Bayesian Analysis (2007) **2**, Number 3, pp. 483–494

Comment on Article by Jain and Neal

Steven N. MacEachern*

1 Introduction

It was with great interest that I read Jain and Neal's paper. In the paper, they address a tough problem, namely how to improve the mixing/convergence of Markov chain Monte Carlo (MCMC) algorithms for an important class of models. The models are those involving mixtures of Dirichlet processes, ranging from a fairly straightforward mixture of Dirichlet processes model to the more complex models that are springing up in a wide variety of applications. The algorithms are in the split-merge vein, allowing a different kind of step than incremental Gibbs samplers. The extension of the split-merge technology with targeted proposals to conditionally conjugate models is a welcome addition to the collection of transitions available for fitting models that include the Dirichlet process as a component.

Jain and Neal's algorithms (see also Dahl, 2005) have refined the technology of split-merge samplers so that proposals are no longer "blind", but, through intermediate Gibbs scans, move toward a region of higher posterior probability. The ability to target better proposals results in algorithms that naturally make better proposals, and this improves mixing of the Markov chain. An important element of these intermediate Gibbs scans is their ability to move toward a more appropriate launch state.

This discussion focuses on two features that are hidden in the innards of the algorithm. The first is the notion of identifiability and the second is that of a random scan. Jain and Neal's algorithms make nice use of a non-identifiable model for the intermediate Gibbs scans (section 4.2, step 3 and following) to produce what are presumably better proposals. They also implicitly use a random scan for split and merge proposals in the sense that cases i and j are selected at random (section 4.2, step 1). The remainder of this discussion looks at these issues in the context of a simple, artificial example where one can explicitly calculate rates of convergence for a variety of incremental Gibbs algorithms. The hope is that the example, in spite of its simplicity, provides insight into the effectiveness of the algorithms and suggests potential directions for their further refinement.

2 Identifiability

While details of various algorithms are left for the next section, one recurring issue in proposals for novel algorithms for Dirichlet based models is identifiability. This issue is not limited to mixture models, but arises in many other contexts. There is

*Department of Statistics, The Ohio State University, Columbus, OH, mailto:snm@stat.ohio-state.edu

often a connection between identifiability and the convergence rate of a Markov chain: Identifiable models may show quicker convergence to the limiting distribution than do non-identifiable models. This has led some to suggest a general principle that non-identifiable models be avoided when MCMC methods are to be used to fit the model. This section reviews the arguments raised against non-identifiable models, and the following section develops the arguments in more detail through consideration of a simple example.

Consider a model where there is a parameter space, say Θ. The distribution of the data depends on the value of the parameter, so that $X \sim F_\theta$ for some $\theta \sim \Theta$. A model is non-identifiable if there exist $\theta_1, \theta_2 \in \Theta$, with $\theta_1 \neq \theta_2$, for which $F_{\theta_1} = F_{\theta_2}$. Models that are not non-identifiable are called identifiable models. Typically, when the model is non-identifiable, it will be the case that for every $\theta_1 \in \Theta$ there exists a $\theta_2 \in \Theta$, with $\theta_2 \neq \theta_1$ for which $F_{\theta_1} = F_{\theta_2}$.

Several reasons have been given for avoiding the use of non-identifiable models. First, while a Bayesian approach places a prior over the parameter space, and so, in principle, there is no difficulty in creating estimates with this methodology, there is the question of consistency. Identifiability is closely connected with parameter estimation. Methods such as maximum likelihood cannot distinguish between parameter values that imply the same distribution for the data, and so may not produce unique estimates. Bayes estimates, heavily based on the likelihood, are typically also inconsistent for non-identifiable models. However, if consideration is restricted to identifiable functionals, the Bayes estimates will typically be consistent, as they are under identifiable models. A desire to interpret parameter values directly is closely related to a desire for consistency. Restricting interpretation to identifiable quantities $g(\theta)$, such that if $g(\theta_1) \neq g(\theta_2)$ then $F_{\theta_1} \neq F_{\theta_2}$, the worry about non-identifiability disappears. A complete, identical Bayes analysis could be done on an identifiable model. This first objection has no connection to the use of MCMC methods.

