Sri Lankan Journal of Applied Statistics (Special Issue)
Modern Statistical Methodologies in the Cutting Edge of Science

A New Method for Tracking Configuration for Dirichlet Process Sampling

Rui Wu[1], Ming-Hui Chen[2*], Lynn Kuo[2], and Paul O. Lewis[3]

[1] Novartis Pharmaceuticals Corporation, East Hanover, New Jersey, USA

[2] Department of Statistics, University of Connecticut, Storrs, Connecticut, USA

[3] Department of Ecology and Evolutionary Biology, University of Connecticut, Storrs, Connecticut, USA

*Corresponding Author: ming-hui.chen@uconn.edu

Received: 30, July 2013 / Revised: 28, January 2014 / Accepted: 10, February 2014

ABSTRACT

The method of fitting a hierarchical model with Dirichlet process mixing is a versatile tool for data analysts. It has been applied to density estimation, classification, clustering, and high dimensional data analysis. Many computing algorithms have been proposed to evaluate this mixture. Different labels in the algorithm that assign data points into clusters may actually yield the same partition configuration. This paper makes this notion rigorous by establishing an equivalence theorem. Thus, we would recommend adding the step of checking for equivalent configurations to the algorithms for evaluating hierarchical Dirichlet process mixing models for improved results, especially when cluster assignments are the major goals of the analysis.

Keywords: Clustering configuration, Dirichlet process, Hierarchical Dirichlet process mixing, MCMC algorithm.

1. Introduction

The Dirichlet process (DP) has received a lot of attention since its development by Ferguson (1973). It constructs a prior measure on a large space of distribution functions. It allows users to specify two parameters α and G_0, where distribution function G_0 reflects the prior mean ('center') of the unknown distribution G chosen by the DP and the positive real number α characterizes the prior precision. Larger α, stronger prior belief, indicates smaller variation of the unknown distribution G. Ferguson (1973) also showed that DP is intuitive and interpretable for some nonparametric statistical problems. Consequently, the Dirichlet process has been applied to many areas including text mining, machine learning, bioinformatics, biostatistics, and phylogenetics, where Bayesian nonparametric inference on an unknown distribution is called for.

Mixture of Dirichlet processes (MDP) originated from Antoniak (1974) as a further development from the DP, where a smoothing kernel with an unknown parameter θ is convolved with an unknown distribution chosen from a DP. Antoniak (1974) studied properties of this mixture and showed how it can be applied to measurement error, empirical Bayes, and quantal bioassay problems. Later Lo (1984) developed it further to construct rigorously a prior measure on a space of density functions. Kuo (1983) has extended MDP to a Bayesian experimental design problem to study the optimal dosages for quantal bioassay with the potency curve chosen from a DP. Kuo (1986a) has also applied MDP to empirical Bayes problems.

Suppose we observe a set of n observations $\mathbf{y} = (y_1, \ldots, y_n)$. We consider a hierarchical model where the ith observation is modeled by the density $f_i(y_i|\theta_i)$ with an unobserved latent variable θ_i. Then we assume the latent variables $\theta_1, \ldots, \theta_n$ are independent and identically distributed with a common unknown distribution G chosen from the $DP(\alpha, G_0)$. Consequently, Neal (2000) pointed out the name Dirichlet process mixture (DPM) model would be more appropriate. The Bayes solution to this mixture problem can be written concisely as in Kuo (1980, 1986a). However, the number of mixing components, which is given by the Bell exponential number (Berge, 1971), increases rapidly as the sample size increases due to the discrete nature of the DP. Therefore, many computing algorithms have been proposed to evaluate this mixture, such as Kuo (1986b), Escobar and West (1995, 1998), MacEachern and Müller (1998), Neal (2000), Ishwaran and James (2001), Walker (2007), Dunson (2010), and Kalli et al. (2010). All these algorithms need to specify the latent allocation process that assigns each observation to a cluster. The confusion comes from the fact that different allocations (usually designated by labels) in the algorithm may actually denote the same cluster. Overlooking this subtlety may lead to an incorrect conclusion. In this paper, we clarify this issue. We define equivalent classes of labels if they lead to the same partition of the latent parameter space. So we suggest adding the checking for equivalence step to the computing algorithm when correct clustering allocation is essential in the solution.

Section 2 contains basic formulations of DP and MDP. Section 3 lists two sampling algorithms for evaluating the MDP. One, called the truncated blocked Gibbs sampler, is based on Ishwaran and James (2001), where the number of atoms of the DP is truncated at a fixed number determined before running the sampler. The other is based on Kalli et al. (2010), where the number of atoms in the DP is determined from slice sampling in each iteration. Section 4 discusses the equivalence formulation which needs to be included in the sampling procedure for MDP, especially when the cluster allocation is of primary interest in the study. Section 5 contains a simulation study and a real data analysis to illustrate our method. We conclude the paper with brief discussion in Section 6.

2. Hierarchical Dirichlet Process Mixing

2.1. Introduction to the Dirichlet Process

The DP process denoted by $DP(\alpha, G_0)$ allows users to select two components: a base distribution G_0 that defines the center of the DP, and a concentration parameter α that reflects the prior strength of belief on the base distribution G_0. If G is taken as a sample from $DP(\alpha, G_0)$, then by construction, we have $E[G(A)] = G_0(A)$ and $\text{Var}(G(A)) = G_0(A)[1 - G_0(A)]/(\alpha + 1)$ for any measurable set A. If we think of the DP constructing a tube of random cumulative distribution function G centered around G_0, then the variance formula shows how α controls the variation of this tube in an inverse relation at a linear rate. Suppose the prior on a random distribution is $DP(\alpha, G_0)$, then Ferguson (1973) showed that the posterior distribution of this random distribution given a sample of size n (from this distribution) is also a DP with new parameters: a concentration parameter $\alpha + n$, and a base distribution $(\alpha G_0 + n\hat{F}_n)/(\alpha + n)$, where $\hat{F}_n$ is the empirical distribution function.

