamma^{(|c|)}, \tag{3.6}$$

$$p(k|t) = \frac{1}{V_n(t)} \frac{k_{(t)}}{(\gamma k)^{(n)}}\, p_K(k), \tag{3.7}$$

where in Equation 3.6, the sum is over partitions $\mathcal{C}$ of $[n]$ such that $|\mathcal{C}| = t$. Note that $p(t|k)$ and $p(k|t)$ depend on n. Wherever possible, we use capital letters (such as K and T) to denote random variables as opposed to particular values (k and t). The formula for $p(k|t)$ is required for doing inference about the number of components K based on posterior samples of $\mathcal{C}$; fortunately, it is easy to compute. We have the conditional independence relations

$$\mathcal{C} \perp K \mid T, \tag{3.8}$$

$$X_{1:n} \perp K \mid T. \tag{3.9}$$

3.2 The coefficients $V_n(t)$

The following recursion is a special case of a more general result for Gibbs partitions (Gnedin and Pitman, 2006).

Proposition 3.2. *The numbers $V_n(t)$ (Equation 3.2) satisfy the recursion*

$$V_{n+1}(t+1) = V_n(t)/\gamma - (n/\gamma + t)V_{n+1}(t) \tag{3.10}$$

for any $0 \le t \le n$ and $\gamma > 0$.

This is easily seen by plugging the identity

$$k_{(t+1)} = (\gamma k + n)k_{(t)}/\gamma - (n/\gamma + t)k_{(t)}$$

into the expression for $V_{n+1}(t+1)$. In the case of $\gamma = 1$, Gnedin (2010) has discovered a beautiful example of a distribution on K for which both $p_K(k)$ and $V_n(t)$ have closed-form expressions.

In previous work on the MFM model, p_K has often been chosen to be proportional to a Poisson distribution restricted to the positive integers or a subset thereof (Phillips and Smith, 1996; Stephens, 2000; Nobile and Fearnside, 2007), and Nobile (2005) has proposed a theoretical justification for this choice. Interestingly, the model has some nice mathematical properties if one instead chooses $K-1$ to be given a Poisson distribution, that is, $p_K(k) = \text{Poisson}(k-1|\lambda)$ for some $\lambda > 0$. For instance, it turns out that if $p_K(k) = \text{Poisson}(k-1|\lambda)$ and $\gamma = 1$ then $V_n(0) = \big(1 - \sum_{k=1}^{n} p_K(k)\big)/\lambda^n$.

However, to do inference, it is not necessary to choose p_K to have any particular form—we just need to be able to compute $p(\mathcal{C})$, and in turn, we need to be able to compute $V_n(t)$. To this end, note that $k_{(t)}/(\gamma k)^{(n)} \leq k^t/(\gamma k)^n$, and thus the infinite series for $V_n(t)$ (Equation 3.2) converges rapidly when $t \ll n$. It always converges to a finite value when $1 \leq t \leq n$; this is clear from the fact that $p(\mathcal{C}) \in [0,1]$. This finiteness can also be seen directly from the series since $k^t/(\gamma k)^n \leq 1/\gamma^n$, and in fact, this shows that the series for $V_n(t)$ converges at least as rapidly (up to a constant) as the series $\sum_{k=1}^{\infty} p_K(k)$ converges to 1. Hence, for any reasonable choice of p_K (i.e., not having an extraordinarily heavy tail), $V_n(t)$ can easily be numerically approximated to a high level of precision. In practice, all of the required values of $V_n(t)$ can be precomputed before MCMC sampling, and this takes a negligible amount of time relative to MCMC (see Appendix B.1).

3.3 Self-consistent marginals

For each $n = 1, 2, \ldots$, let $q_n(\mathcal{C})$ denote the MFM distribution on partitions of $[n]$ (Equation 3.1). This family of partition distributions is preserved under marginalization, in the following sense.

Proposition 3.3. *If $m < n$ then q_m coincides with the marginal distribution on partitions of $[m]$ induced by q_n.*

In other words, drawing a sample from q_n and removing elements $m+1, \ldots, n$ from it yields a sample from q_m. This can be seen directly from the model definition (Equation 2.1), since $\mathcal{C}$ is the partition induced by the Z's, and the distribution of $Z_{1:m}$ is the same when the model is defined with any $n \geq m$. This property is sometimes referred to as *consistency in distribution* (Pitman, 2006).

By Kolmogorov's extension theorem (e.g., Durrett, 1996), it is well-known that this implies the existence of a unique probability distribution on partitions of the positive integers $\mathbb{Z}_{>0} = \{1, 2, \ldots\}$ such that the marginal distribution on partitions of $[n]$ is q_n for all $n \in \{1, 2, \ldots\}$ (Pitman, 2006).

4 Equivalent representations

4.1 Pólya urn scheme / Restaurant process

Pitman (1996) considered a general class of urn schemes, or restaurant processes, corresponding to exchangeable partition probability functions (EPPFs). The following scheme for the MFM falls into this general class.

Theorem 4.1. *The following process generates partitions $\mathcal{C}_1, \mathcal{C}_2, \ldots$ such that for any $n \in \{1, 2, \ldots\}$, the probability mass function of $\mathcal{C}_n$ is given by Equation 3.1.*

- *Initialize with a single cluster consisting of element 1 alone: $\mathcal{C}_1 = \{\{1\}\}$.*
- *For $n = 2, 3, \ldots$, element n is placed in . . .*

$$\text{an existing cluster } c \in \mathcal{C}_{n-1} \text{ with probability} \propto |c| + \gamma$$
$$\text{a new cluster with probability} \propto \frac{V_n(t+1)}{V_n(t)}\gamma$$

where $t = |\mathcal{C}_{n-1}|$.

Clearly, this bears a close resemblance to the Chinese restaurant process (i.e., the Blackwell–MacQueen urn process), in which the nth element is placed in an existing cluster c with probability $\propto |c|$ or a new cluster with probability $\propto \alpha$ (the concentration parameter) (Blackwell and MacQueen, 1973; Aldous, 1985).

4.2 Random discrete measures

The MFM can also be formulated starting from a distribution on discrete measures that is analogous to the Dirichlet process. With K, π, and $\theta_{1:K}$ as in Equation 2.1, let

$$G = \sum_{i=1}^{K} \pi_i \delta_{\theta_i}$$

where δ_θ is the unit point mass at θ. Let us denote the distribution of G by $\mathcal{M}(p_K, \gamma, H)$. Note that G is a random discrete measure over Θ. If H is continuous (i.e., $H(\{\theta\}) = 0$ for all $\theta \in \Theta$), then with probability 1, the number of atoms is K; otherwise, there may be fewer than K atoms. If we let $X_1, \ldots, X_n | G$ be i.i.d. from the resulting mixture, i.e., from

$$f_G(x) := \int f_\theta(x) G(d\theta) = \sum_{i=1}^{K} \pi_i f_{\theta_i}(x),$$

then the distribution of $X_{1:n}$ is the same as in Equation 2.1. So, in this notation, the MFM model is:

$$G \sim \mathcal{M}(p_K, \gamma, H)$$
$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} f_G, \text{ given } G.$$

This random discrete measure perspective is connected to work on *species sampling models* (Pitman, 1996) in the following way. When H is continuous, we can construct a species sampling model by letting $G \sim \mathcal{M}(p_K, \gamma, H)$ and modeling the observed data as $\beta_1, \ldots, \beta_n \sim G$ given G. We refer to Pitman (1996), Hansen and Pitman (2000), Ishwaran and James (2003), Lijoi et al. (2005, 2007), and Lijoi et al. (2008) for more background on species sampling models and further examples. Pitman derived a general formula for the posterior predictive distribution of a species sampling model; the following result is a special case. Note the close relationship to the restaurant process (Theorem 4.1).

Theorem 4.2. *If H is continuous, then $\beta_1 \sim H$ and the distribution of β_n given $\beta_1, \ldots, \beta_{n-1}$ is proportional to*

$$\frac{V_n(t+1)}{V_n(t)} \gamma H + \sum_{i=1}^{t} (n_i + \gamma) \delta_{\beta_i^*}, \tag{4.1}$$

where $\beta_1^*, \ldots, \beta_t^*$ are the distinct values taken by $\beta_1, \ldots, \beta_{n-1}$, and $n_i = \#\{j \in [n-1] : \beta_j = \beta_i^*\}$.

For comparison, when $G \sim \mathrm{DP}(\alpha H)$ instead, the distribution of β_n given $\beta_1, \ldots, \beta_{n-1}$ is proportional to $\alpha H + \sum_{j=1}^{n-1} \delta_{\beta_j} = \alpha H + \sum_{i=1}^{t} n_i \delta_{\beta_i^*}$, since $G | \beta_1, \ldots, \beta_{n-1} \sim \mathrm{DP}(\alpha H + \sum_{j=1}^{n-1} \delta_{\beta_j})$ (Ferguson, 1973; Blackwell and MacQueen, 1973).

4.3 Stick-breaking representation

The Dirichlet process has an elegant stick-breaking representation for the mixture weights $\pi_1, \pi_2, \ldots$ (Sethuraman, 1994; Sethuraman and Tiwari, 1981). This extraordinarily clarifying perspective has inspired a number of nonparametric models (MacEachern, 1999, 2000; Hjort, 2000; Ishwaran and Zarepour, 2000; Ishwaran and James, 2001; Griffin and Steel, 2006; Dunson and Park, 2008; Chung and Dunson, 2009; Rodriguez and Dunson, 2011; Broderick et al., 2012), has provided insight into the properties of related models (Favaro et al., 2012; Teh et al., 2007; Thibaux and Jordan, 2007; Paisley et al., 2010), and has been used to develop efficient inference algorithms (Ishwaran and James, 2001; Blei and Jordan, 2006; Papaspiliopoulos and Roberts, 2008; Walker, 2007; Kalli et al., 2011).

In a certain special case—namely, when $p_K(k) = \mathrm{Poisson}(k-1|\lambda)$ and $\gamma = 1$—we have noticed that the MFM also has an interesting representation that we describe using the stick-breaking analogy, although it is somewhat different in nature. Consider the following procedure:

> *Take a unit-length stick, and break off i.i.d.* $\mathrm{Exponential}(\lambda)$ *pieces until you run out of stick.*

In other words, let $\epsilon_1, \epsilon_2, \ldots \overset{\text{iid}}{\sim} \mathrm{Exponential}(\lambda)$, define $\tilde{K} = \min\{j : \sum_{i=1}^{j} \epsilon_i \geq 1\}$, let $\tilde{\pi}_i = \epsilon_i$ for $i = 1, \ldots, \tilde{K} - 1$, and let $\tilde{\pi}_{\tilde{K}} = 1 - \sum_{i=1}^{\tilde{K}-1} \tilde{\pi}_i$.

Proposition 4.3. *The stick lengths $\tilde{\pi}$ have the same distribution as the mixture weights π in the MFM model when $p_K(k) = \mathrm{Poisson}(k-1|\lambda)$ and $\gamma = 1$.*

This is a consequence of a standard construction for Poisson processes. This suggests a way of generalizing the MFM model: take any sequence of nonnegative random variables $(\epsilon_1, \epsilon_2, \ldots)$ (not necessarily independent or identically distributed) such that $\sum_{i=1}^{\infty} \epsilon_i > 1$ with probability 1, and define $\tilde{K}$ and $\tilde{\pi}$ as above. Although the distribution of $\tilde{K}$ and $\tilde{\pi}$ may be complicated, in some cases it might still be possible to do inference based on the stick-breaking representation. This might be an interesting way to introduce different kinds of prior information on the mixture weights, however, we have not explored this possibility.

5 Asymptotics

In this section, we consider the asymptotics of $V_n(t)$, the asymptotic relationship between the number of components and the number of clusters, and the approximate form of the conditional distribution on cluster sizes given the number of clusters.

5.1 Asymptotics of $V_n(t)$

Recall that $V_n(t) = \sum_{k=1}^{\infty} \frac{k_{(t)}}{(\gamma k)^{(n)}}\, p_K(k)$ (Equation 3.2) for $1 \leq t \leq n$, where $\gamma > 0$ and p_K is a p.m.f. on $\{1, 2, \dots\}$. When $a_n, b_n > 0$, we write $a_n \sim b_n$ to mean $a_n/b_n \to 1$ as $n \to \infty$.

Theorem 5.1. *For any $t \in \{1, 2, \dots\}$, if $p_K(t) > 0$ then*

$$V_n(t) \sim \frac{t_{(t)}}{(\gamma t)^{(n)}}\, p_K(t) \sim \frac{t!}{n!}\, \frac{\Gamma(\gamma t)}{n^{\gamma t - 1}}\, p_K(t) \tag{5.1}$$

as $n \to \infty$.

In particular, $V_n(t)$ has a simple interpretation, asymptotically—it behaves like the $k = t$ term in the series. All proofs have been collected in Appendix A.

5.2 Relationship between the number of clusters and number of components

In the MFM, it is perhaps intuitively clear that, under the prior at least, the number of clusters $T = |\mathcal{C}|$ behaves very similarly to the number of components K when n is large. It turns out that under the posterior they also behave very similarly for large n.