Second, there are examples where the convergence rate of a Markov chain is improved by the choice of an identifiable model. The convergence here is convergence of π_n, the distribution of $\theta^n|\theta^0$, to the limiting distribution of the Markov chain. The limiting distribution is, by construction, also the posterior distribution of θ. The main purpose of the simulation is to provide estimates of posterior summaries, and, although there is a difference between the accuracy of these estimates and the convergence rate, in most circumstances the two produce qualitative agreement: A better convergence rate means more accurate estimators. This issue is examined in the next section.

Third, there is the practical issue of how well the MCMC algorithm works when actually implemented. The main concerns are the numerical accuracy and stability of the computations. In some instances, particularly with very diffuse posterior distributions, some of the parameter values generated during the course of the simulation may be enormous. This can lead to unstable computations and hence to inaccurate estimates.

Fourth, there is the issue of prior elicitation. The choice of a model has an impact on the particular prior that is chosen. This choice is not directly tied to the use of MCMC methods, but is an issue of increasing importance now that more complex models are being fit. Examples include collinearity and the variable selection problem where priors are chosen according to prescription, problems based on the hierarchical model, and nonparametric Bayes problems.

Consider fitting models with MCMC methods. The Markov chain upon which the simulation is based is realized through successive generations of a parameter vector, θ. The chain, assumed to be irreducible and aperiodic, is also assumed to have a fixed transition matrix, say P. Consequently, it has a limiting distribution, π. The transition matrix is chosen in such fashion that π is the posterior distribution for $\theta|X$. A realization of the chain consists of a sequence $\theta^1, \theta^2, \ldots, \theta^N$. Convergence is often described in terms of the total variation norm: We wish $||\pi_n - \pi||$ to approach 0 quickly. For finite state chains, the rate of convergence is governed by the second largest eigenvalue of the transition matrix. The convergence rate of more complex chains is determined by a similar quantity.

The MCMC method constructs P by creating a set of transition kernels. For a fixed scan algorithm, the overall transition kernel is the product of, say, p transition kernels, $P = P_1 \ldots P_p$. A random scan sampler selects one of the P_i at random. A popular choice is to select the P_i with equal probabilities, so that $P = p^{-1} \sum P_i$.

Two useful techniques for improving convergence of a sampler are (i) to generate a block of parameters at a time (say $\theta_1, \ldots, \theta_c$ is generated from $[\theta_1, \ldots, \theta_c | \theta - \{\theta_1, \ldots, \theta_c\}, X]$), and (ii) to collapse or coarsen the state space of the Markov chain by reducing the dimension of θ. The dimension of θ is reduced through integration. For example, θ_p may be marginalized, leaving only $\theta_1, \ldots, \theta_{p-1}$. For discussion, theory and examples of (i) and (ii) see Liu (1995); of (ii) see also MacEachern (1994). The impact of non-identifiability on MCMC algorithms is closely connected to blocking and coarsening.

3 Illustration

Nonparametric Bayesian models have been considered for several decades. Early models, such as those of Kraft and van Eeden (1964) and Ramsey (1972) for the bioassay problem, provided a start in the area. These models were based on the notion of a Dirichlet distribution being the conjugate prior for multinomial data. The models were nonparametric in the sense that the prior had full support on the set of multinomial probability vectors. This work was followed by the well-known work of Ferguson (1973) and Antoniak (1974). Early work exploiting mixtures of Dirichlet processes includes Berry and Christensen (1979) and Lo (1984).

The mixture of Dirichlet process model has many applications beyond bioassay. The basic mixture of Dirichlet processes model may be written as follows:

$$\begin{aligned} F &\sim Dir(\alpha) \\ \theta_1, \ldots, \theta_p | F &\sim F \\ X_i | \theta_i &\sim G_{\theta_i}, \text{ for } i = 1, \ldots, p. \end{aligned}$$

Here, following Ferguson's notation, α, the positive, finite measure that parameterizes the Dirichlet process, is often split into its total mass, M, and its shape, say F_0. Thus if α is a measure on the real line, F_0 is a distribution function, $M > 0$, and $\alpha((-\infty, x]) = MF_0(x)$.