2.2. A Constructive Representation of the Dirichlet Process

In order to sample a random distribution from DP, it is helpful to know the constructive definition of the DP. There are several versions of it, which include the Pólya urn scheme by Blackwell and MacQueen (1973), the stick-breaking process by Sethuraman and Tiwari (1982) and Sethuraman (1994), and the Chinese restaurant process by Aldous (1985).

We are summarizing the Sethuraman (1994) construction here. Suppose G is a random sample chosen from $DP(\alpha, G_0)$. Then G can be represented by

$$G = \sum_{k=1}^{\infty} w_k \delta_{\phi_k}, \tag{2.1}$$

where δ_{ϕ_k} is a measure of mass 1 concentrated at ϕ_k and the locations ϕ_k of the random jumps of the distribution G are independent and identically distributed (i.i.d.) from G_0 with stick-breaking weights w_k given by the following formula:

$$w_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \tag{2.2}$$

where v_l are also i.i.d. from a beta density $\mathcal{B}e(1, \alpha)$.

So this representation shows that DP selects discrete distributions with probability 1. Each distribution has atoms chosen as a sample from G_0 and with weights defined as the stick-breaking weights given in (2.2).

2.3. Hierarchical Dirichlet Process Mixing Model

The hierarchical model of a compound decision problem starts with modeling each observation with an unobserved latent variable at the first stage. Then DP

is used as a prior over the common distribution shared by the latent parameters at the second stage:

$$\begin{aligned}
&\boldsymbol{y}_i|\boldsymbol{\theta}_i \overset{ind}{\sim} f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_i), \text{ for } i = 1, \ldots, n. \\
&\boldsymbol{\theta}_i|G \overset{i.i.d.}{\sim} G, \text{ for } i = 1, \ldots, n. \\
&G \sim DP(\alpha, G_0).
\end{aligned} \tag{2.3}$$

This model was first proposed and called MDP by Antoniak (1974). Instead, we will call it DPM (Dirichlet process mixture) as in Neal (2000) to emphasize the mixing is done by the DP construction.

The mixture model allows borrowing information across components. Its formulation is flexible and adaptable, yet quite efficient in combating the curse of dimensionality. It avoids specifying a fixed number of clusters for the latent population. It is an ideal candidate for clustering problems where the distinct number of clusters is unknown beforehand. Although the DP assumes that there are infinitely many clusters in the latent population, the posterior distribution for the number of clusters for the latent variable has a sparseness favoring structure due to the discrete nature of the DP. Suppose we have latent variables $\theta_1, \ldots, \theta_n$ in k clusters with $k \leq n$. When an $(n+1)$st latent variable is added, then the new θ_{n+1} is assigned to a new cluster with a probability of $\alpha/(\alpha+n)$, and to a previous cluster j with probability of $n_j/(\alpha+n)$, where n_j is the number of θs in the jth cluster. Note we have $\sum_{j=1}^{k} n_j = n$. The prior expected number of clusters for the latent variables of size n is proportional to $\alpha \log n$. Thus, the number of clusters *a priori* increases slowly with the sample size at a rate determined by α. The posterior distribution of the number of clusters not only depends on n and α, and is also sensitive to the choice of G_0 as pointed out by Dunson (2010). Therefore, he argued for the need to choose G_0 with careful thought; he also suggested to standardize the data first and specify G_0 with location zero and scale one in order to have good practical performance.

2.3.1. Bayes Solution

Typically, the objective is to estimate each of the latent parameters $\boldsymbol{\theta}_i$ for $i = 1, \ldots, n$. The above DPM setting allows multivariate observations and multivariate latent parameters for each component in the mixture model. However, in the following expression for the posterior mean of each latent variable, we will just use the univariate version for a more streamlined presentation. To derive the posterior mean of θ_i, we usually integrate out the random G. Following Antoniak (1974), the posterior distribution of G is a mixture of DP with concentration parameter $\alpha + n$ and base distribution $(\alpha G_0 + \sum_{j=1}^{n} \delta_{\theta_j})/(\alpha+n)$ and mixing distribution $H(\boldsymbol{\theta}|\mathbf{y})$. That is

$$G|\mathbf{y} \sim \int \mathcal{P}_{\alpha G_0 + \sum_{j=1}^{n} \delta_{\theta_j}}(dG) dH(\boldsymbol{\theta}|\mathbf{y}).$$

Let $H(\boldsymbol{\theta})$ denote the unconditional marginal distribution of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$, where

$$dH(\boldsymbol{\theta}) = \prod_{j=1}^{n} \frac{\alpha G_0 + \sum_{l=1}^{j-1} \delta_{\theta_l}}{\alpha + j - 1}(d\theta_j).$$

Then by applying Lo (1984), we can show the posterior distribution of $\boldsymbol{\theta}$ given $\mathbf{y}$ is

$$dH(\boldsymbol{\theta}|\mathbf{y}) = \frac{\prod_{j=1}^{n} f_j(y_j|\theta_j) dH(\boldsymbol{\theta})}{\int \prod_{j=1}^{n} f_j(y_j|\theta_j) dH(\boldsymbol{\theta})}.$$