Theorem 5.2. *Let $x_1, x_2, \dots \in \mathcal{X}$ and $k \in \{1, 2, \dots\}$. If $p_K(1), \dots, p_K(k) > 0$ then*

$$\big| p(T = k \mid x_{1:n}) - p(K = k \mid x_{1:n}) \big| \longrightarrow 0$$

as $n \to \infty$.

5.3 Distribution of the cluster sizes under the prior

Here, we examine one of the major differences between the MFM and DPM priors. Roughly speaking, under the prior, the MFM prefers all clusters to be the same order of magnitude, while the DPM prefers having a few large clusters and many very small clusters. In the following calculations, we quantify the preceding statement more precisely. (See Green and Richardson (2001) for informal observations along these lines.) Interestingly, these prior influences remain visible in certain aspects of the posterior, even in the limit as n goes to infinity.

Let $\mathcal{C}$ be the random partition of $[n]$ in the MFM model (Equation 3.1), and let $A = (A_1, \dots, A_T)$ be the ordered partition of $[n]$ obtained by randomly ordering the blocks of $\mathcal{C}$, uniformly among the $T!$ possible choices, where $T = |\mathcal{C}|$. Then

$$p(A) = \frac{p(\mathcal{C})}{|\mathcal{C}|!} = \frac{1}{t!} V_n(t) \prod_{i=1}^{t} \gamma^{(|A_i|)},$$

where $t = |\mathcal{C}|$. Now, let $S = (S_1, \ldots, S_T)$ be the vector of block sizes of A, that is, $S_i = |A_i|$. Then

$$p(S = s) = \sum_{A : S(A) = s} p(A) = V_n(t)\, \frac{n!}{t!} \prod_{i=1}^{t} \frac{\gamma^{(s_i)}}{s_i!}$$

for $s \in \Delta_t$, $t \in \{1, \ldots, n\}$, where $\Delta_t = \{s \in \mathbb{Z}^t : \sum_i s_i = n,\ s_i \geq 1\, \forall i\}$ (i.e., Δ_t is the set of t-part compositions of n). For any $x > 0$, by writing $x^{(m)}/m! = \Gamma(x + m)/(m!\,\Gamma(x))$ and using Stirling's approximation, we have $x^{(m)}/m! \sim m^{x-1}/\Gamma(x)$ as $m \to \infty$. This yields the approximations

$$p(S = s) \approx \frac{V_n(t)}{\Gamma(\gamma)^t}\, \frac{n!}{t!} \prod_{i=1}^{t} s_i^{\gamma - 1} \approx \frac{p_K(t)}{n^{\gamma t - 1}}\, \frac{\Gamma(\gamma t)}{\Gamma(\gamma)^t} \prod_{i=1}^{t} s_i^{\gamma - 1}$$

(using Theorem 5.1 in the second step), and

$$p(S = s \mid T = t) \approx \kappa \prod_{i=1}^{t} s_i^{\gamma - 1}$$

for $s \in \Delta_t$, where κ is a normalization constant. Thus, $p(s|t)$ has approximately the same shape as a symmetric t-dimensional Dirichlet distribution, except it is discrete. This would be obvious if we were conditioning on the number of components K, and it makes intuitive sense when conditioning on T, since K and T are essentially the same for large n.

It is interesting to compare $p(s)$ and $p(s|t)$ to the corresponding distributions for the Dirichlet process. For the DP with a prior on α, we have $p_{\text{DP}}(\mathcal{C}) = V_n^{\text{DP}}(t) \prod_{c \in \mathcal{C}} (|c| - 1)!$ by Equation 3.3, and $p_{\text{DP}}(A) = p_{\text{DP}}(\mathcal{C})/|\mathcal{C}|!$ as before, so for $s \in \Delta_t$, $t \in \{1, \ldots, n\}$,

$$p_{\text{DP}}(S = s) = V_n^{\text{DP}}(t)\, \frac{n!}{t!}\, s_1^{-1} \cdots s_t^{-1}$$

and

$$p_{\text{DP}}(S = s \mid T = t) \propto s_1^{-1} \cdots s_t^{-1},$$

which has the same shape as a t-dimensional Dirichlet distribution with all the parameters taken to 0 (noting that this is normalizable since Δ_t is finite). Asymptotically in n, $p_{\text{DP}}(s|t)$ puts all of its mass in the "corners" of the discrete simplex Δ_t, while under the MFM, $p(s|t)$ remains more evenly dispersed.

6 Inference algorithms

Many approaches have been used for posterior inference in the MFM model (Equation 2.1). Nobile (1994) approximates the marginal likelihood $p(x_{1:n}|k)$ of each k in order to compute the posterior on k, and uses standard methods given k. Phillips and Smith (1996) and Stephens (2000) use jump diffusion and point process approaches, respectively, to sample from $p(k, \pi, \theta | x_{1:n})$. Richardson and Green (1997) employ reversible jump moves to sample from $p(k, \pi, \theta, z | x_{1:n})$. In cases with conjugate priors, Nobile and Fearnside (2007) use various Metropolis–Hastings moves on $p(k, z|x_{1:n})$, and McCullagh and Yang (2008) develop an approach that could in principle be used to sample from $p(k, \mathcal{C}|x_{1:n})$.

Due to the results of Sections 3 and 4, much of the extensive body of work on MCMC samplers for DPMs can be directly applied to MFMs. This leads to a variety of MFM samplers, of which we consider two main types: (a) sampling from $p(\mathcal{C}|x_{1:n})$ in cases where the marginal likelihood $m(x_c) = \int_\Theta \big[\prod_{j\in c} f_\theta(x_j)\big] H(d\theta)$ can easily be computed (typically, this means H is a conjugate prior), and (b) sampling from $p(\mathcal{C}, \phi|x_{1:n})$ (in the notation of Equation 3.4) in cases where $m(x_c)$ is intractable or when samples of the component parameters are desired.

When $m(x_c)$ can easily be computed, the following Gibbs sampling algorithm can be used to sample from the posterior on partitions, $p(\mathcal{C}|x_{1:n})$. Given a partition $\mathcal{C}$, let $\mathcal{C} \setminus j$ denote the partition obtained by removing element j from $\mathcal{C}$.

1. Initialize $\mathcal{C} = \{[n]\}$ (i.e., one cluster).

2. Repeat the following steps N times, to obtain N samples.

```
For $j = 1, \ldots, n$: Remove element $j$ from $\mathcal{C}$, and place it ...
    in $c \in \mathcal{C} \setminus j$ with probability $\propto (|c| + \gamma)\dfrac{m(x_{c\cup j})}{m(x_c)}$
    in a new cluster with probability $\propto \gamma\, \dfrac{V_n(t+1)}{V_n(t)}\, m(x_j)$
where $t = |\mathcal{C} \setminus j|$.
```

This is a direct adaptation of "Algorithm 3" for DPMs (MacEachern, 1994; Neal, 1992, 2000). The only differences are that in Algorithm 3, $|c| + \gamma$ is replaced by $|c|$, and $\gamma V_n(t+1)/V_n(t)$ is replaced by α (the concentration parameter). Thus, the differences between the MFM and DPM versions of the algorithm are precisely the same as the differences between their respective urn/restaurant processes. In order for the algorithm to be valid, the Markov chain needs to be irreducible, and to achieve this it is necessary that $\big\{t \in \{1, 2, \ldots\} : V_n(t) > 0\big\}$ be a block of consecutive integers containing 1. It turns out that this is always the case, since for any k such that $p_K(k) > 0$, we have $V_n(t) > 0$ for all $t = 1, \ldots, k$.

Note that one only needs to compute $V_n(1), \ldots, V_n(t^*)$ where t^* is the largest value of t visited by the sampler. In practice, it is convenient to precompute these values using a guess at an upper bound t_{pre} on t^*, and this takes a negligible amount of time compared to running the sampler (see Appendix B.1). Just to be clear, t_{pre} is only introduced for computational expedience—the model allows unbounded t, and in the event that the sampler visits $t > t_{\text{pre}}$, it is trivial to increase t_{pre} and compute the required values of $V_n(t)$ on demand.

When $m(x_c)$ cannot easily be computed, a clever auxiliary variable technique referred to as "Algorithm 8" can be used for inference in the DPM (Neal, 2000; MacEachern and Müller, 1998). Making the same substitutions as above, we can apply Algorithm 8 for inference in the MFM as well, to sample from $p(\mathcal{C}, \phi|x_{1:n})$.

A well-known issue with incremental Gibbs samplers such as Algorithms 3 and 8, when applied to DPMs, is that mixing can be somewhat slow, since it may take a long time to create or destroy substantial clusters by moving one element at a time. With MFMs, this issue seems to be exacerbated, since compared with DPMs, MFMs tend to put small probability on partitions with tiny clusters (see Section 5.3), making it difficult for the sampler to move through these regions of the space.

To deal with this issue, split-merge samplers for DPMs have been developed, in which a large number of elements can be reassigned in a single move (Dahl, 2003, 2005; Jain and Neal, 2004, 2007). In the same way as the incremental samplers, one can directly apply these split-merge samplers to MFMs as well, by simply plugging-in the MFM partition distribution in place of the DPM partition distribution. More generally, it seems likely that any partition-based MCMC sampler for DPMs could be applied to MFMs as well. In Section 7, we apply the Jain–Neal split-merge samplers, coupled with incremental Gibbs samplers, to do inference for MFMs in both conjugate and non-conjugate settings.

As usual in mixture models, the MFM is invariant under permutations of the component labels, and this can lead to so-called "label-switching" issues if one naively attempts to estimate quantities that are not invariant to labeling—such as the mean of a given component—by averaging MCMC samples. In the experiments below, we only estimate quantities that are invariant to labeling, such as the density, the number of components, and the probability that two given points belong to the same cluster. Consequently, the label-switching issue does not arise here. If inference for non-invariant quantities is desired, methods for this purpose have been proposed (Papastamoulis and Iliopoulos, 2010).

In addition to the partition-based algorithms above, several DPM inference algorithms are based on the stick-breaking representation, such as the slice samplers for mixtures (Walker, 2007; Kalli et al., 2011). Although we have not explored this approach, it should be possible to adapt these to MFMs using the stick-breaking representation described in Section 4.3.

7 Empirical demonstrations

In Section 7.1, we illustrate some of the similarities and differences between MFMs and DPMs. Section 7.2 compares the mixing performance of reversible jump MCMC versus the Jain–Neal split-merge algorithms proposed in Section 6 for inference in the MFM. In Section 7.3, we apply an MFM model to discriminate cancer subtypes based on gene expression data. All of the examples below happen to involve Gaussian component densities, but of course our approach is not limited to mixtures of Gaussians.

7.1 Similarities and differences between MFMs and DPMs

In the preceding sections, we established that MFMs share many of the mathematical properties of DPMs. Here, we empirically compare MFMs and DPMs, using simulated data from a three-component bivariate Gaussian mixture. Our purpose is not to argue that either model should be preferred over the other, but simply to illustrate that in certain respects they are very similar, while in other respects they differ. In general, we would not say that either model is uniformly better than the other; rather, one should choose the model which is best suited to the application at hand—specifically, if one believes there to be infinitely many components, then an infinite mixture such as a DPM is more appropriate, while if one expects finitely many components, an MFM is likely to be a better choice. For additional empirical analysis of MFMs versus DPMs, we refer to Green and Richardson (2001), who present different types of comparisons than we consider here.

7.1.1 Setup: data, model, and inference

Consider the data distribution $\sum_{i=1}^{3} w_i \mathcal{N}(\mu_i, C_i)$ where $w = (0.45, 0.3, 0.25)$, $\mu_1 = \left(\begin{smallmatrix}4\\4\end{smallmatrix}\right)$, $\mu_2 = \left(\begin{smallmatrix}7\\4\end{smallmatrix}\right)$, $\mu_3 = \left(\begin{smallmatrix}6\\2\end{smallmatrix}\right)$, $C_1 = \left(\begin{smallmatrix}1 & 0\\0 & 1\end{smallmatrix}\right)$, $C_2 = R\left(\begin{smallmatrix}2.5 & 0\\0 & 0.2\end{smallmatrix}\right)R^{\mathsf{T}}$ where $R = \left(\begin{smallmatrix}\cos\rho & -\sin\rho\\ \sin\rho & \cos\rho\end{smallmatrix}\right)$ with $\rho = \pi/4$, and $C_3 = \left(\begin{smallmatrix}3 & 0\\0 & 0.1\end{smallmatrix}\right)$. See Figure 1 (bottom).

For the model, we use multivariate normal component densities $f_\theta(x) = f_{\mu,\Lambda}(x) = \mathcal{N}(x|\mu, \Lambda^{-1})$, and for the base measure (prior) H on $\theta = (\mu, \Lambda)$, we take $\mu \sim \mathcal{N}(\widehat{\mu}, \widehat{C})$, $\Lambda \sim \text{Wishart}_d(V, \nu)$ independently, where $\widehat{\mu}$ is the sample mean, $\widehat{C}$ is the sample covariance, $\nu = d = 2$, and $V = \widehat{C}^{-1}/\nu$. Here, $\text{Wishart}_d(\Lambda|V, \nu) \propto |\det\Lambda|^{(\nu-d-1)/2} \exp\left(-\frac{1}{2}\text{tr}(V^{-1}\Lambda)\right)$. Note that this is a data-dependent prior. Note further that taking μ and Λ to be independent results in a non-conjugate prior; this is appropriate when the location of the data is not informative about the covariance (and vice versa).