G_θ is a distribution indexed by the parameter θ. The models are easily generalized to include hyperparameters that index α, groups of observations associated with each θ_i, observation specific covariates, and additional parameters common to some or all observations. The bioassay problem is one which fits into this framework.

There are three main types of MCMC methods that have been widely used for the mixture of Dirichlet process models. The first is based directly on the hierarchical model written above. It makes use of the sequence of conditional generations $[F|\theta], [\theta|F]$. See Kuo and Smith (1992), Gelfand and Kuo (1991) and, in a general setting, Ishwaran and Zarepour (2000) for details. See also Diebolt and Robert (1994) in the context of a related finite mixture model.

The second type of Markov chain method makes use of an alternative representation of the Dirichlet process known as the Polya urn scheme (Blackwell and MacQueen, 1973). Under the Polya urn scheme, the random distribution function F is marginalized, resulting in the model

$$\begin{aligned} \theta_1, \ldots, \theta_p &\sim F_{\theta_1,\ldots,\theta_p} \\ X_i|\theta_i &\sim G_{\theta_i}, \text{ for } i = 1, \ldots, p. \end{aligned}$$

To simplify description, take F_0 to be continuous. With this model, the components θ_i are no longer conditionally independent. Instead, they have a distribution that is built up sequentially: $\theta_1 \sim F_0$. For $i > 1$, θ_i is set equal to θ_j with probability $1/(M + i - 1)$ and is drawn from F_0, independent of previous draws from F_0, with probability $M/(M + i - 1)$. The induced distribution on the vector θ is often thought of in two parts. The first is the partition of θ into distinct values, and the second is the location of the, say k, elements of the partition. Each partition receives positive probability under the prior. Given a partition, the k locations of the elements, denoted $\theta_1^*, \ldots, \theta_k^*$, are i.i.d. draws from F_0. A Markov chain based on this representation of the model involves sequential generation of $[\theta_i|\theta_{-i}]$, for $i = 1, \ldots, p$, with the updating performed immediately in each case. See Escobar (1994) and Escobar and West (1995) for algorithms of this sort. These algorithms may be refined by discarding the locations of the clusters and running a Markov chain on only the space of partitions of θ (Neal, 1992; MacEachern, 1994). Such chains tend to produce quicker convergence to the posterior and naturally suggest better estimators. The calculations below refer to this last refinement of the algorithm, though they can be replicated when the locations are present.

The third type of algorithm is the split-merge algorithm with its ability to make large moves in directions not easily traveled in with algorithms of the first two types. The simple example can be fit with the simple split-merge algorithm of Jain and Neal (2000). In this case, the improvements in the algorithm do not change its performance.

The Markov chain runs on a state space which consists of all partitions of θ into clusters. This is a finite state space, which is denoted by S. An element in the state space is a p-dimensional vector, $s = (s_1, \ldots, s_p)$, with component s_i indicating to which cluster θ_i belongs. If there are k clusters of θ_i, there will be k distinct integers in the partition vector. If θ_i and θ_j are in the same cluster, $s_i = s_j$; if in different clusters, $s_i \neq s_j$. The fact that the state space is finite allows us to perform exact calculations on the transition matrix of the Markov chain in small examples. Several chains are compared for the case of $p = 3$. A major issue is the labelling of the state space. Two identifiable labellings and one non-identifiable labelling are considered. The labelling/identifiability issue is cleanest for Type II algorithms. The labellings are presented in Table 1.