So the posterior mean for θ_i, for any i, can be written as

$$\begin{aligned} \hat{\theta}_i(\mathbf{y}) &= \int_{\mathcal{R}^n} \cdots \int \theta_i dH(\boldsymbol{\theta}|\mathbf{y}) = \frac{\int_{\mathcal{R}^n} \cdots \int \theta_i \prod_{j=1}^{n} f_j(y_j|\theta_j) dH(\boldsymbol{\theta})}{\int \prod_{j=1}^{n} f_j(y_j|\theta_j) dH(\boldsymbol{\theta})} \\ &= \frac{\int_{\mathcal{R}^n} \cdots \int \theta_i \prod_{j=1}^{n} f_j(y_j|\theta_j) \prod_{j=1}^{n} \frac{\alpha G_0 + \sum_{l=1}^{j-1} \delta_{\theta_l}}{\alpha + j - 1}(d\theta_j)}{\int_{\mathcal{R}^n} \cdots \int \prod_{j=1}^{n} f_j(y_j|\theta_j) \prod_{j=1}^{n} \frac{\alpha G_0 + \sum_{l=1}^{j-1} \delta_{\theta_l}}{\alpha + j - 1}(d\theta_j)}. \end{aligned} \tag{2.4}$$

A direct derivation for the above expression given in Kuo (1980) is omitted here. Although the above expression (2.4) looks deceptively simple, it is actually a weighted mean from many distributions. Antoniak (1974) had given a detailed account on calculating the probabilities for various configurations for the latent variables. For $n = 3$ and $i = 3$ as an example, then (2.4) is a weighted mean of 5 terms which is proportional to

$$\begin{aligned} & \frac{\alpha^2}{(\alpha+2)(\alpha+1)} \int f_1(y_1|\theta) G_0(d\theta) \int f_2(y_2|\theta) G_0(d\theta) \\ & \quad \times \int \theta f_3(y_3|\theta) G_0(d\theta) \\ + & \frac{\alpha}{(\alpha+2)(\alpha+1)} \int \theta f_1(y_1|\theta) f_3(y_3|\theta) G_0(d\theta) \int f_2(y_2|\theta) G_0(d\theta) \\ + & \frac{\alpha}{(\alpha+2)(\alpha+1)} \int \theta f_2(y_1|\theta) f_3(y_3|\theta) G_0(d\theta) \int f_1(y_1|\theta) G_0(d\theta) \\ + & \frac{\alpha}{(\alpha+2)(\alpha+1)} \int f_1(y_1|\theta) f_2(y_2|\theta) G_0(d\theta) \int \theta f_3(y_3|\theta) G_0(d\theta) \\ + & \frac{2}{(\alpha+2)(\alpha+1)} \int \theta f_1(y_1|\theta) f_2(y_2|\theta) f_3(y_3|\theta) G_0(d\theta), \end{aligned}$$

where the five terms are obtained from cases (1) $\theta_3 \neq \theta_2 \neq \theta_1$, (2) $\theta_3 = \theta_1 \neq \theta_2$, (3) $\theta_3 = \theta_2 \neq \theta_1$, (4) $\theta_2 = \theta_1 \neq \theta_3$, and (5) $\theta_3 = \theta_2 = \theta_1$, respectively. So the first term (for case (1)) is a product of three integrals where each kernel is integrated against its prior latent variable density. Each of the next three terms

(for cases (2), (3) or (4)) is a product of two integrals where the first integral having two observations sharing the same latent variable integrated against the prior latent density, and the second integral using the third kernel integrated against the latent density G_0. For the fifth term, all observations share the same latent variable θ, and integrated against the $G_0(d\theta)$ distribution.

The number of components in the mixture distribution given by the Bell exponential number goes up to 15, 52, 203, and 787 for n=4, 5, 6, and 7. So many computational algorithms have been developed to evaluate the posterior distribution. In the following we will discuss two algorithms: the truncated blocked Gibbs sampler by Ishwaran and James (2001), and the slice sampler by Kalli et al. (2010) to handle computation for large samples.

3. Sampling From DPM Process

3.1. Sampling from DP

From Sethuraman's representation of DP (Sethuraman, 1994), we can derive an equivalent representation for a DPM process:

$$
\begin{aligned}
&w_1, w_2, \ldots \sim \text{GEM}(\alpha);\\
&\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \overset{i.i.d.}{\sim} G_0;\\
&Pr(s_i = j) = w_j, \text{with } j = 1, 2, \ldots \text{for each } i = 1, \ldots, n;\\
&\boldsymbol{y}_i \overset{ind}{\sim} f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_{s_i}), \text{ for each } i = 1, \ldots, n,
\end{aligned}
$$

where s_i is a discrete latent allocation variable that assigns the ith latent variable to one of the latent class; so $s_i = j$ denotes the ith latent variable belongs to the jth cluster. The GEM condition is the same as described in (2.2), the name was attributed to Griffiths, Engen, and McCloskey (Ewens and Tavaré, 1998).

We next describe two samplers: truncated blocked Gibbs sampler and slice Gibbs sampler for the DPM.