For the MFM, we take $K \sim \text{Geometric}(r)$ ($p_K(k) = (1-r)^{k-1}r$ for $k = 1, 2, \ldots$) with $r = 0.1$, and we choose $\gamma = 1$ for the finite-dimensional Dirichlet parameters. For the DPM, we put an Exponential(1) prior on the concentration parameter, α. This makes the priors on the number of clusters t roughly comparable for the range of n values considered below; see Figure 4.

For both the MFM and DPM, we use the split-merge sampler of Jain and Neal (2007) for non-conjugate priors, coupled with Algorithm 8 of Neal (2000) (using a single auxiliary variable) for incremental Gibbs updates to the partition. Specifically, following Jain and Neal (2007), we use the (5,1,1,5) scheme: 5 intermediate scans to reach the split launch state, 1 split-merge move per iteration, 1 incremental Gibbs scan per iteration, and 5 intermediate moves to reach the merge launch state. Gibbs updates to the DPM concentration parameter α are made using Metropolis–Hastings moves.

Five independent datasets were used for each $n \in \{50, 100, 250, 1000, 2500\}$, and for each model (MFM and DPM), the sampler was run for 5,000 burn-in iterations and 95,000 sample iterations (for a total of 100,000). Judging by traceplots and running averages of various statistics, this appeared to be sufficient for mixing. The cluster sizes were recorded after each iteration, and to reduce memory storage requirements, the full state of the chain was recorded only once every 100 iterations. For each run, the seed of the random number generator was initialized to the same value for both the MFM and DPM. The sampler used for these experiments took approximately $8 \times 10^{-6}\, n$ seconds per iteration, where n is the number of observations in the dataset. All experiments were performed using a 2.80 GHz processor with 6 GB of RAM.

7.1.2 Density estimation

For certain nonparametric density estimation problems, both the DPM and MFM have been shown to exhibit posterior consistency at the minimax optimal rate, up to logarithmic

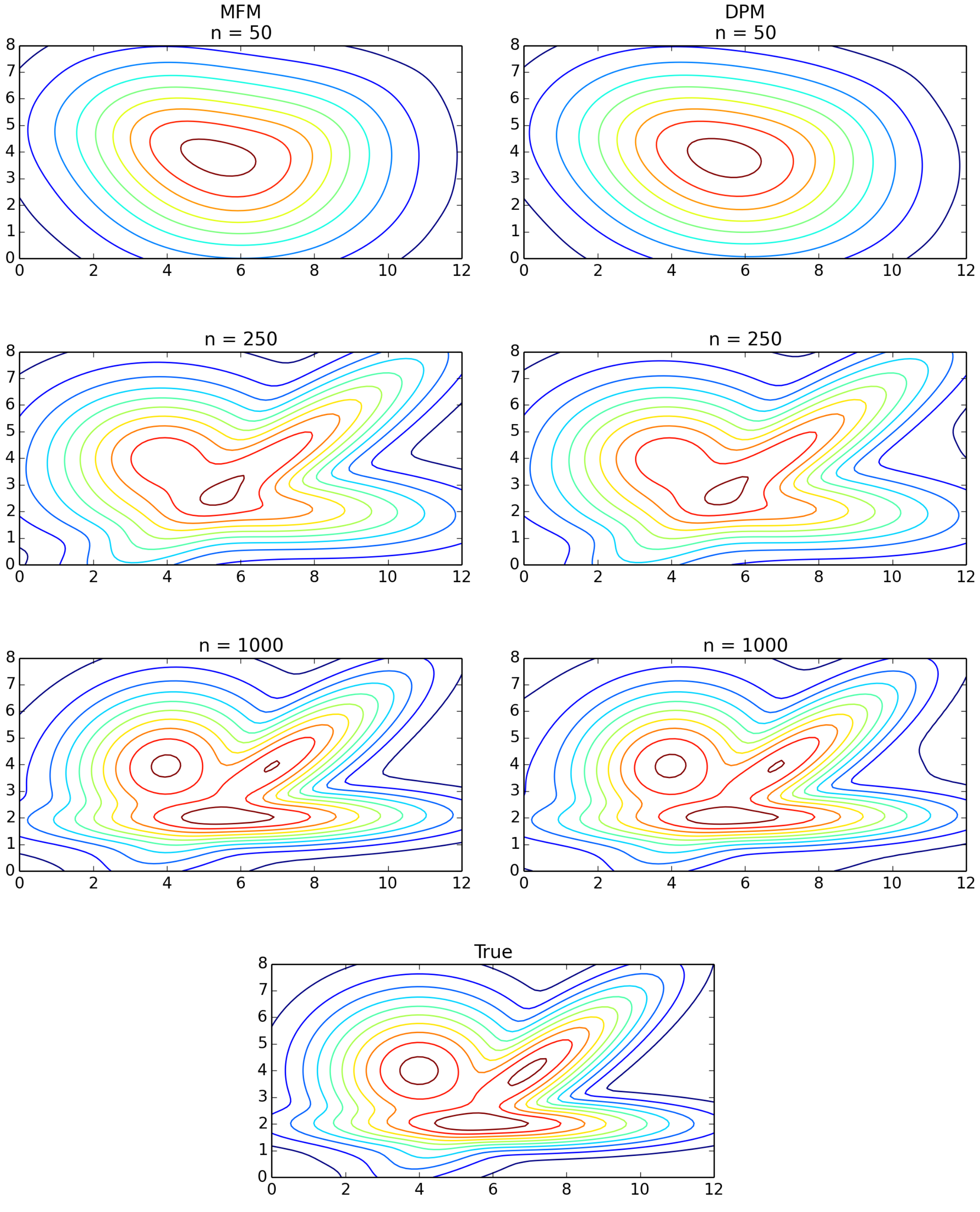

Figure 1: Density estimates for MFM (left) and DPM (right) on increasing amounts of data from the bivariate example (bottom). As n increases, the estimates appear to be converging to the true density, as expected.

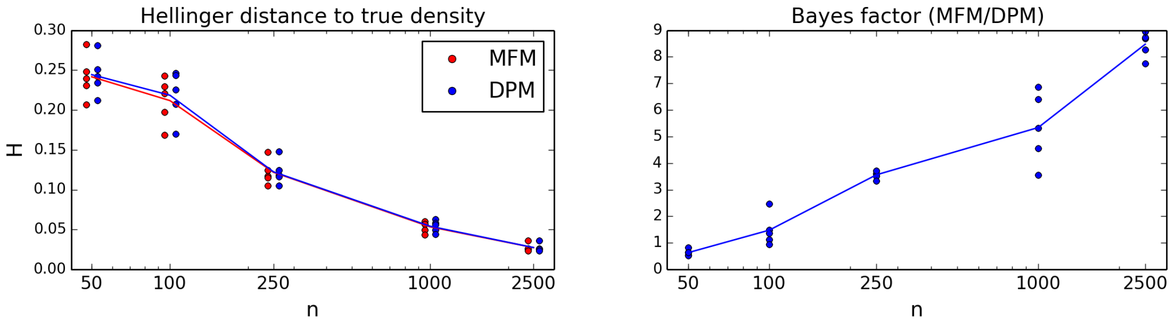

Figure 2: Left: Hellinger distance between the estimated and true densities, for MFM (red, left) and DPM (blue, right) density estimates, on the bivariate example. Right: Bayes factors in favor of the MFM over the DPM. For each $n \in \{50, 100, 250, 1000, 2500\}$, five independent datasets of size n were used, and the lines connect the averages for each n.

factors (Kruijer et al., 2010; Ghosal and Van der Vaart, 2007). In fact, even for small sample sizes, we observe empirically that density estimates under the two models are remarkably similar; for example, see Figure 1. (See Appendix B.2 for details on computing the density estimates.) As the amount of data increases, the density estimates for both models appear to be converging to the true density, as expected. Indeed, Figure 2 indicates that the Hellinger distance between the estimated density and the true density is going to zero, and further, that the rate of convergence appears to be nearly identical for the DPM and the MFM. This suggests that for density estimation, the behavior of the two models seems to be essentially the same.

On the other hand, Figure 2 shows that the Bayes factors $p(x_{1:n}|\text{MFM})/p(x_{1:n}|\text{DPM})$ are increasing as n grows, indicating that the MFM is a better fit to this data than the DPM, in the sense that it has a higher marginal likelihood. This makes sense, since the MFM is correctly specified for data from a finite mixture, while the DPM is not.

7.1.3 Clustering

Frequently, mixture models are used for clustering and latent class discovery, rather than for density estimation. MFMs have a partition distribution that takes a very similar form to that of the Dirichlet process, as discussed in Section 3. Despite this similarity, the MFM partition distribution differs in two fundamental respects.

(1) The prior on the number of clusters t is very different. In an MFM, one has complete control over the prior on the number of components k, which in turn provides control over the prior on t. Further, as the sample size n grows, the MFM prior on t converges to the prior on k. In contrast, in a Dirichlet process, the prior on t takes a particular parametric form and diverges at a $\log n$ rate.

(2) Given t, the prior on the cluster sizes is very different. In an MFM, most of the prior mass is on partitions in which the sizes of the clusters are all the same order of magnitude, while in a Dirichlet process, most of the prior mass is on partitions in

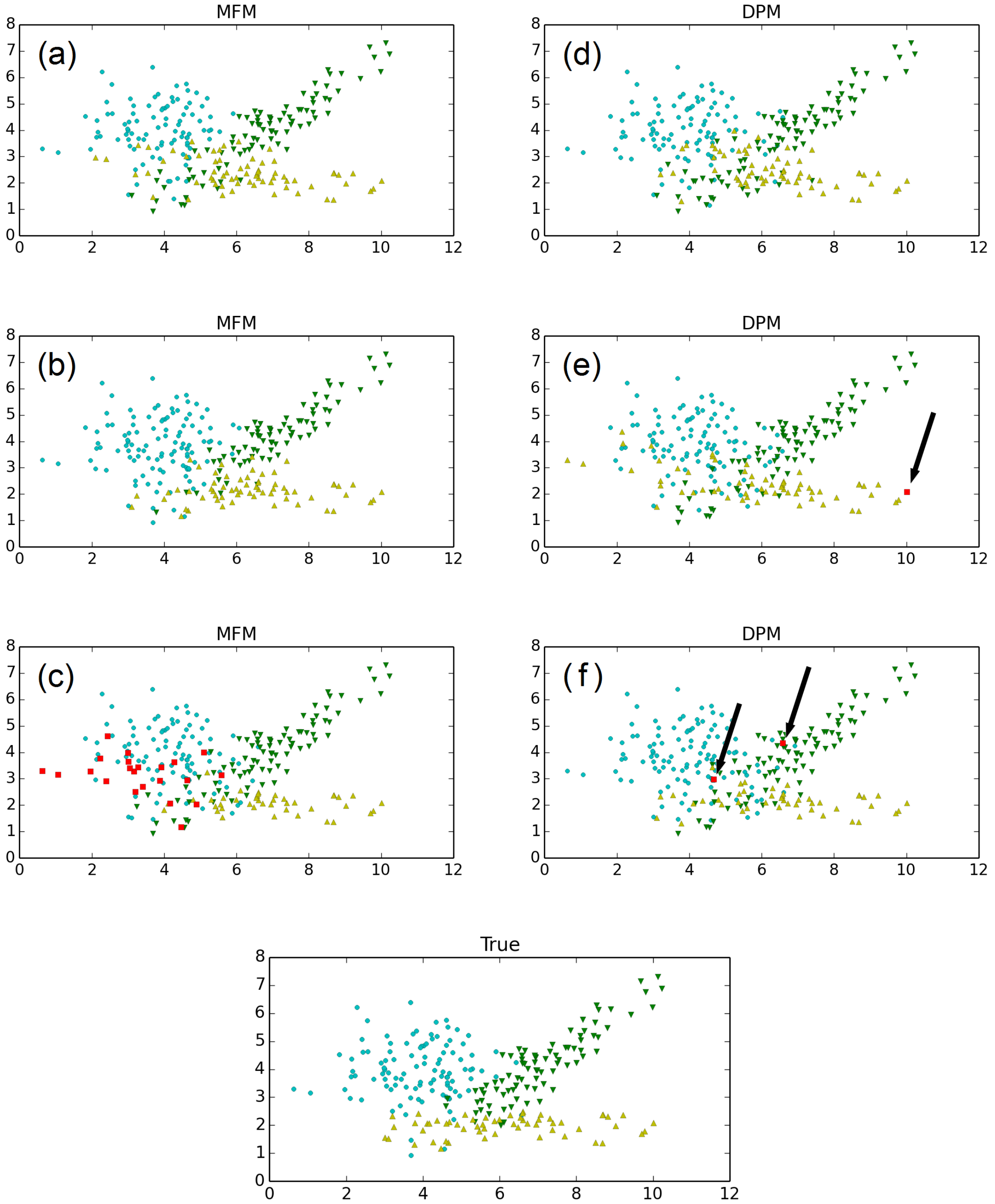

Figure 3: Representative sample clusterings from the posterior for the MFM and DPM, on $n = 250$ data points from the bivariate example. The bottom plot shows the true component assignments. See discussion in the text. (Best viewed in color.)

which the sizes vary widely, with a few large clusters and many very small clusters (see Section 5.3).