The first Type II scheme numbers the clusters consecutively from 1 to k as they are built up from the Polya urn scheme. Thus $s_1 = 1$, and for all i for which $\theta_i = \theta_1$, $s_i = 1$. The second cluster is begun by the first $\theta_i \neq \theta_1$, and so $s_i = 2$ for $i = inf[j|\theta_j \neq \theta_1]$. All other θ_j equal to this θ_i are in this cluster and so are assigned $s_j = 2$. The numbering of the later clusters proceeds in a similar fashion, so that for a legitimate partition vector (i.e., one which receives positive probability under the prior) representing k clusters, the numbers 1 through k will appear and their first appearances will occur in increasing order. The final legitimate values of s for the case $p = 3$ appear in Table 1 under the heading scheme 1. With this parameter space, the model is identifiable. Each legitimate configuration vector produces a distinct partition of the θ and hence (under the mild regularity condition that there is a set of θ_i with positive F_0 probability such that $G_{\theta_1} = G_{\theta_2}$ iff $\theta_1 = \theta_2$) produces a distinct distribution for X.

The second Type II scheme is similar to the first in that there is a 1-1 mapping between partitions and legitimate configuration vectors. The difference is in how the clusters are labelled. Again, all θ_i in a cluster will have the same index in the configuration vector. Those θ_i in the cluster with θ_1 have $s_i = 1$. Further clusters have an index equal to $inf[j|\theta_j$ in cluster]. For example, define $i = inf[j|\theta_j \neq \theta_1]$. Then $s_j = i$ for all j such that $\theta_j = \theta_i$. The legitimate values for s under this labelling scheme when $p = 3$ appear in Table 1 under the heading scheme 2. Since there is a $1 - 1$ mapping between this labelling and the previous one, identifiability for this model follows from identifiability of scheme 1.

The third Type II scheme produces a non-identifiable model. With this scheme, the clusters will each receive a distinct integer from 1 to n, and each θ_i in a particular cluster will receive the same index. There is, however, no other restriction on the index values assigned to the clusters. To create this scheme formally, begin with the first labelling scheme. Probabilities of the legitimate states are determined by the Polya urn scheme. Then the probability for a particular configuration is distributed among the possible labellings for the configuration. For a configuration with k clusters, there are $n!/(n-k)!$ distinct labellings. The probability for this configuration is distributed uniformly among these labellings. This model is clearly non-identifiable, since there are several parameter values (here several different configuration vectors) which produce the same distribution for the data. Interestingly, $[F|\theta, X, s]$ depends on s only through the configuration. Hence, any inference depends only on the equivalence class on s defined by the configuration itself.

State	Configuration	scheme 1	scheme 2
a	$\theta_1, \theta_2, \theta_3$	1,2,3	1,2,3
b	$\theta_1 = \theta_2; \theta_3$	1,1,2	1,1,3
c	$\theta_1 = \theta_3; \theta_2$	1,2,1	1,2,1
d	$\theta_1; \theta_2 = \theta_3$	1,2,2	1,2,2
e	$\theta_1 = \theta_2 = \theta_3$	1,1,1	1,1,1

Table 1: Labellings of configurations under schemes 1 and 2.

Gibbs samplers were developed for each of the labelling schemes above for the no-data problem. The transition matrix for a fixed scan, in the order $[s_1|s_2, s_3]$, $[s_2|s_1, s_3]$, and then $[s_3|s_1, s_2]$ was calculated analytically. For the first two schemes, the second largest eigenvalue of the transition matrix was determined. To compare the third scheme to the first two, identifiable functions are considered. In order to determine an effective rate of convergence for these functions, the transition matrix for the sampler is rewritten in terms of an identifiable model. Happily, all of the transition vectors from each non-identifiable state corresponding to a particular configuration to the distinct configurations are identical (e.g., the transition probability for moving from the state $s = (1, 1, 3)$ to the configuration $\theta_1 = \theta_2 = \theta_3$ is the same as the transition probability for moving from the state $s = (2, 2, 1)$ to the configuration $\theta_1 = \theta_2 = \theta_3$). The chain, in terms of this identifiable state space, retains the Markov property. The implication is that the second largest eigenvalue of the rewritten transition matrix governs the rate of convergence in the identifiable space.

The three Gibbs samplers corresponding to the three labelling schemes were compared by means of the second largest eigenvalue of their transition matrices, presented in Table 2. The comparison of the three schemes shows that scheme 3, based on the non-identifiable model, produces the best performance. The non-identifiable model results in better mixing.