Truncated Blocked Gibbs Sampler for DPM:

This algorithm developed by Ishwaran and James (2001) consists of truncating the infinitely many atoms of the random mixing distribution G to a fixed finite number of atoms before hand. This is done by truncating the stick-breaking weights to at most R terms, where R is prespecified and v_ls are defined as in (2.2) for $l < R$ and $V_R = 1$. So the number of atoms of G is fixed at at most R. Then the latent parameters θ's are clustered and updated with a dimension to be at most R, where the clusters are represented by their distinct values $\theta_1^*, \ldots, \theta_R^*$. Then we list the MCMC steps as follows:

Step 1. Initialization

1.1 Generate $v_1, \ldots, v_{R-1} \sim \mathcal{B}e(1, \alpha)$, and set $V_R = 1$; Calculate $w_j = v_j \prod_{\ell<j}(1 - v_\ell)$, for $j = 1, \ldots, R$;

1.2 Generate $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_R^* \sim G_0$;

1.3 For each i, $i = 1, \ldots, n$, generate a discrete random variable s_i with integer values $1, \ldots, R$ such that $Pr(s_i = j) = w_j = v_j \prod_{\ell<j}(1 - v_\ell)$, for $j = 1, \ldots, R$.

Step 2. Update the discrete random variable s_i for each i, $i = 1, \ldots, n$, where

$$Pr(s_i = j|...) = \frac{v_j \prod_{\ell<j}(1 - v_\ell) \times f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_j^*)}{\sum_{k=1}^{R}[v_k \prod_{\ell<k}(1 - v_\ell) \times f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_k^*)]},$$

with $j = 1, ..., R$;

Step 3. For each j, $j = 1, 2, \ldots, R$, sample $\boldsymbol{\theta}_j^*$ with

$$\pi(\boldsymbol{\theta}_j^*|...) \propto dG_0(\boldsymbol{\theta}_j^*) \prod_{s_i=j} f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_j^*).$$

These θ^*s can be updated simultaneously.

Step 4. Update $v_j, j = 1, ..., R$, such that

$$\pi(v_j|...) \sim \mathcal{B}e(1 + n_j, \alpha + m_j)$$

where $n_j = \sum_{k=1}^{n} I(s_k = j)$ and $m_j = \sum_{k=1}^{n} I(s_k > j)$.

Step 5. Go back to Step 2 and repeat.

If DPM has another parameter, say σ, shared by all components in the first stage, then we can easily add Step 4.5 after Step 4. In Step 4.5, σ is updated by the Metropolis-Hastings algorithm (Hastings, 1970), where the posterior density of σ is proportional to $\pi(\sigma) \prod_{i=1}^{n} f_i(\boldsymbol{y}_i|\theta_i, \sigma)$ with $\pi(\sigma)$ being the prior density of σ.

Slice Gibbs Sampler for DPM:
Walker (2007) first proposed a slice sampler for DPM. Later Kalli et al. (2010) proposed a more efficient version of the slice sampler. The slice sampler updates the truncation point at each iteration, so only needs to update a finite mixture in each iteration of the MCMC algorithm. We briefly describe the Kalli et al. algorithm in the followings:
Let R denote the number of mixing components. Unlike the truncated blocked Gibbs sampler, the slice Gibbs sampler updates R as follows.

Step 1. Initialization

1.1 Generate $v_1, \ldots, v_j \sim \mathcal{B}e(1, \alpha)$; Calculate $w_j = v_j \prod_{\ell<j}(1 - v_\ell)$, for $j = 1, \ldots, n$;

1.2 Generate $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \sim G_0$;

1.3 $Pr(s_i = j) = w_j = v_j \prod_{\ell<j}(1 - v_\ell)$, for $i = 1, .., n, j = 1, \ldots, n$;

1.4 R=1.

Step 2. Update s_i

2.1 Sample $u_i \sim U(0, w_{s_i})$, $i = 1, \ldots, n$;

2.2 Let $u^* = \min\{u_1, ..., u_n\}$,

-If $\sum_{j=1}^{\text{R (number of clusters)}} w_j > 1 - u^*$, sample s_i according to

$$Pr(s_i = j|...) = \frac{w_j \times f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_j)}{\sum_{\ell=1}^{R}[w_\ell \times f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_\ell)]}$$

with $j = 1, ..., R$;

-Otherwise, let R=R+1.

Step 3. For each j, sample $\boldsymbol{\theta}_j$ with

$$\pi(\boldsymbol{\theta}_j|...) \propto dG_0(\boldsymbol{\theta}_j) \prod_{s_i=j} f_i(\boldsymbol{y}_i|\boldsymbol{\theta}_j),$$

for $j = 1, 2, ..., R$.

Step 4. Update $v_j, j = 1, ..., R$, such that

$$\pi(v_j|...) \sim \mathcal{B}e(1 + n_j, \alpha + m_j)$$

where $n_j = \sum_{\ell=1}^{n} I(s_\ell = j)$ and $m_j = \sum_{\ell=1}^{n} I(s_\ell > j)$.

Step 5. Go back to Step 2 and repeat.

4. DP Configuration Tracking

4.1. General Theory

We first define a few notations to facilitate describing the equivalence relation. Let the superscript m denote the m^{th} iteration of the Markov chain Monte Carlo (MCMC) sampler, where $m = 1, ..., M$.