These prior differences carry over to noticeably different posterior clustering behavior. Figure 3 displays a few representative clusterings sampled from the posterior for the three-component bivariate Gaussian example. Around 92% of the MFM samples look like (a) and (b), while around 2/3 of the DPM samples look like (d). Since the MFM is consistent for k, the proportion of MFM samples with three clusters tends to 100% as n increases, however, we do not expect this to occur for the DPM (see Section 7.1.4); indeed, when $n = 2500$, it is 98% for the MFM and still around 2/3 for the DPM. Many of the DPM samples have tiny "extra" clusters consisting of only a few points, as seen in Figure 3 (e) and (f). Meanwhile, when the MFM samples have extra clusters, they tend to be large, as in (c). All of this is to be expected, due to items (1) and (2) above, along with the fact that these data are drawn from a finite mixture over the assumed family, and thus the MFM is correctly specified, while the DPM is not.

7.1.4 Mixing distribution and the number of components

Assuming that the data are from a finite mixture, it is also sometimes of interest to infer the mixing distribution or the number of components, subject to the caveat that these inferences are meaningful only to the extent that the component distributions are correctly specified and the model is finite-mixture identifiable. (In the notation of Section 4.2, the *mixing distribution* is G, and we say the model is *finite-mixture identifiable* if for any discrete measures G, G' supported on finitely-many points, $f_G = f_{G'}$ a.e. implies $G = G'$.) While Nguyen (2013) has shown that under certain conditions, DPMs are consistent for the mixing distribution (in the Wasserstein metric), Miller and Harrison (2013, 2014) have shown that the DPM posterior on the number of clusters is typically not consistent for the number of components, at least when the concentration parameter is fixed; the question remains open when using a prior on the concentration parameter, but we conjecture that it is still not consistent. On the other hand, MFMs are consistent for the mixing distribution and the number of components (for Lebesgue almost-all values of the true parameters) under very general conditions (Nobile, 1994); this is a straightforward consequence of Doob's theorem. The relative ease with which this consistency can be established for MFMs is due to the fact that in an MFM, the parameter space is a countable union of finite-dimensional spaces.

These consistency/inconsistency properties are readily observed empirically—they are not simply large-sample phenomena. As seen in Figure 4, the tendency of DPM samples to have tiny extra clusters causes the number of clusters t to be somewhat inflated, apparently making the DPM posterior on t fail to concentrate, while the MFM posterior on t concentrates at the true value (by Section 5.2). In addition to the number of clusters t, the MFM also permits inference for the number of components k in a natural way (Figure 4), while in the DPM the number of components is always infinite. One might wonder whether the DPM would perform better if the data were drawn from a mixture with an additional one or two components with much smaller weight, however, this doesn't seem to make a difference; see Appendix B.3. On the other hand, on data from an infinite mixture,

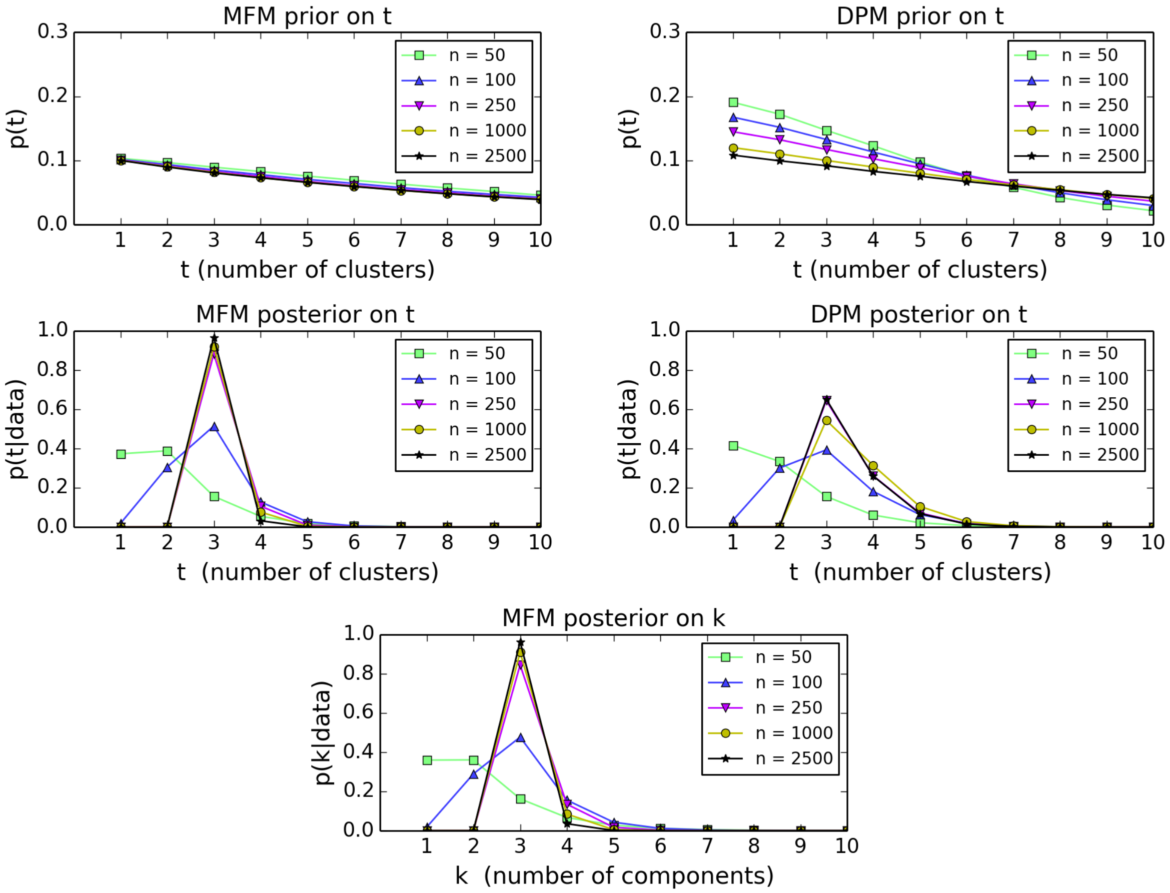

Figure 4: Prior and posterior on the number of clusters t for the MFM (left) and DPM (right), and the posterior on the number of components k for the MFM (bottom), on increasing amounts of data from the bivariate example. Each prior on t is approximated from 10^6 samples from the prior.

one would expect the DPM to perform better than the MFM. See Appendix B.2 for the formulas used to compute the posterior on k.

7.1.5 Issues with estimating the number of components

Due to the fact that inference for the number of components is a topic of high interest in many research communities, it seems prudent to make some cautionary remarks in this regard. Many approaches have been proposed for estimating the number of components (Henna, 1985; Keribin, 2000; Leroux, 1992; Ishwaran et al., 2001; James et al., 2001; Henna, 2005; Woo and Sriram, 2006, 2007). In theory, the MFM model provides a Bayesian approach to consistently estimating the number of components, making it a potentially attractive method of assessing the heterogeneity of the data. However, there are several possible pitfalls to consider, some of which are more obvious than others.

An obvious potential issue is that in many applications, the clusters which one wishes to distinguish are purely notional (for example, perhaps, clusters of images or documents), and a mixture model is used for practical purposes, rather than because the data are actually thought to arise from a mixture. Clearly, in such cases, inference for the "true" number of components is meaningless. On the other hand, in some applications, the data definitely come from a mixture (for example, extracellular recordings of multiple neurons), so there is in reality a true number of components, although usually the form of the mixture components is far from clear.

More subtle issues are that the posteriors on k and t can be (1) strongly affected by the base measure H, and (2) sensitive to misspecification of the family of component distributions $\{f_\theta\}$. Issue (1) can be seen, for instance, in the case of normal mixtures: it might seem desirable to choose the prior on the component means to have large variance in order to be less informative, however, this causes the posteriors on k and t to favor smaller values (Richardson and Green, 1997; Stephens, 2000; Jasra et al., 2005). The basic mechanism at play here is the same as in the Bartlett–Lindley paradox, and shows up in many Bayesian model selection problems. With some care, this issue can be dealt with by varying the base measure H and observing the effect on the posterior—that is, by performing a sensitivity analysis—for instance, see Richardson and Green (1997).

Issue (2) is more serious. In practice, we typically cannot expect our choice of $\{f_\theta : \theta \in \Theta\}$ to contain the true component densities (assuming the data are even from a mixture). When the model is misspecified in this way, the posteriors of k and t can be severely affected and depend strongly on n. For instance, if the model uses Gaussian components, and the true data distribution is not a finite mixture of Gaussians, then the posteriors of k and t can be expected to diverge to infinity as n increases. Consequently, the effects of misspecification need to be carefully considered if these posteriors are to be used as measures of heterogeneity. Steps toward addressing the issue of robustness have been taken by Woo and Sriram (2006, 2007) and Rodríguez and Walker (2014), however, this is an important problem demanding further study.

Despite these issues, sample clusterings and estimates of the number of components or clusters can provide a useful tool for exploring complex datasets, particularly in the case of high-dimensional data that cannot easily be visualized. It should always be borne in mind, though, that the results can only be interpreted as being correct to the extent that the model assumptions are correct.

7.2 Comparison with reversible jump MCMC

We compare two methods of inference for the MFM: the usual reversible jump MCMC (RJMCMC) approach, and our proposed approach based on the Jain–Neal split-merge algorithms. This purpose of this is (1) to compare the mixing performance of our method versus RJMCMC, and (2) to demonstrate the validity of our method by showing agreement with results obtained using RJMCMC. In addition to having better mixing performance, our method has the significant advantage of providing generic algorithms that apply to a wide variety of component distributions and priors, while RJMCMC requires one to design specialized reversible jump moves.

Note that RJMCMC is a very general framework and an enormous array of MCMC algorithms could be considered to be a special case of it. The most commonly-used instantiation of RJMCMC for univariate Gaussian mixtures seems to be the original algorithm of Richardson and Green (1997), so it makes sense to compare with this. For multivariate Gaussian mixtures, several RJMCMC algorithms have been proposed (Marrs, 1998; Zhang et al., 2004; Dellaportas and Papageorgiou, 2006), however, the model we consider below uses diagonal covariance matrices, and in this case, the Richardson and Green (1997) moves can simply be extended to each coordinate. More efficient algorithms almost certainly exist within the vast domain of RJMCMC, but the goal here is to compare with "plain-vanilla RJMCMC" as it is commonly-used.

7.2.1 Comparison with RJMCMC on the galaxy dataset

The galaxy dataset (Roeder, 1990) is a standard benchmark for mixture models, consisting of measurements of the velocities of 82 galaxies in the Corona Borealis region. To facilitate the comparison with RJMCMC, we use exactly the same model as Richardson and Green (1997). The component densities are univariate normal, $f_\theta(x) = f_{\mu,\lambda}(x) = \mathcal{N}(x|\mu, \lambda^{-1})$, and the base measure H on $\theta = (\mu, \lambda)$ is $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, $\lambda \sim \text{Gamma}(a, b)$ independently (where $\text{Gamma}(\lambda|a, b) \propto \lambda^{a-1}e^{-b\lambda}$). Further, a hyperprior is placed on b, by taking $b \sim \text{Gamma}(a_0, b_0)$. The remaining parameters are set to $\mu_0 = (\max\{x_i\} + \min\{x_i\})/2$, $\sigma_0 = \max\{x_i\} - \min\{x_i\}$, $a = 2$, $a_0 = 0.2$, and $b_0 = 10/\sigma_0^2$. Note that the parameters μ_0, σ_0, and b_0 are functions of the observed data $x_1, \ldots, x_n$. See Richardson and Green (1997) for the rationale behind these parameter choices. (Note: This choice of σ_0 may be a bit too large, affecting the posteriors on the number of clusters and components, however, we stick with it to facilitate the comparison.) Following Richardson and Green (1997), we take $K \sim \text{Uniform}\{1, \ldots, 30\}$ and $\gamma = 1$.

As in Section 7.1, we use the split-merge sampler of Jain and Neal (2007) for non-conjugate priors, coupled with Algorithm 8 of Neal (2000). We use Gibbs sampling to handle the hyperprior on b (i.e., append b to the state of the Markov chain, run the sampler given b as usual, and periodically sample b given everything else). More general hyperprior structures can be handled similarly. In all other respects, the same inference algorithm as in Section 7.1 is used. We do not restrict the parameter space in any way (e.g., forcing the component means to be ordered to obtain identifiability, as was done by Richardson and Green, 1997). All of the quantities we consider are invariant to the labeling of the clusters; see Jasra et al. (2005) for discussion on this point. Results for RJMCMC were obtained using the Nmix software provided by Peter Green,[1] which implements the algorithm of Richardson and Green (1997).