Simulations were carried out to compare the Type I algorithm to the Type II algorithms. The simulation made use of a non-identifiable version of the Type I algorithm. The estimated second largest eigenvalue of the Type I algorithm appears in Table 2 along the row labelled Type I. Scheme 3 appears to dominate this type of algorithm. This conclusion agrees with results that suggest a collapse of the state space improves the convergence rate of a Markov chain, since the scheme 3 algorithm may be constructed by adding generations to a Type I algorithm and then collapsing the state space. Interestingly, this is in opposition to the sometimes expressed intuition that a two-stage Gibbs sampler, as the Type I method, should show quicker convergence than a three-stage Gibbs sampler, as the scheme 3 algorithm is. These results in this simple context are in agreement with the careful simulations for more realistic settings in Papasiliopoulos and Roberts (2008).

The random scan Gibbs sampler was investigated in a similar fashion. Table 2 contains a summary of the results for 3 transitions (so chosen to match the three transitions

M	1	5	10	100
scheme 1	.222	.327	.389	.485
scheme 2	.222	.0408	.0139	.000192
scheme 3	.0370	.00292	.000579	9.42e-7
Type I	.301	.0837	.0332	.000559

M	1	5	10	100
scheme 1	.559	.630	.669	.726
scheme 2	.559	.395	.352	.303
scheme 3	.171	.0787	.0588	.0393
Split-merge	1.00	.152	.216	.287

Table 2: Second largest eigenvalues for MCMC algorithms. The top table is for fixed scan samplers; the bottom table is for random scan samplers. M is the mass of the base measure of the Dirichlet process.

of the fixed scan sampler). Notice that the second largest eigenvalues are considerably larger for random scan samplers, corresponding to the potentially long lags between successive sampling of a component. Again, scheme 3, corresponding to the non-identifiable model, is preferable to the Type II schemes. The Type III (split-merge) sampler is, for the larger values of M, preferable to the Type II samplers that impose identifiability. In this example, it does not mix as well as the non-identifiable algorithm. Interestingly, when $M = 1$, the sampler yields a periodic Markov chain, and so mixing is poor although estimation (barring an even subsampling rate) is fine. It should be noted that this periodicity is very special to this example.

4 Heuristics

The simplicity of the example allows us to focus on features of the algorithms that impact mixing: Comparisons among the Type II algorithms suggest that non-identifiability (of a certain sort) improves mixing; the comparison between fixed and random scans suggests that fixed scans lead to better mixing; a good Type II algorithm leads to better mixing than a Type I algorithm; for *small* clusters, the Type II algorithm mixes better than the Type III algorithm.

Within Type II algorithms, the example shows a remarkable advantage for the non-identifiable model. This appears to follow from the conditioning sets used to create the Gibbs sampler. The non-identifiable model leads to conditioning sets that contain the conditioning sets arising from the identifiable model. To illustrate this point, a schematic of the transition matrices is provided in Table 3. Comparing the two P_1's, for instance, under scheme 1 the transition matrix is the identity while under scheme 3 it is a block diagonal matrix with only two blocks. Both chains are based on conditional generations. For each current state, the set conditioned upon for the generation under

P_1											
From	To	a	b	c	d	e	a	b	c	d	e
a		x	-	-	-	-	x	x	x	-	-
b		-	x	-	-	-	x	x	x	-	-
c		-	-	x	-	-	x	x	x	-	-
d		-	-	-	x	-	-	-	-	x	x
e		-	-	-	-	x	-	-	-	x	x
P_2											
From	To	a	b	c	d	e	a	b	c	d	e
a		x	-	-	-	-	x	x	-	x	-
b		-	x	-	x	-	x	x	-	x	-
c		-	-	x	-	x	-	-	x	-	x
d		-	x	-	x	-	x	x	-	x	-
e		-	-	x	-	x	-	-	x	-	x
P_3											
From	To	a	b	c	d	e	a	b	c	d	e
a		x	-	x	x	-	x	-	x	x	-
b		-	x	-	-	x	-	x	-	-	x
c		x	-	x	x	-	x	-	x	x	-
d		x	-	x	x	-	x	-	x	x	-
e		-	x	-	-	x	-	x	-	-	x