We now introduce three index sets:

- Cluster allocation index set: $\boldsymbol{s}^{(m)} = \{s_1^{(m)}, ..., s_n^{(m)}\}$, for $i = 1, ..., n$, where $s_i^{(m)}$ denotes the allocated cluster for the ith latent variable in the mth iteration.
- Distinct cluster index set: $\boldsymbol{k}^{(m)} = \{k_1^{(m)}, \ldots, k_{K^{(m)}}^{(m)}\}$, where $K^{(m)}$ denotes the number of distinct cluster allocation index from $\boldsymbol{s}^{(m)}$, and each component of $\boldsymbol{k}^{(m)}$ denotes the unique allocation index for each cluster.
- Configuration index set: $\boldsymbol{L}^{(m)} = \{L_1^{(m)}, \ldots, L_{K^{(m)}}^{(m)}\}$, where $L_j^{(m)} = \{i : s_i^{(m)} = k_j^{(m)}\}$ for $j = 1, ..., K^{(m)}$. So the configuration index set is the set of subject labels for each distinct cluster.

Next, we present the definition of equivalence between two cluster allocation index sets.

Definition of Equivalence. *Let $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ denote two sets of allocation indices. Then, $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ are said to be equivalent or belong to the same configuration, denoted by*

$$\boldsymbol{s}^{(m_1)} \backsimeq \boldsymbol{s}^{(m_2)},$$

if the conditional posterior distribution of the distinct components of $\boldsymbol{\theta}^{(m_1)} = (\theta_{s_1^{(m_1)}}, ..., \theta_{s_n^{(m_1)}})'$ is identical to the conditional posterior distribution of the distinct components of $\boldsymbol{\theta}^{(m_2)} = (\theta_{s_1^{(m_2)}}, ..., \theta_{s_n^{(m_2)}})'$, which is denoted by

$$\boldsymbol{\theta}^{(m_1)} \overset{L}{=} \boldsymbol{\theta}^{(m_2)}.$$

When $\boldsymbol{s}^{(m_1)} \backsimeq \boldsymbol{s}^{(m_2)}$, we can claim that $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ belong to the same configuration in MCMC sampling from the DP process. The following theorem characterizes the conditions to determine the equivalence between $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ or whether $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ belong to the same configuration.

Theorem 4.1. *Let $\boldsymbol{s}^{(m_1)}$ and $\boldsymbol{s}^{(m_2)}$ denote two sets of allocation indices from the m_1th and m_2th iterations of MCMC sampling. Then, $\boldsymbol{s}^{(m_1)} \backsimeq \boldsymbol{s}^{(m_2)}$ if and only if*

(i) $K^{(m_1)} = K^{(m_2)}$*; and*

(ii) $L_j^{(m_1)} = L_j^{(m_2)}$ *for all* $j = 1, \ldots, K^{(m_1)}$.

The proof is straightforward.

We next count the number of distinct configurations and the number of iterations that yield the same configurations.

DP Configuration Tracking Algorithm:

Step 0. Set $N_c = 1$, $m = 1$, $t^{(1)} = 1$, $K^{*(1)} = K^{(1)}$, $\boldsymbol{L}^{*(1)} = \boldsymbol{L}^{(1)}$, $\boldsymbol{\Gamma} = \{(K^{*(1)}, \boldsymbol{L}^{*(1)})\}$, and $\mathcal{T} = \{t^{(1)}\}$.

Step 1. Let $m \leftarrow m + 1$.

Step 2. If $(K^{(m)}, \boldsymbol{L}^{(m)}) = (K^{*(j)}, \boldsymbol{L}^{*(j)}) \in \boldsymbol{\Gamma}$, then $t^{(j)} \leftarrow t^{(j)} + 1$ and go to Step 4.

Step 3. If $(K^{(m)}, \boldsymbol{L}^{(m)}) \notin \boldsymbol{\Gamma}$, then let $K^{*(N_c+1)} = K^{(m)}$, $t^{(N_c+1)} = 1$, and $\boldsymbol{L}^{*(N_c+1)} = \boldsymbol{L}^{(m)}$, add $(K^{*(N_c+1)}, \boldsymbol{L}^{*(N_c+1)})$ to $\boldsymbol{\Gamma}$, also add $t^{(N_c+1)}$ to $\mathcal{T}$, and then update $N_c \leftarrow N_c + 1$.

Step 4. if $m < M$, then go to Step 1 and if $m = M$, then stop.

At the end of the above algorithm, N_c is the number of distinct configurations, $\boldsymbol{\Gamma}$ includes all members of N_c distinct configurations, and $\mathcal{T}$ tracks the corresponding frequencies of the members of N_c distinct configurations in $\boldsymbol{\Gamma}$ in the M MCMC iterations. In addition, $t^{(j)}/M$ gives an MCMC estimate of the probability of configuration $\boldsymbol{L}^{*(j)}$ for $j = 1, \ldots, N_c$.

4.2. An Illustrative Example

Suppose we have $M = 4$ MCMC iterations for four observations $y_1, \ldots, y_4$ ($n = 4$). Assume the allocation indices for them are $\boldsymbol{s}^{(1)} = \{1, 2, 2, 3\}$; $\boldsymbol{s}^{(2)} = \{2, 1, 1, 3\}$; $\boldsymbol{s}^{(3)} = \{3, 3, 1, 2\}$; and $\boldsymbol{s}^{(4)} = \{2, 2, 1, 1\}$. Hence, by applying the notations from Section 4.1, we have

$$\boldsymbol{k}^{(1)} = \{1, 2, 3\};\ \boldsymbol{k}^{(2)} = \{2, 1, 3\};\ \boldsymbol{k}^{(3)} = \{3, 1, 2\};\ \text{and}\ \ \boldsymbol{k}^{(4)} = \{2, 1\}.$$