Results

First, we demonstrate agreement between results from RJMCMC and our method based on the Jain–Neal algorithm. Table 1 compares estimates of the MFM posterior on the number of components k using both Jain–Neal and RJMCMC. Each algorithm was run for 10^6 iterations, the first 10^5 of which were discarded as burn-in. The two methods are in

[1] Currently available at `http://www.maths.bris.ac.uk/~peter/Nmix/`.

very close agreement, empirically verifying that they are both correctly sampling from the MFM posterior.

Table 1: Estimates of the MFM posterior on k for the galaxy dataset.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Jain–Neal	0.000	0.000	0.060	0.134	0.187	0.194	0.158	0.110	0.069	0.040	0.023	0.012	0.007	0.004	0.002
RJMCMC	0.000	0.000	0.059	0.131	0.187	0.197	0.160	0.110	0.068	0.039	0.022	0.012	0.006	0.003	0.002

To compare the mixing performance of the two methods, Figure 5 (top) shows traceplots of the number of clusters over the first 5000 iterations. The number of clusters is often used to assess mixing, since it is usually one of the quantities for which mixing is slowest. One can see that the Jain–Neal split-merge algorithm appears to explore the space more quickly than RJMCMC. Figure 5 (bottom left) shows estimates of the autocorrelation functions for the number of clusters, scaled by iteration. The autocorrelation of Jain–Neal decays significantly more rapidly than that of RJMCMC, confirming the visual intuition from the traceplot. This makes sense since the Jain–Neal algorithm makes splitting proposals in an adaptive, data-dependent way. The effective sample size (based on the number of clusters) is estimated to be 1.6% for Jain–Neal versus 0.6% for RJMCMC; in other words, 100 Jain–Neal samples are equivalent to 1.6 independent samples (Kass et al., 1998).

However, each iteration of the Jain–Neal algorithm takes approximately $2.9 \times 10^{-6}\, n$ seconds, where $n = 82$ is the sample size, compared with $1.3 \times 10^{-6}\, n$ seconds for RJMCMC, which is a little over twice as fast. In both algorithms, one iteration consists of a reassignment move for each datapoint, updates to component parameters and hyperparameters, and a split or merge move, but it makes sense that Jain–Neal would be a bit slower per iteration since its reassignment moves and split-merge moves require a few more steps. Nonetheless, it seems that this is compensated for by the improvement in mixing per iteration: Figure 5 (bottom right) shows that when rescaled to account for computation time, the estimated autocorrelation functions are nearly the same. Thus, in this experiment, the two methods perform roughly equally well.

7.2.2 Comparison with RJMCMC as dimensionality increases

In this subsection, we show that our approach can provide a massive improvement over RJMCMC when the parameter space is high-dimensional. We illustrate this with a model that will be applied to gene expression data in Section 7.3.

We simulate data of increasing dimensionality, and assess the effect on mixing performance. Specifically, for a given dimensionality d, we draw $X_1, \ldots, X_n \sim \frac{1}{3}\mathcal{N}(m, I) + \frac{1}{3}\mathcal{N}(0, I) + \frac{1}{3}\mathcal{N}(-m, I)$, where $m = (3/\sqrt{d}, \ldots, 3/\sqrt{d}) \in \mathbb{R}^d$, and then normalize each dimension to have zero mean and unit variance. The purpose of the $1/\sqrt{d}$ factor is to prevent

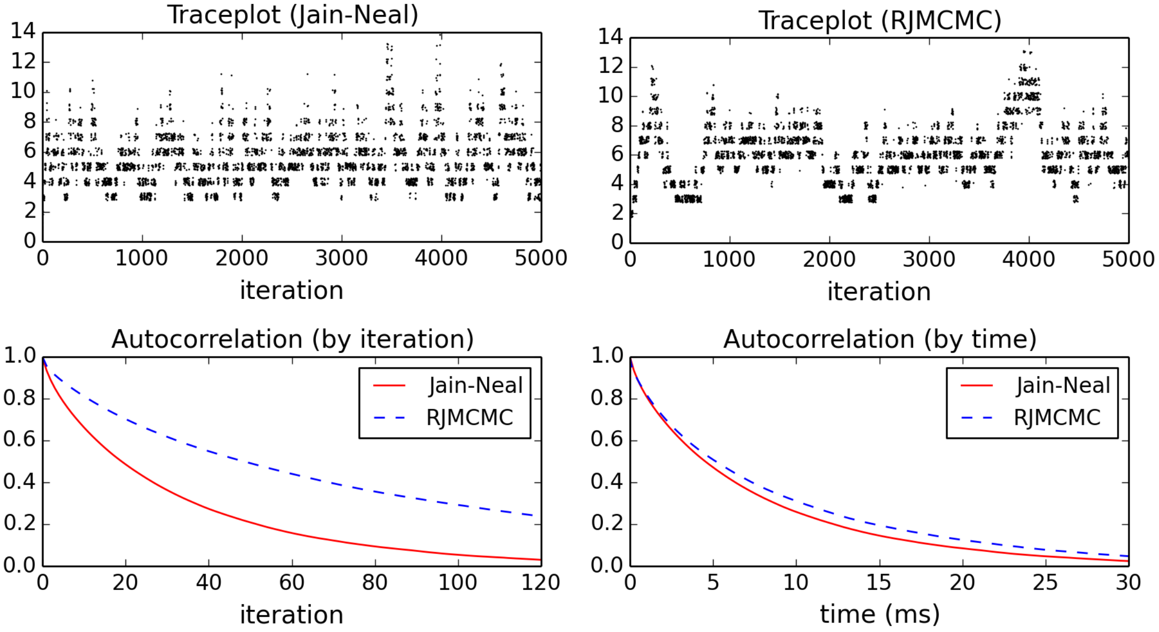

Figure 5: Results on the galaxy dataset. Top: Traceplots of the number of clusters at each iteration. Bottom: Estimated autocorrelation function of the number of clusters, scaled by iteration (left) and time in milliseconds (right).

the posterior from concentrating too quickly as d increases, so that one can still see the sampler exploring the space.

To enable scaling to the high-dimensional gene expression data in Section 7.3, we use multivariate Gaussian component densities with diagonal covariance matrices (i.e., for each component, the dimensions are independent univariate Gaussians), and we place independent conjugate priors on each dimension. Specifically, for each component, for $i = 1, \ldots, d$, dimension i is $\mathcal{N}(\mu_i, \lambda_i^{-1})$ given (μ_i, λ_i), and $\lambda_i \sim \text{Gamma}(a, b)$, $\mu_i | \lambda_i \sim \mathcal{N}(0, (c\lambda_i)^{-1})$. We set $a = 1$, $b = 1$, and $c = 1$ (recall that the data are zero mean, unit variance in each dimension), $K \sim \text{Geometric}(0.1)$, and $\gamma = 1$.

We compare the mixing performance of three samplers:

1. *Collapsed Jain–Neal.* Given the partition $\mathcal{C}$ of the data into clusters, the parameters can be integrated out analytically since the prior is conjugate. Thus, we can use the split-merge sampler of Jain and Neal (2004) for conjugate priors, coupled with Algorithm 3 of Neal (2000). Following Jain and Neal (2004), we use the (5,1,1) scheme: 5 intermediate scans to reach the split launch state, 1 split-merge move per iteration, and 1 incremental Gibbs scan per iteration.

2. *Jain–Neal.* Even though the prior is conjugate, we can still use the Jain and Neal (2007) split-merge algorithm for non-conjugate priors, coupled with Algorithm 8 of Neal (2000). As in Section 7.1, we use the (5,1,1,5) scheme.

3. *RJMCMC.* For reversible jump MCMC, we use a natural multivariate extension of the moves from Richardson and Green (1997). Specifically, for the split move, we propose

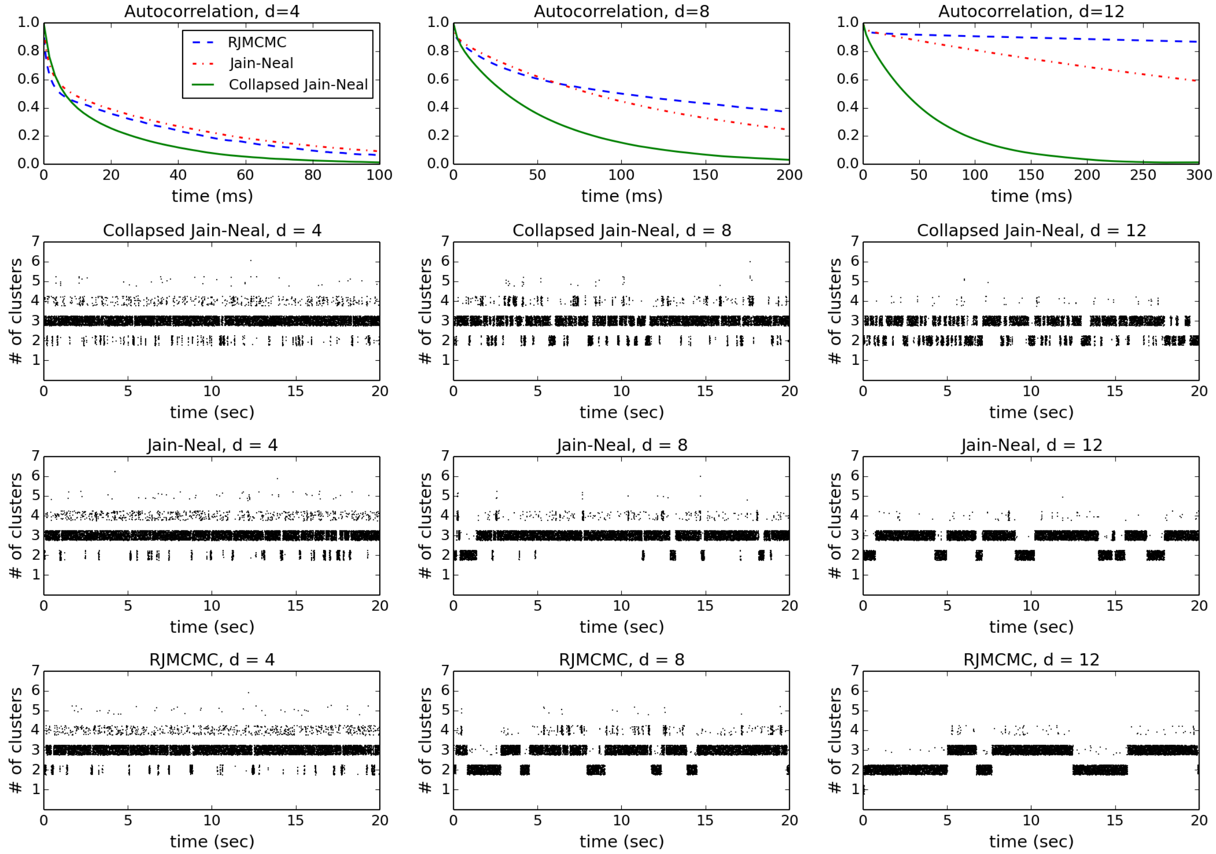

Figure 6: Top: Estimated autocorrelation functions for the number of clusters, for each algorithm, for each $d \in \{4, 8, 12\}$. Bottom: Traceplots of the number of clusters over the first 20 seconds of sampling.

splitting the mean and variance for each dimension i according to the formulas in Richardson and Green (1997), using independent variables u_{2i} and u_{3i}. For the birth move, we make a proposal from the prior. The remaining moves are essentially the same as in Richardson and Green (1997).

Results

For each $d \in \{4, 8, 12\}$, a dataset of size $n = 100$ was generated as described above, and each of the three algorithms was run for 10^6 iterations. For each algorithm and each d, Figure 6 shows the estimated autocorrelation function for the number of clusters (scaled to milliseconds), as well as a traceplot of the number of clusters over the first 20 seconds of sampling. We see that as the dimensionality increases, mixing performance degrades for all three algorithms, which is to be expected. However, it degrades significantly more rapidly for RJMCMC than for the Jain–Neal algorithms, especially the collapsed version. This difference can be attributed to the clever data-dependent split-merge proposals in the Jain–Neal algorithms, and the fact that the collapsed version benefits further from the effective reduction in dimensionality resulting from integrating out the parameters.

As the dimensionality increases, the advantage over RJMCMC becomes ever greater. This demonstrates the major improvement in performance that can be obtained using the approach we propose. In fact, in the next section, we apply our method to very high-dimensionality gene expression data on which the time required for RJMCMC to mix becomes unreasonably long, making it impractical. Meanwhile, our approach using the collapsed Jain–Neal algorithm still mixes well.

7.3 Discriminating cancer subtypes using gene expression data

In cancer research, gene expression profiling—that is, measuring the amount that each gene is expressed in a given tissue sample—enables the identification of distinct subtypes of cancer, leading to greater understanding of the mechanisms underlying cancers as well as potentially providing patient-specific diagnostic tools. In gene expression datasets, there are typically a small number of very high-dimensional data points, each consisting of the gene expression levels in a given tissue sample under given conditions.