Table 3: Scheme 1 transition matrices on the left, scheme 3 transition matrices on the right. The states are described in Table 1. A dash indicates that a transition cannot take place, an x that it can. Note the enlargement of the sets over which conditional generations take place with scheme 3.

the scheme 3 chain contains the set conditioned upon for the generation under the scheme 1 chain. Thus the conditioning sets for the scheme 1 chain are nested in those for the scheme 3 chain. The following result connects the nesting of conditioning sets to total variation distance.

Proposition 1. Suppose that we have a countable state space, and a distribution π which assigns positive probability to each state. Further suppose that this state space is partitioned into conditioning sets C_i. Define row i of the transition matrix P to consist of the distribution π, restricted to the conditioning set in which state i lies. Consider two partitions, A and B, where $\{C_{A,i}\}$ is a refinement of $\{C_{B,i}\}$ and the corresponding transition matrices P_A and P_B. Then, for any initial distribution, π_I, $||\pi_I^{'}P_A - \pi|| \geq ||\pi_I^{'}P_B - \pi||$.

Proof. The total variation distance between the distributions F and G is defined by $||F - G|| = sup_A(|F(A) - G(A)| + |F(A^C) - G(A^C)|)$ where A ranges over all subsets of the state space. When the initial distribution π_I is modified through a transition governed by a conditional distribution over a partition, the supremum is attained by a set A for which each element of the partition is either entirely contained in A or entirely contained in A^C. Since the conditioning sets used to create P_A are a refinement of those used to create P_B, we may view the supremum in the former case as being taken over a larger set. Hence, $||\pi_I^{'} P_A - \pi||$ is at least as large as $||\pi_I^{'} P_B - \pi||$.

Proposition 1 shows that one step of the chain based on larger conditioning sets (i.e., the sampler based on the non-identifiable model) is preferable to one step of the chain based on the smaller conditioning sets. However, the proof given here does not extend to more steps. Presumably, the quicker one-step movement toward the posterior will often carry over into a quicker rate of convergence for the chain, as it does in the example of Section 3. Consideration of the impact of identifiability underlay, in part, the development of nonconjugate algorithms in MacEachern and Muller (1998).

As Jain and Neal comment, the Type III algorithms are most beneficial when there are large clusters of observations. With only a few large clusters, all observations will frequently have a chance to switch clusters. However, my experience with models involving the Dirichlet process is that the posterior distribution typically includes a number of small clusters (in addition to the large clusters). The simple example suggests that including Type II steps is important to facilitate mixing for these small clusters.

5 Conclusions

The example presented herein, as well as others that I have examined, lead to the following viewpoint on the four reasons presented earlier for avoiding non-identifiable models. The first, interpretation of the model, has no connection to whether MCMC methods are used to fit the model, and so in no way suggests that one restrict themself to use of identifiable models. The second reason seems to be largely irrelevant. The important convergence rate (if an identifiable model is to be considered at all) is convergence for estimates of identifiable functionals. This may be quicker than the convergence rate of the chain in the non-identifiable space. In any event, if an effective chain can be created based on the identifiable form of the model, the same chain can be created based on the non-identifiable form of the model. The third concern, for numerical stability of the computations, remains a concern. The fourth issue is one of prior elicitation. Since models and prior distributions are subjective and situation specific, any recommendation for one form of model over another is open to criticism. Nevertheless, some classes of models seem much more natural than do others. Often, as in the case of the hierarchical model, these classes contain non-identifiable models. A decision to replace a natural, non-identifiable model with an identifiable model that seems to be less natural seems unwise without a demonstrated improvement in the ease or effectiveness with which the model is fit.