By definition, we have $K^{(1)} = K^{(2)} = K^{(3)} = 3$, $K^{(4)} = 2$,

$$\begin{aligned}
&L_1^{(1)} = \{1\},\ L_2^{(1)} = \{2, 3\},\ L_3^{(1)} = \{4\},\ \boldsymbol{L}^{(1)} = (L_1^{(1)}, L_2^{(1)}, L_3^{(1)});\\
&L_1^{(2)} = \{1\},\ L_2^{(2)} = \{2, 3\},\ L_3^{(2)} = \{4\},\ \boldsymbol{L}^{(2)} = (L_1^{(2)}, L_2^{(2)}, L_3^{(2)});\\
&L_1^{(3)} = \{1, 2\},\ L_2^{(3)} = \{3\},\ L_3^{(3)} = \{4\},\ \boldsymbol{L}^{(3)} = (L_1^{(3)}, L_2^{(3)}, L_3^{(3)}); \text{and}\\
&L_1^{(4)} = \{1, 2\},\ L_2^{(4)} = \{3, 4\},\ \boldsymbol{L}^{(4)} = (L_1^{(4)}, L_2^{(4)}).
\end{aligned}$$

Based on the configuration tracking algorithm, we further have $N_c = 3$,

$$\begin{aligned}
&K^{*(1)} = K^{*(2)} = 3,\ K^{*(3)} = 2,\ \boldsymbol{L}^{*(1)} = \boldsymbol{L}^{(1)} = \boldsymbol{L}^{(2)},\\
&\boldsymbol{L}^{*(2)} = \boldsymbol{L}^{(3)},\ \boldsymbol{L}^{*(3)} = \boldsymbol{L}^{(4)},\ t^{(1)} = 2,\ t^{(2)} = t^{(3)} = 1,\\
&\boldsymbol{\Gamma} = \{(3, \boldsymbol{L}^{*(1)}), (3, \boldsymbol{L}^{*(2)}), (2, \boldsymbol{L}^{*(3)})\},\ \text{and}\ \mathcal{T} = \{2, 1, 1\}.
\end{aligned}$$

It is clear that $K^{(1)} = K^{(2)} = K^{(3)} = 3$, which satisfies **Theorem 4.1** (i). Since $K^{(4)} = 2$ which is different from the other 3 iterations, it is ruled out by **Theorem 4.1** (i) that $\boldsymbol{s}^{(4)}$ is equivalent to any of $\boldsymbol{s}^{(1)}$, $\boldsymbol{s}^{(2)}$, and $\boldsymbol{s}^{(3)}$. We continue to check condition (ii) in **Theorem 4.1**. It shows that $\boldsymbol{L}_j^{(1)} = \boldsymbol{L}_j^{(2)}$ for $j = 1, 2, 3$, which satisfies **Theorem 4.1** (ii). And $\boldsymbol{L}_j^{(1)} \neq \boldsymbol{L}_j^{(3)}$ for $j = 1, 2$, which does not satisfy **Theorem 4.1** (ii). Thus, in this example, we have showed that $\boldsymbol{s}^{(1)} \backsimeq \boldsymbol{s}^{(2)}$, $\boldsymbol{s}^{(1)} \neq \boldsymbol{s}^{(3)}$, $\boldsymbol{s}^{(1)} \neq \boldsymbol{s}^{(4)}$, $\boldsymbol{s}^{(2)} \neq \boldsymbol{s}^{(3)}$, $\boldsymbol{s}^{(2)} \neq \boldsymbol{s}^{(4)}$, and $\boldsymbol{s}^{(3)} \neq \boldsymbol{s}^{(4)}$.

5. Examples

5.1. Simulation Study

In this subsection, we generated two data sets with sample sizes 5 and 10 each from two different mixtures of normal distributions, where the first mixture consists of 3 normal distributions well separated with equal weights, and the second mixture, similar, but not so well separated. We applied DPM in (2.3) with f_i to be a normal density with mean θ_i and variance 1. The base distribution G_0 was specified to be $N(0, 1/\tau)$. We varied the precision τ in G_0 with $\tau = 1, 0.1,$ or 0.00001, and α to be 0.1, 1 or 10. Then we applied the truncated blocked Gibbs sampler with fixed R to be the same as the sample size. We report the results from each MCMC run with 20,000 iterations after 1000 "burn-in" iterations. In the following, we list the populations, the simulated data sets (denoted by 1a, 1b, and 2a, 2b) from each population, the resulting most probable cluster configurations and tables for the probabilities for the most probable configuration with various choices of α and τ. For the varying choices of α and τ, the resulting most probable configuration was always the same except in the 1b scenario, so we summarized them before the table of probabilities.

1.Population $\frac{1}{3}N(-5, 1) + \frac{1}{3}N(0, 1) + \frac{1}{3}N(5, 1)$

1a:

Simulated Data	-5.33	4.16	5.41	-5.82	4.71
Most Probable Configuration	1	2	2	1	2

1b:

Simulated Data									
-5.33	4.16	5.41	-5.82	4.71	0.58	0.76	-4.61	-5.29	1.12
Most Probable Configuration									
1	2	2	1	2	3	3	1	1	3
* Most probable configuration									
1	2	2	1	2	2	2	1	1	2
** Most probable configuration									
1	2	2	1	2	3	4	1	1	2

2. Population $\frac{1}{3}N(-1, 1) + \frac{1}{3}N(0, 1) + \frac{1}{3}N(1, 1)$

Table 5.1: Probability for the most probable configuration for three choices of α and τ each

Prior Choice	Scenario 1a			Scenario 1b		
$\alpha\backslash\tau$	1	0.1	0.00001	1	0.1	0.00001
0.1	0.999	0.971	0.999	0.413	0.841	0.608*
1	0.986	0.780	0.990	0.297	0.409	0.858
10	0.984	0.757	0.988	0.350**	0.098	0.879

2a:

Simulated Data	-0.51	-0.37	-1.61	0.39	-0.76
Most Probable Configuration	1	1	1	1	1

2b.