One approach to analyzing gene expression data is to use Gaussian mixture models to identify clusters which may represent distinct cancer subtypes (Yeung et al., 2001; McLachlan et al., 2002; Medvedovic and Sivaganesan, 2002; Medvedovic et al., 2004; de Souto et al., 2008; Rasmussen et al., 2009; McNicholas and Murphy, 2010). In fact, in a comparative study of seven clustering methods on 35 cancer gene expression datasets with known ground truth, de Souto et al. (2008) found that finite mixtures of Gaussians provided the best results, as long as the number of components k was set to the true value. However, in practice, choosing an appropriate value of k can be difficult. Using the methods developed in this paper, the MFM provides a principled approach to inferring the clusters even when k is unknown, as well as doing inference for k, provided that the components are well-modeled by Gaussians.

The purpose of this example is to demonstrate that our approach can work well even in very high-dimensional settings, and may provide a useful tool for this application. It should be emphasized that we are not cancer scientists, so the results reported here should not be interpreted as scientifically relevant, but simply as a proof-of-concept.

7.3.1 Setup: data, model, and inference

We apply the MFM to gene expression data collected by Armstrong et al. (2001) in a study of leukemia subtypes. Armstrong et al. (2001) measured gene expression levels in samples from 72 patients who were known to have one of two leukemia types, acute lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML), and they found that a previously undistinguished subtype of ALL, which they termed mixed-lineage leukemia (MLL), could be distinguished from conventional ALL and AML based on the gene expression profiles.

We use the preprocessed data provided by de Souto et al. (2008), which they filtered to include only genes with expression levels differing by at least 3-fold in at least 30 samples, relative to their mean expression level across all samples. The resulting dataset consists of 72 samples and 1081 genes per sample, i.e., $n = 72$ and $d = 1081$. Following standard practice, we take the base-2 logarithm of the data before analysis, and normalize each

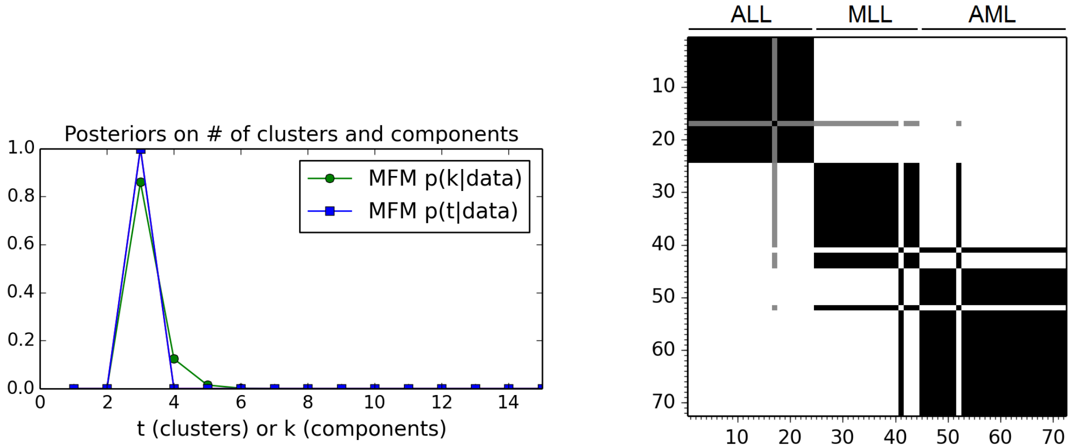

Figure 7: Results on the leukemia gene expression dataset. Left: Posteriors on the number of clusters and components. Right: Posterior similarity matrix. See the text for discussion.

dimension to have zero mean and unit variance.

We use the same model as in Section 7.2.2. Note that these parameter settings are simply defaults and have not been tailored to the problem; a careful scientific investigation would involve thorough prior elicitation, sensitivity analysis, and model checking.

The collapsed Jain–Neal algorithm described in Section 7.2.2 was used for inference. The sampler was run for 1,000 burn-in iterations, and 19,000 sample iterations. This appears to be many more iterations than required for burn-in and mixing in this particular example—in fact, only 5 to 10 iterations are required to separate the clusters, and the results are indistinguishable from using only 10 burn-in and 190 sample iterations. The full state of the chain was recorded every 20 iterations. Each iteration took approximately $1.3 \times 10^{-3}\, n$ seconds, with $n = 72$.

7.3.2 Results

The posterior on the number of clusters t is concentrated at 3 (see Figure 7), in agreement with the division into ALL, MLL, and AML determined by Armstrong et al. (2001). The posterior on k is shifted slightly to the right because there are a small number of observations; this accounts for uncertainty regarding the possibility of additional components that were not observed in the given data.

Figure 7 also shows the posterior similarity matrix, that is, the matrix in which entry (i, j) is the posterior probability that data points i and j belong to the same cluster; in the figure, white is probability 0 and black is probability 1. The rows and columns of the matrix are ordered according to ground truth, such that 1-24 are ALL, 25-44 are MLL, and 45-72 are AML. The model has clearly separated the subjects into these three groups, with a small number of exceptions: subject 41 is clustered with the AML subjects instead of MLL, subject 52 with the MLL subjects instead of AML, and subject 17 is about 50% ALL and 50% MLL. Thus, the MFM has successfully clustered the subjects according to cancer subtype. This demonstrates the viability of our approach in high-dimensional settings.

A Proofs

Proof of Theorem 3.1. Note that Equation A.1 below is well-known (see, e.g., Green and Richardson (2001) or McCullagh and Yang (2008)); we derive it here for completeness. Letting $E_i = \{j : z_j = i\}$, and writing $\mathcal{C}(z)$ for the partition induced by $z = (z_1, \ldots, z_n)$, by Dirichlet-multinomial conjugacy we have

$$p(z|k) = \int p(z|\pi)p(\pi|k)d\pi = \frac{\Gamma(k\gamma)}{\Gamma(\gamma)^k}\frac{\prod_{i=1}^k \Gamma(|E_i| + \gamma)}{\Gamma(n + k\gamma)} = \frac{1}{(k\gamma)^{(n)}}\prod_{c\in\mathcal{C}(z)} \gamma^{(|c|)},$$

for $z \in [k]^n$, provided that $p_K(k) > 0$. Recall that $x^{(m)} = x(x + 1)\cdots(x + m - 1)$ and $x_{(m)} = x(x-1)\cdots(x-m+1)$, with $x^{(0)} = 1$ and $x_{(0)} = 1$ by convention; note that $x_{(m)} = 0$ when x is a nonnegative integer less than m. It follows that for any partition $\mathcal{C}$ of $[n]$,

$$\begin{aligned} p(\mathcal{C}|k) &= \sum_{z\in[k]^n\,:\,\mathcal{C}(z)=\mathcal{C}} p(z|k) \\ &= \#\Big\{z \in [k]^n : \mathcal{C}(z) = \mathcal{C}\Big\}\,\frac{1}{(\gamma k)^{(n)}}\prod_{c\in\mathcal{C}} \gamma^{(|c|)} \\ &= \frac{k_{(t)}}{(\gamma k)^{(n)}}\prod_{c\in\mathcal{C}} \gamma^{(|c|)}, \end{aligned} \tag{A.1}$$

where $t = |\mathcal{C}|$, since $\#\big\{z \in [k]^n : \mathcal{C}(z) = \mathcal{C}\big\} = \binom{k}{t}t! = k_{(t)}$. Finally,

$$p(\mathcal{C}) = \sum_{k=1}^\infty p(\mathcal{C}|k)p_K(k) = \Big(\prod_{c\in\mathcal{C}} \gamma^{(|c|)}\Big)\sum_{k=1}^\infty \frac{k_{(t)}}{(\gamma k)^{(n)}}\,p_K(k) = V_n(t)\prod_{c\in\mathcal{C}} \gamma^{(|c|)},$$

with $V_n(t)$ as in Equation 3.2. □

Proof of Equation 3.4. Theorem 3.1 shows that the distribution of $\mathcal{C}$ is as shown. Next, note that instead of sampling only $\theta_1, \ldots, \theta_k \overset{\text{iid}}{\sim} H$ given $K = k$, we could simply sample $\theta_1, \theta_2, \ldots \overset{\text{iid}}{\sim} H$ independently of K, and the distribution of $X_{1:n}$ would be the same. Now, $Z_{1:n}$ determines which subset of the i.i.d. variables $\theta_1, \theta_2, \ldots$ will actually be used, and the indices of this subset are independent of $\theta_1, \theta_2, \ldots$; hence, denoting these random indices $I_1 < \cdots < I_T$, we have that $\theta_{I_1}, \ldots, \theta_{I_T}|Z_{1:n}$ are i.i.d. from H. For $c \in \mathcal{C}$, let $\phi_c = \theta_{I_i}$ where i is such that $c = \{j : z_j = I_i\}$. This completes the proof. □

Proof of the properties in Section 3.1. Abbreviate $x = x_{1:n}$, $z = z_{1:n}$, and $\theta = \theta_{1:k}$, and assume $p(z, k) > 0$. Letting $E_i = \{j : z_j = i\}$, we have $p(x|\theta, z, k) = \prod_{i=1}^k \prod_{j\in E_i} f_{\theta_i}(x_j)$

and

$$p(x|z,k) = \int_{\Theta^k} p(x|\theta,z,k)p(d\theta|k) = \prod_{i=1}^{k} \int_{\Theta} \Big[\prod_{j\in E_i} f_{\theta_i}(x_j)\Big] H(d\theta_i)$$
$$= \prod_{i=1}^{k} m(x_{E_i}) = \prod_{c\in\mathcal{C}(z)} m(x_c).$$

Since this last expression depends only on z, k through $\mathcal{C} = \mathcal{C}(z)$, we have $p(x|\mathcal{C}) = \prod_{c\in\mathcal{C}} m(x_c)$, establishing Equation 3.5. Next, recall that $p(\mathcal{C}|k) = \frac{k_{(t)}}{(\gamma k)^{(n)}} \prod_{c\in\mathcal{C}} \gamma^{(|c|)}$ (where $t = |\mathcal{C}|$) from Equation A.1, and thus

$$p(t|k) = \sum_{\mathcal{C}:|\mathcal{C}|=t} p(\mathcal{C}|k) = \frac{k_{(t)}}{(\gamma k)^{(n)}} \sum_{\mathcal{C}:|\mathcal{C}|=t} \prod_{c\in\mathcal{C}} \gamma^{(|c|)},$$

(where the sum is over partitions $\mathcal{C}$ of $[n]$ such that $|\mathcal{C}| = t$) establishing Equation 3.6. Equation 3.7 follows, since

$$p(k|t) \propto p(t|k)p(k) \propto \frac{k_{(t)}}{(\gamma k)^{(n)}} p_K(k),$$

(provided $p(t) > 0$) and the normalizing constant is precisely $V_n(t)$. To see that $\mathcal{C} \perp K \mid T$ (Equation 3.8), note that if $t = |\mathcal{C}|$ then

$$p(\mathcal{C}|t,k) = \frac{p(\mathcal{C},t|k)}{p(t|k)} = \frac{p(\mathcal{C}|k)}{p(t|k)},$$

(provided $p(t,k) > 0$) and due to the form of $p(\mathcal{C}|k)$ and $p(t|k)$ just above, this quantity does not depend on k; hence, $p(\mathcal{C}|t,k) = p(\mathcal{C}|t)$. To see that $X \perp K \mid T$ (Equation 3.9), note that $X \perp K \mid \mathcal{C}$; using this in addition to $\mathcal{C} \perp K \mid T$, we have

$$p(x|t,k) = \sum_{\mathcal{C}:|\mathcal{C}|=t} p(x|\mathcal{C},t,k)p(\mathcal{C}|t,k) = \sum_{\mathcal{C}:|\mathcal{C}|=t} p(x|\mathcal{C},t)p(\mathcal{C}|t) = p(x|t).$$

□

Proof of Theorem 4.1. Let $\mathcal{C}_\infty$ be the random partition of $\mathbb{Z}_{>0}$ as in Section 3.3, and for $n \in \{1,2,\dots\}$, let $\mathcal{C}_n$ be the partition of $[n]$ induced by $\mathcal{C}_\infty$. Then

$$p(\mathcal{C}_n|\mathcal{C}_{n-1},\dots,\mathcal{C}_1) = p(\mathcal{C}_n|\mathcal{C}_{n-1}) \propto q_n(\mathcal{C}_n)\, I(\mathcal{C}_n \setminus n = \mathcal{C}_{n-1}),$$

where $\mathcal{C}\setminus n$ denotes $\mathcal{C}$ with element n removed, and $I(\cdot)$ is the indicator function ($I(E) = 1$ if E is true, and $I(E) = 0$ otherwise). Recalling that $q_n(\mathcal{C}_n) = V_n(|\mathcal{C}_n|)\prod_{c\in\mathcal{C}_n} \gamma^{(|c|)}$ (Equation 3.1), we have, letting $t = |\mathcal{C}_{n-1}|$,

$$p(\mathcal{C}_n|\mathcal{C}_{n-1}) \propto \begin{cases} V_n(t+1)\gamma & \text{if } n \text{ is a singleton in } \mathcal{C}_n\text{, i.e., } \{n\} \in \mathcal{C}_n \\ V_n(t)(\gamma + |c|) & \text{if } c \in \mathcal{C}_{n-1} \text{ and } c \cup \{n\} \in \mathcal{C}_n, \end{cases}$$

for $\mathcal{C}_n$ such that $\mathcal{C}_n \setminus n = \mathcal{C}_{n-1}$ (and $p(\mathcal{C}_n|\mathcal{C}_{n-1}) = 0$ otherwise). With probability 1, $q_{n-1}(\mathcal{C}_{n-1}) > 0$, thus $V_{n-1}(t) > 0$ and hence also $V_n(t) > 0$, so we can divide through by $V_n(t)$ to get the result. □

Proof of Theorem 4.2. Let $G \sim \mathcal{M}(p_K, \gamma, H)$ and let $\beta_1, \ldots, \beta_n \overset{\text{iid}}{\sim} G$, given G. Then the joint distribution of $(\beta_1, \ldots, \beta_n)$ (with G marginalized out) is the same as $(\theta_{Z_1}, \ldots, \theta_{Z_n})$ in the original model (Equation 2.1). Let $\mathcal{C}_n$ denote the partition induced by $Z_1, \ldots, Z_n$ as usual, and for $c \in \mathcal{C}_n$, define $\phi_c = \theta_I$ where I is such that $c = \{j : Z_j = I\}$; then, as in the proof of Equation 3.4, $(\phi_c : c \in \mathcal{C}_n)$ are i.i.d. from H, given $\mathcal{C}_n$.