Simulated Data									
-0.51	-0.37	-1.61	0.39	-0.76	-1.63	0.98	0.76	0.54	-.26
Most Probable Configuration									
1	1	1	1	1	1	1	1	1	1

Table 5.2: Probability for the most probable configuration for three choices of α and τ each

Prior Choice	Scenario 2a			Scenario 2b		
$\alpha\backslash\tau$	1	0.1	0.00001	1	0.1	0.00001
0.1	0.854	0.916	0.997	0.776	0.895	0.999
1	0.256	0.465	0.991	0.125	0.317	0.986
10	0.234	0.443	0.990	0.003	0.021	0.933

The results are expected and reasonable. Moreover, the columns with $\tau = 0.00001$ show the sampler converges to the resulting configuration with a high probability. Viewing for each fixed column, we can also observe the probabilities tend to decrease as α increases, which is also expected because the number of clusters increases as α increases.

5.2. Real Data Analysis

The real data set consists of $(y_1, \ldots, y_{12}) = (8, 4, 0, 0, 0, 0, 1, 4, 4, 0, 0, 0)$ ($n = 12$), which are pollen counts over 12 days collected in the late season of 1991 at Kalamazoo, Michigan taken from Chen and Ibrahim (2000). We fit a hierarchical mixture model (2.3) to it with a Poisson model in the first stage, and G_0, to be a gamma distribution $\mathcal{G}(a, b)$ with mean a/b in the second stage. We used the blocked truncated Gibbs sampler to truncate the model to be at most three clusters. We report the three most frequent configurations for $\alpha = 0.1, 1,$ or 10, and four choices of hyperparameters a and b with a MCMC run of 10000

iterations with 1000 iterations as burn-in in Table 5.3. The hyperparameters $a = 0.591$ and $b = 0.338$ in the second block were chosen from the matching functional form consideration for the parametric empirical Bayes model, that is the marginal mean and variance of y are matched to the empirical mean and variance of y. The other choices were more arbitrary except the mean of G_0 was chosen to be the empirical mean or close to it and the variance of G_0 was chosen around ten fold up as we go down each block of the table.

Table 5.3: Three most frequent configurations with their probabilities in parentheses for three choices of α

Prior choice		Most frequent configuration with probability		
$G_0 = \mathcal{G}(a, b)$	α	First	Second	Third
$a = 1.75$	0.1	112222211222	112222111222	111111111111
		(0.350)	(0.138)	(0.0648)
&	1	112222211222	112222111222	112222311222
		(0.126)	(0.0541)	(0.0308)
$b = 1$	10	112222211222	112222111222	112222311222
		(0.367)	(0.0477)	(0.0382)
$a = 0.591$	0.1	112222211222	112222111222	111111111111
		(0.312)	(0.242)	(0.0397)
&	1	112222211222	112222111222	112222311222
		(0.0989)	(0.0922)	(0.0416)
$b = 0.338$	10	112222211222	112222111222	112222311222
		(0.346)	(0.0822)	(0.0582)
$a = 0.2$	0.1	112222111222	112222211222	111111111111
		(0.363)	(0.199)	(0.0281)
&	1	112222111222	112222211222	112222311222
		(0.130)	(0.0652)	(0.0368)
$b = 0.1$	10	112222211222	112222111222	112222311222
		(0.272)	(0.152)	(0.076)
$a = 0.0175$	0.1	112222111222	111111111111	112222211222
		(0.546)	(0.0347)	(0.0297)
&	1	112222111222	112322111222	112232111222
		(0.197)	(0.0264)	(0.0263)
$b = 0.01$	10	112222111222	112222211222	112222311222
		(0.430)	(0.0765)	(0.0311)

The results show two clusters are well supported for various choices of a, b, and α. The first cluster consists of pollen counts $(8, 4, 4, 4, 4)$ observed on day 1, 2,

8, and 9. The second cluster consists of all pollen counts that are zero, observed on 3, 4, 5, 6, 10, 11, and 12. The count 1, observed on day 7th, is clustered into the second cluster of count zero for the first two choices of G_0 (small variance) and clustered into the first cluster for the last two choices of G_0 (big variance). When α was chosen to be 0.1, we see the results also support only one cluster with 6% to 3% probabilities depending on the hyperparameters we chose. This is expected due to higher tendency for doubling up of the atoms of the DP in the mixing distribution for small α. When $\alpha = 10$, we see the results also support three (the maximum number allowed) clusters with 3% to 8% probabilities, where the count 1 observed on the day 7th often forms the third cluster.