Therefore, we have the following equivalent construction for $(\beta_1, \ldots, \beta_n)$:

$$\begin{aligned}
&\mathcal{C}_n \sim q_n, \text{ with } q_n \text{ as in Section 3.3}\\
&\phi_c \overset{\text{iid}}{\sim} H \text{ for } c \in \mathcal{C}_n, \text{ given } \mathcal{C}_n\\
&\beta_j = \phi_c \text{ for } j \in c,\ c \in \mathcal{C}_n, \text{ given } \mathcal{C}_n, \phi.
\end{aligned}$$

Due to the self-consistency property of $q_1, q_2, \ldots$ (Proposition 3.3), we can sample $\mathcal{C}_n, (\phi_c : c \in \mathcal{C}_n), \beta_{1:n}$ sequentially for $n = 1, 2, \ldots$ by sampling from the restaurant process for $\mathcal{C}_n | \mathcal{C}_{n-1}$, sampling $\phi_{\{n\}}$ from H if n is placed in a cluster by itself (or setting $\phi_{c \cup \{n\}} = \phi_c$ if n is added to $c \in \mathcal{C}_{n-1}$), and setting β_n accordingly.

In particular, if the base measure H is continuous, then the ϕ's are distinct with probability 1, so conditioning on $\beta_{1:n-1}$ is the same as conditioning on $\mathcal{C}_{n-1}, (\phi_c : c \in \mathcal{C}_{n-1}), \beta_{1:n-1}$, and hence we can sample $\beta_n | \beta_{1:n-1}$ in the same way as was just described. In view of the form of the restaurant process (Theorem 4.1), the result follows. □

We use the following elementary result in the proof of Theorem 5.1; it is a special case of the dominated convergence theorem.

Proposition A.1. *For $j = 1, 2, \ldots$, let $a_{1j} \geq a_{2j} \geq \cdots \geq 0$ such that $a_{ij} \to 0$ as $i \to \infty$. If $\sum_{j=1}^{\infty} a_{1j} < \infty$ then $\sum_{j=1}^{\infty} a_{ij} \to 0$ as $i \to \infty$.*

Proof of Theorem 5.1. For any $x > 0$, writing $x^{(n)}/n! = \Gamma(x+n)/(n!\,\Gamma(x))$ and using Stirling's approximation, we have

$$\frac{x^{(n)}}{n!} \sim \frac{n^{x-1}}{\Gamma(x)}$$

as $n \to \infty$. Therefore, the $k = t$ term of $V_n(t)$ (Equation 3.2) is

$$\frac{t_{(t)}}{(\gamma t)^{(n)}}\, p_K(t) \sim \frac{t!}{n!}\, \frac{\Gamma(\gamma t)}{n^{\gamma t - 1}}\, p_K(t).$$

The first $t-1$ terms of $V_n(t)$ are 0, so to prove the result, we need to show that the rest of the series, divided by the $k = t$ term, goes to 0. (Recall that we have assumed $p_K(t) > 0$.) To this end, let

$$b_{nk} = (\gamma t)^{(n)} \frac{k_{(t)}}{(\gamma k)^{(n)}}\, p_K(k).$$

We must show that $\sum_{k=t+1}^{\infty} b_{nk} \to 0$ as $n \to \infty$. We apply Proposition A.1 with $a_{ij} = b_{t+i,t+j}$. For any $k > t$, $b_{1k} \geq b_{2k} \geq \cdots \geq 0$. Further, for any $k > t$,

$$\frac{(\gamma t)^{(n)}}{(\gamma k)^{(n)}} \sim \frac{n^{\gamma t - 1}}{\Gamma(\gamma t)}\, \frac{\Gamma(\gamma k)}{n^{\gamma k - 1}} \longrightarrow 0$$

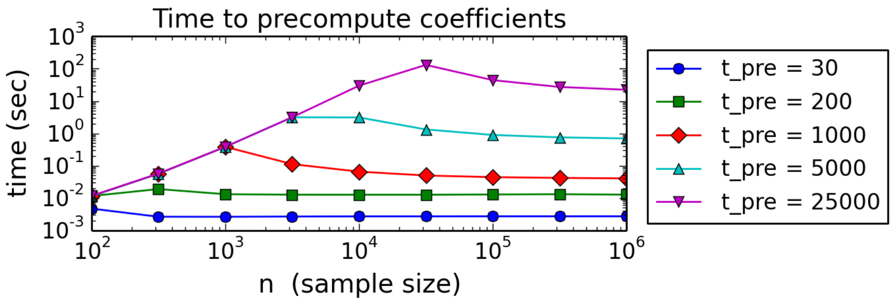

Figure 8: Amount of time required to precompute the MFM coefficients $V_n(1), \ldots, V_n(t_{\mathrm{pre}})$ for various values of t_{pre}, for increasing n.

as $n \to \infty$, hence, $b_{nk} \to 0$ as $n \to \infty$ (for any $k > t$). Finally, observe that $\sum_{k=t+1}^{\infty} b_{nk} \leq (\gamma t)^{(n)} V_n(t) < \infty$ for any $n \geq t$. Therefore, by Proposition A.1, $\sum_{k=t+1}^{\infty} b_{nk} \to 0$ as $n \to \infty$. This proves the result. □

Proof of Theorem 5.2. For any $t \in \{1, \ldots, k\}$,

$$p_n(K = t \mid T = t) = \frac{1}{V_n(t)} \frac{t_{(t)}}{(\gamma t)^{(n)}} p_K(t) \longrightarrow 1 \tag{A.2}$$

as $n \to \infty$ (where p_n denotes the MFM distribution with n samples), by Equation 3.7 and Theorem 5.1. For any $n \geq k$,

$$p(K = k \mid x_{1:n}) = \sum_{t=1}^{k} p(K = k \mid T = t, x_{1:n})\, p(T = t \mid x_{1:n}),$$

and note that by Equations 3.9 and A.2, $p(K = k \mid T = t, x_{1:n}) = p_n(K = k \mid T = t) \longrightarrow I(k = t)$ for $t \leq k$. The result follows. □

B Miscellanea

B.1 Time required to precompute the MFM coefficients

In all of the empirical demonstrations in this paper, the largest value of t visited by the sampler was less than 30. Thus, in each case it was sufficient to precompute $V_n(t)$ for $t = 1, \ldots, 30$, and reuse these values throughout MCMC sampling.

To see how long this precomputation would take if the sample size n and/or the number of clusters were much larger, Figure 8 shows the amount of time required to compute $V_n(t)$ for $t = 1, \ldots, t_{\mathrm{pre}}$, for each $t_{\mathrm{pre}} \in \{30, 200, 1000, 5000, 25000\}$, for increasing values of n, when $K \sim \text{Geometric}(0.1)$ and $\gamma = 1$. For $t_{\mathrm{pre}} = 30$ it only takes around 0.001 seconds for any n. For much larger values of t_{pre} it takes longer, but the time required relative to MCMC sampling would still be negligible. The reason why the computation time decreases as n grows past t_{pre} is that, as discussed in Section 3.2, the infinite series for $V_n(t)$ (Equation 3.2) converges more rapidly when n is much bigger than t.

B.2 Formulas for some posterior quantities

Posterior on the number of components k

The posterior on $t = |\mathcal{C}|$ is easily estimated from posterior samples of $\mathcal{C}$. To compute the MFM posterior on k, note that

$$p(k|x_{1:n}) = \sum_{t=1}^{\infty} p(k|t, x_{1:n})p(t|x_{1:n}) = \sum_{t=1}^{n} p(k|t)p(t|x_{1:n}),$$

by Equation 3.9 and the fact that t cannot exceed n. Using this and the formula for $p(k|t)$ given by Equation 3.7, it is simple to transform our estimate of the posterior on t into an estimate of the posterior on k. For the DPM, the posterior on the number of components k is always trivially a point mass at infinity.

Density estimates

Using the restaurant process (Theorem 4.1), it is straightforward to show that if $\mathcal{C}$ is a partition of $[n]$ and $\phi = (\phi_c : c \in \mathcal{C})$ then

$$p(x_{n+1} \mid \mathcal{C}, \phi, x_{1:n}) \propto \frac{V_{n+1}(t+1)}{V_{n+1}(t)}\gamma\, m(x_{n+1}) + \sum_{c\in\mathcal{C}}(|c|+\gamma) f_{\phi_c}(x_{n+1}) \tag{B.1}$$

where $t = |\mathcal{C}|$, and, using the recursion for $V_n(t)$ (Equation 3.10), this is normalized when multiplied by $V_{n+1}(t)/V_n(t)$. Further,

$$p(x_{n+1} \mid \mathcal{C}, x_{1:n}) \propto \frac{V_{n+1}(t+1)}{V_{n+1}(t)}\gamma\, m(x_{n+1}) + \sum_{c\in\mathcal{C}}(|c|+\gamma)\frac{m(x_{c\cup\{n+1\}})}{m(x_c)}, \tag{B.2}$$

with the same normalization constant. Therefore, when $m(x_c)$ can be easily computed, Equation B.2 can be used to estimate the posterior predictive density $p(x_{n+1}|x_{1:n})$ based on samples from $\mathcal{C} \mid x_{1:n}$. When $m(x_c)$ cannot be easily computed, Equation B.1 can be used to estimate $p(x_{n+1}|x_{1:n})$ based on samples from $\mathcal{C}, \phi \mid x_{1:n}$, along with samples $\theta_1, \ldots, \theta_N \overset{\text{iid}}{\sim} H$ to approximate $m(x_{n+1}) \approx \frac{1}{N}\sum_{i=1}^{N} f_{\theta_i}(x_{n+1})$.

The posterior predictive density is, perhaps, the most natural estimate of the density. However, following Green and Richardson (2001), a simpler way to obtain a natural estimate is by assuming that element $n+1$ is added to an existing cluster; this will be very similar to the posterior predictive density when n is sufficiently large. To this end, we define $p_*(x_{n+1} \mid \mathcal{C}, \phi, x_{1:n}) = p(x_{n+1} \mid \mathcal{C}, \phi, x_{1:n}, |\mathcal{C}_{n+1}| = |\mathcal{C}|)$, where $\mathcal{C}_{n+1}$ is the partition of $[n+1]$, and observe that

$$p_*(x_{n+1} \mid \mathcal{C}, \phi, x_{1:n}) = \sum_{c\in\mathcal{C}} \frac{|c|+\gamma}{n+\gamma t} f_{\phi_c}(x_{n+1})$$

where $t = |\mathcal{C}|$ (Green and Richardson, 2001). Using this, we can estimate the density by

$$\frac{1}{N}\sum_{i=1}^{N} p_*(x_{n+1} \mid \mathcal{C}^{(i)}, \phi^{(i)}, x_{1:n}), \tag{B.3}$$

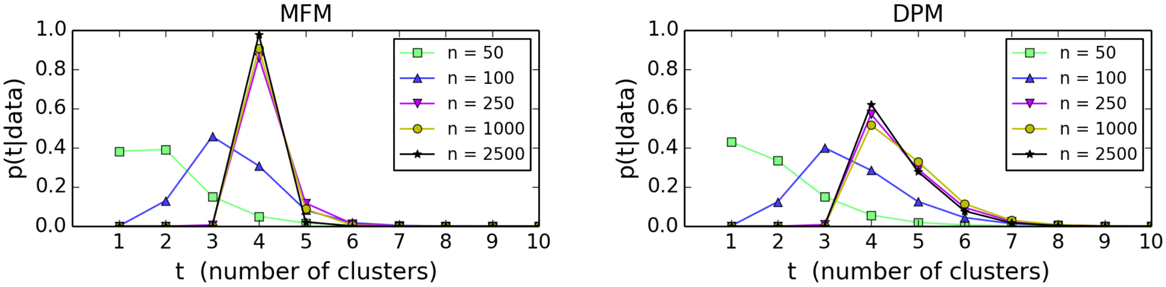

Figure 9: Posterior on the number of clusters t for the MFM and DPM on data from the modified bivariate example with a small fourth component.

where $(\mathcal{C}^{(1)}, \phi^{(1)}), \dots, (\mathcal{C}^{(N)}, \phi^{(N)})$ are samples from $\mathcal{C}, \phi \mid x_{1:n}$. The corresponding expressions for the DPM are all very similar, using its restaurant process instead. The density estimates shown in this paper are obtained using this approach.