6. Discussion

In this paper, we have discussed a hierarchical Dirichlet process mixing model as in (2.3). We wrote it very general to allow different components to have different models, and also to allow vector forms of observation for each component and vector latent parameters. The DP process constructed on the distribution of the latent parameter relaxes the strong parametric assumptions made in the usual hierarchical model. We discussed two MCMC algorithms for updating the parameters. The blocked truncated Gibbs sampler restricts the number of atoms in DP to be at most R; the slice Gibbs sampler updates the number of atoms in DP using slice with no *a priori* restriction on it. When applying these algorithms to formulate cluster configurations, allocation indices (labels) are usually not uniquely defined. In order to preserve the posterior distribution of the latent variables given the data, we establish an equivalence theorem for two allocation indices to be equivalent from two iterations of the MCMC run. This theorem establishes that the two sets of indices not only need to have the same number of distinct clusters, but also need to have the same components for each cluster. We recommend adding this check for equivalence algorithm to the DPM algorithm. This check will identify permutations of labels that lead to the same cluster configuration. It will effectively recognize the label switching problem and tabulate the equivalent cluster configuration correctly. We have conducted a simulation study and a real data analysis using the blocked truncated Gibbs sampler and the configuration checking algorithm in Section 4. Our simulation study shows our method is efficient; and our real data analysis with varying prior choice provides more insights on the sensitivity analysis to the prior choice.

Acknowledgements

This work was supported by the US National Institutes of Health under Grants #GM 70335 and #CA 74015 to M.-H. Chen and the US National Science Foundation under Grant No. DEB-1036448 to P. O. Lewis.

References

1. Aldous, D. J. (1985). Exchangeability and Related Topics. *Ecole dEte de Probabilites de Saint- Flonr XIII-1983* Lecture Notes in Mathematics 1117, P. L. Hennequin, ed., Springer-Verlag, Berlin, pp. 2-198.

2. Antoniak, C. E. (1974). Mixture of Dirichlet Processes with Applications to Bayesian Nonparametric Problems, *Annals of Statistics* 2: 1152-1174. DOI: 10.1214/aos/1176342871

3. Berge, C. (1971). *Principles of Combinatorics,* New York: Academic Press.

4. Blackwell, D. and MacQueen, J. B. (1973). Ferguson Distributions Via Pólya Urn Schemes, *Annals of Statistics* 1:353-355.

5. Chen, M.-H. and Ibrahim, J. G. (2000). Bayesian Prediction Inference for Time Series Count Data. *Biometrics* 56:678-685. DOI: 10.1111/j.0006-341X.2000.00678.x

6. Dunson, D. B. (2010). Nonparametric Bayes Applications to Biostatistics, *Bayesian Non-parametrics*, ed. N. L. Hjort. New York: Cambridge University Press, pp. 223-270.

7. Escobar, M. D. and West, M. (1995). Bayesian Density Estimation and Inference Using Mixtures, *Journal of the American Statistical Association* 90: 577-588. DOI: 10.1080/01621459.1995.10476550

8. Escobar, M. D. and West, M. (1998). Computing Nonparametric Hierarchical Models, *Practical Nonparametric and Semiparametric Bayesian Statistics* eds. D. Dey, P. Mueller, and D. Sinha. New York: Springer-Verlag, pp. 1-22.

9. Ewens, W. J. and Tavaré, S. (1998). Ewens Sampling Formula, *Encyclopedia of Statistical Sciences, updated Volume 2*, S. Kotz, C. B. Read, and D. L. Banks eds., pp. 230-234, New York: John Wiley & Sons.

10. Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problmes, *Annals of Statistics* 1: 209-230.

11. Hastings, W. (1970). Monte Carlo Sampling Methods Using Markov Chainsd and Their Application, *Biometrika* 57: 97-109. DOI: 10.1093/biomet/57.1.97

12. Ishwaran, H. and James, L. F. (2001). Gibbs Sampling Methods for Stick-Breaking Priors, *Journal of the American Statistical Association* 101: 179-194.

13. Kalli, M., Griffin, J. E., and Walker, S. G. (2010). Slice Sampling Mixture Models, *Statistics and Computing* 21: 93-105.

14. Kuo, L. (1980). Computation and Applications of Mixtures of Dirichlet Processes. PhD Dissertation, UCLA.

15. Kuo, L. (1983). Bayesin Bioassay Design, *Annals of Statistics* 11: 886-895. DOI: 10.1214/aos/1176346254

16. Kuo, L. (1986a). A Note on Bayes Empirical Bayes Estimation by Means of Dirichlet Processes, *Stattistics & Probability Letters* 4: 145-150.

17. Kuo, L. (1986b). Computation of Mixtures of Dirichlet Processes, *SIAM Journal on Scientific and Statistical Computing* 7: 60-71.

18. Lo, A. Y. (1984). On a Class of Bayesian Nonparametric Estimates: 1. Density Estimates, *Annals of Statistics* 12: 351-357.

19. MacEachern, S. N. and Müller, P. (1998). Estimating Mixture of Dirichlet Process Models, *Journal of Computational and Graphical Statistics* 7: 223-238.

20. Neal, R. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models, *Journal of Computational and Graphical Statistics* 9: 249-265.

21. Sethuraman, J. (1994). A Constructive Definition of Dirichlet Priors, *Statistica Sinica* 4: 639-650.

22. Sethuraman, J. and Tiwari, R. C. (1982). Convergence of Dirichlet Measures and the Interpretation of Their Parameter, *Statistical Decision Theory and Related Topics III, Volume 2*, S. S. Gupta and J. O. Berger eds., pp. 305-315, New York: Academic Press.

23. Walker, S. G. (2007). Sampling the Dirichlet Mixture Models with Slices, *Communications in Statsitics: Simulation and Computation* 36: 45-54. DOI: 10.1080/03610910601096262