These formulas are conditional on additional parameters such as γ for the MFM, and α for the DPM. If priors are placed on such parameters and they are sampled along with $\mathcal{C}$ and ϕ given $x_{1:n}$, then the posterior predictive density can be estimated using the same formulas as above, but also using the posterior samples of these additional parameters.

B.3 Small components

In Section 7.1, we noted that the DPM tends to introduce one or two tiny extra components, and it is natural to wonder whether the DPM would fare better if the data were actually drawn from a mixture with an additional one or two small components. To see, we modify the data distribution from Section 7.1 to be a four-component mixture in which w_1 is reduced from 0.45 to 0.44, and the fourth component has weight $w_4 = 0.01$, mean $\mu_4 = \binom{8}{11}$, and covariance $C_4 = \left(\begin{smallmatrix} 0.1 & 0 \\ 0 & 0.1 \end{smallmatrix}\right)$. We use exactly the same model and inference parameters as in Section 7.1.

Figure 9 shows that the MFM still more accurately infers the number of clusters than the DPM. We expect that in order to have a situation where the DPM performs more favorably in terms of clustering and inferring the number of clusters, the number of components would have to be sufficiently large relative to the sample size, or infinite.

References

D. J. Aldous. *Exchangeability and related topics.* Springer, 1985.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, and S. J. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1):41–47, 2001.

D. Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, pages 260–279, 1992.

D. Blackwell and J. B. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, pages 353–355, 1973.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

T. Broderick, M. I. Jordan, and J. Pitman. Beta processes, stick-breaking and power laws. *Bayesian Analysis*, 7(2):439–476, 2012.

S. P. Brooks, P. Giudici, and G. O. Roberts. Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–39, 2003.

C. A. Bush and S. N. MacEachern. A semiparametric Bayesian model for randomised block designs. *Biometrika*, 83(2):275–285, 1996.

A. Cerquetti. Conditional α-diversity for exchangeable Gibbs partitions driven by the stable subordinator. *arXiv:1105.0892*, 2011.

A. Cerquetti et al. Marginals of multivariate Gibbs distributions with applications in Bayesian species sampling. *Electronic Journal of Statistics*, 7:697–716, 2013.

Y. Chung and D. B. Dunson. Nonparametric Bayes conditional distribution modeling with variable selection. *Journal of the American Statistical Association*, 104(488), 2009.

D. B. Dahl. An improved merge-split sampler for conjugate Dirichlet process mixture models. *Technical Report, Department of Statistics, University of Wisconsin – Madison*, 2003.

D. B. Dahl. Sequentially-allocated merge-split sampler for conjugate and nonconjugate Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 11, 2005.

D. B. Dahl. Modal clustering in a class of product partition models. *Bayesian Analysis*, 4 (2):243–264, 2009.

M. C. de Souto, I. G. Costa, D. S. de Araujo, T. B. Ludermir, and A. Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):497, 2008.

P. Dellaportas and I. Papageorgiou. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, 16(1):57–68, 2006.

D. B. Dunson and J.-H. Park. Kernel stick-breaking processes. *Biometrika*, 95(2):307–323, 2008.

R. Durrett. *Probability: Theory and Examples*, volume 2. Cambridge University Press, 1996.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

S. Favaro, A. Lijoi, and I. Pruenster. On the stick-breaking representation of normalized inverse Gaussian priors. *Biometrika*, 99(3):663–674, 2012.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

S. Ghosal and A. Van der Vaart. Posterior convergence rates of Dirichlet mixtures at smooth densities. *The Annals of Statistics*, 35(2):697–723, 2007.

A. Gnedin. A species sampling model with finitely many types. *Elect. Comm. Probab.*, 15: 79–88, 2010.

A. Gnedin and J. Pitman. Exchangeable Gibbs partitions and Stirling triangles. *Journal of Mathematical Sciences*, 138(3):5674–5685, 2006.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

P. J. Green and S. Richardson. Modeling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics*, 28(2):355–375, June 2001.

J. E. Griffin and M. J. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101(473):179–194, 2006.

B. Hansen and J. Pitman. Prediction rules for exchangeable sequences related to species sampling. *Statistics & Probability Letters*, 46(3):251–256, 2000.

J. A. Hartigan. Partition models. *Communications in Statistics – Theory and Methods*, 19 (8):2745–2756, 1990.

D. I. Hastie and P. J. Green. Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338, 2012.

J. Henna. On estimating of the number of constituents of a finite mixture of continuous distributions. *Annals of the Institute of Statistical Mathematics*, 37(1):235–240, 1985.

J. Henna. Estimation of the number of components of finite mixtures of multivariate distributions. *Annals of the Institute of Statistical Mathematics*, 57(4):655–664, 2005.

N. L. Hjort. Bayesian analysis for a generalised Dirichlet process prior. *Technical Report, University of Oslo*, 2000.

M.-W. Ho, L. F. James, and J. W. Lau. Gibbs partitions (EPPF's) derived from a stable subordinator are Fox H and Meijer G transforms. *arXiv:0708.0619*, 2007.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453), 2001.

H. Ishwaran and L. F. James. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13(4):1211–1236, 2003.

H. Ishwaran and M. Zarepour. Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

H. Ishwaran, L. F. James, and J. Sun. Bayesian model selection in finite mixtures by marginal density decompositions. *Journal of the American Statistical Association*, 96 (456), 2001.

S. Jain and R. M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1), 2004.

S. Jain and R. M. Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2(3):445–472, 2007.

L. F. James, C. E. Priebe, and D. J. Marchette. Consistent estimation of mixture complexity. *The Annals of Statistics*, pages 1281–1296, 2001.

A. Jasra, C. Holmes, and D. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, pages 50–67, 2005.

M. Kalli, J. E. Griffin, and S. G. Walker. Slice sampling mixture models. *Statistics and Computing*, 21(1):93–105, 2011.

R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Markov chain Monte Carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

C. Keribin. Consistent estimation of the order of mixture models. *Sankhya Ser. A*, 62(1): 49–66, 2000.

W. Kruijer, J. Rousseau, and A. Van der Vaart. Adaptive Bayesian density estimation with location-scale mixtures. *Electronic Journal of Statistics*, 4:1225–1257, 2010.

B. G. Leroux. Consistent estimation of a mixing distribution. *The Annals of Statistics*, 20 (3):1350–1360, 1992.

A. Lijoi and I. Prünster. Models beyond the Dirichlet process. *Bayesian Nonparametrics*, 28:80, 2010.

A. Lijoi, R. H. Mena, and I. Prünster. Hierarchical mixture modeling with normalized inverse-Gaussian priors. *Journal of the American Statistical Association*, 100(472):1278–1291, 2005.

A. Lijoi, R. H. Mena, and I. Prünster. Bayesian nonparametric estimation of the probability of discovering new species. *Biometrika*, 94(4):769–786, 2007.

A. Lijoi, I. Prünster, and S. G. Walker. Bayesian nonparametric estimators derived from conditional Gibbs structures. *The Annals of Applied Probability*, 18(4):1519–1547, 2008.

J. S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427): 958–966, 1994.

S. N. MacEachern. Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics – Simulation and Computation*, 23(3):727–741, 1994.

S. N. MacEachern. Computational methods for mixture of Dirichlet process models. In *Practical nonparametric and semiparametric Bayesian statistics*, pages 23–43. Springer, 1998.

S. N. MacEachern. Dependent nonparametric processes. In *ASA Proceedings of the Section on Bayesian Statistical Science*, pages 50–55, 1999.

S. N. MacEachern. Dependent Dirichlet processes. *Unpublished manuscript, Department of Statistics, The Ohio State University*, 2000.

S. N. MacEachern and P. Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

A. D. Marrs. An application of reversible-jump MCMC to multivariate spherical Gaussian mixtures. In *Advances in Neural Information Processing Systems*, pages 577–583, 1998.

P. McCullagh and J. Yang. How many clusters? *Bayesian Analysis*, 3(1):101–120, 2008.

G. J. McLachlan, R. W. Bean, and D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

P. D. McNicholas and T. B. Murphy. Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics*, 26(21):2705–2712, 2010.

M. Medvedovic and S. Sivaganesan. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18(9):1194–1206, 2002.

M. Medvedovic, K. Y. Yeung, and R. E. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232, 2004.

J. W. Miller and M. T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in Neural Information Processing Systems, Vol. 26*, 2013.

J. W. Miller and M. T. Harrison. Inconsistency of Pitman–Yor process mixtures for the number of components. *Journal of Machine Learning Research*, 15:3333–3370, 2014.

P. Müller and F. Quintana. Random partition models with regression on covariates. *Journal of Statistical Planning and Inference*, 140(10):2801–2808, 2010.

P. Müller, F. Quintana, and G. L. Rosner. A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics*, 20(1), 2011.

R. M. Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods*, pages 197–211. Springer, 1992.

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

X. L. Nguyen. Convergence of latent mixing measures in finite and infinite mixture models. *The Annals of Statistics*, 41(1):370–400, 2013.

A. Nobile. *Bayesian Analysis of Finite Mixture Distributions.* PhD thesis, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, 1994.

A. Nobile. Bayesian finite mixtures: a note on prior specification and posterior computation. *Technical Report, Department of Statistics, University of Glasgow*, 2005.

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

M. Pagel and A. Meade. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. *Systematic Biology*, 53(4):571–581, 2004.

J. W. Paisley, A. K. Zaas, C. W. Woods, G. S. Ginsburg, and L. Carin. A stick-breaking construction of the beta process. In *Proceedings of the 27th International Conference on Machine Learning*, pages 847–854, 2010.

O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008.

P. Papastamoulis and G. Iliopoulos. An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2), 2010.

J.-H. Park and D. B. Dunson. Bayesian generalized product partition model. *Statistica Sinica*, 20:1203–1226, 2010.

D. B. Phillips and A. F. M. Smith. Bayesian model comparison via jump diffusions. In *Markov chain Monte Carlo in Practice*, pages 215–239. Springer, 1996.

J. Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158, 1995.

J. Pitman. Some developments of the Blackwell-MacQueen urn scheme. *Lecture Notes-Monograph Series*, pages 245–267, 1996.

J. Pitman. *Combinatorial Stochastic Processes*. Springer–Verlag, Berlin, 2006.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

F. A. Quintana and P. L. Iglesias. Bayesian clustering and product partition models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):557–574, 2003.

C. E. Rasmussen, B. J. de la Cruz, Z. Ghahramani, and D. L. Wild. Modeling and visualizing uncertainty in gene expression clusters using Dirichlet process mixtures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4):615–628, 2009.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1):19–41, 2000.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

A. Rodriguez and D. B. Dunson. Nonparametric Bayesian models through probit stick-breaking processes. *Bayesian Analysis*, 6(1), 2011.

C. E. Rodríguez and S. G. Walker. Univariate Bayesian nonparametric mixture modeling with unimodal kernels. *Statistics and Computing*, 24(1):35–49, 2014.

K. Roeder. Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, 85(411):617–624, 1990.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

J. Sethuraman and R. C. Tiwari. Convergence of Dirichlet measures and the interpretation of their parameter. *Technical Report, Department of Statistics, Florida State University*, 1981.

C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999.

M. Stephens. Bayesian analysis of mixture models with an unknown number of components—An alternative to reversible jump methods. *The Annals of Statistics*, 28 (1):40–74, 2000.

Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, pages 556–563, 2007.

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, pages 564–571, 2007.

S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics – Simulation and Computation*, 36(1):45–54, 2007.

M. West. Hyperparameter estimation in Dirichlet process mixture models. *ISDS Discussion Paper #92-A03, Duke University*, 1992.

M. West, P. Müller, and M. D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In P. Freeman and A. F. Smith, editors, *Aspects of Uncertainty: A Tribute to D.V. Lindley*, pages 363–386. Wiley, 1994.

M.-J. Woo and T. N. Sriram. Robust estimation of mixture complexity. *Journal of the American Statistical Association*, 101(476), 2006.

M.-J. Woo and T. N. Sriram. Robust estimation of mixture complexity for count data. *Computational Statistics and Data Analysis*, 51(9):4379–4392, 2007.

K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

Z. Zhang, K. L. Chan, Y. Wu, and C. Chen. Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithm. *Statistics and Computing*, 14(4): 343–355, 2004.