My own view on problems necessitating MCMC methods is this. One should first write down the most natural model, whether it be identifiable or non-identifiable. Next, lay out several MCMC methods for this version of the model. Further consider expanding the parameter space to create non-identifiable models. Particular consideration should be given to inducing non-identifiability by adding symmetries such as the relabelling of the clusters in the simple Dirichlet process example. Again, examine a batch of MCMC algorithms, with attention to generating blocks of parameters and to marginalizing parameters. Finally, select an algorithm based on the heuristics of preferring those derived from larger conditioning sets, those that have collapsed the state space, and those that generate blocks of parameters at a time. To this algorithm, add steps that target particularly difficult transitions–such as splitting and merging large clusters.

The hints in Jain and Neal's paper and the simple example suggest a natural direction for extension of the split-merge moves: a move away from a random scan (i.e., random selection of observations i and j that determine the attempted split/merge) and toward a scan with reduced randomness. The randomness of the scan can be lessened, for example, by permuting the indices from 1 through n, and using successive pairs for i and j. This type of permutation bounds the time between successive attempts at updating each observation's cluster membership. In turn, this ensures that the number of iterates until every observation-specific parameter has had a chance to be updated is controlled. I suspect that the benefits that Jain and Neal have demonstrated of combining both incremental and split-merge moves in an algorithm are partly due to the implicit reduction in randomness–a complete incremental Gibbs scan ensures that all cases have had the opportunity to move.

A second possible extension is to reserve the split/merge moves for clusters of substantial size. To do so, one could partition the parameter space into two parts–one part where the combined number of cases in clusters identified by observations i and j exceeds some threshold and the second part where the combined number of cases is small. If the current state were in the first part, a split-merge move would be attempted, and the state after transition would also remain in the first part. If the current state were in the second part, slightly modified incremental steps would be attempted, with the modification ensuring that the state after transition would also remain in the second part. Alternatively, for this second part, one could make no transition at all. With the posterior distribution invariant for each potential step, the posterior distribution would remain invariant for the chain as a whole. Supplementing this type of move with incremental Gibbs scans would yield irreducibility of the chain.

References

Antoniak, C. E. (1974). "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems." *The Annals of Statistics*, 2: 1152–1174.

Berry, D. A. and Christensen, R. (1979). "Empirical Bayes Estimation of a Binomial Parameter Via Mixtures of Dirichlet Processes." *The Annals of Statistics*, 7: 558–568.

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

Dahl, D. (2005). "Sequentially-allocated merge-split sampler for conjugate and nonconjugate Dirichlet process mixture models." Technical report, Department of Statistics, Texas A& M University.

Diebolt, J. and Robert, C. P. (1994). "Estimation of Finite Mixture Distributions through Bayesian Sampling." *Journal of the Royal Statistical Society, Series B: Methodological*, 56: 363–375.

Escobar, M. D. (1994). "Estimating Normal Means with a Dirichlet Process Prior." *Journal of the American Statistical Association*, 89: 268–277.

Escobar, M. D. and West, M. (1995). "Bayesian Density Estimation and Inference Using Mixtures." *Journal of the American Statistical Association*, 90: 577–588.

Ferguson, T. S. (1973). "A Bayesian Analysis of Some Nonparametric Problems." *The Annals of Statistics*, 1: 209–230.

Gelfand, A. E. and Kuo, L. (1991). "Nonparametric Bayesian Bioassay Including Ordered Polytomous Response." *Biometrika*, 78: 657–666.

Ishwaran, H. and Zarepour, M. (2000). "Markov Chain Monte Carlo in Approximate Dirichlet and Beta Two-parameter Process Hierarchical Models." *Biometrika*, 87(2): 371–390.

Jain, S. and Neal, R. M. (2000). "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model." Technical report, Department of Statistics, University of Toronto.

Kraft, C. H. and van Eeden, C. (1964). "Bayesian Bio-assay." *The Annals of Mathematical Statistics*, 35: 886–890.

Kuo, L. and Smith, A. F. M. (1992). "Bayesian Computations in Survival Models Via the Gibbs Sampler (Disc: P22-24)." In Klein, J. P. and Goel, P. K. (eds.), *Survival Analysis: State of the Art*, 11–22. Kluwer Academic Publishers Group.

Liu, J. S. (1994). "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem." *Journal of the American Statistical Association*, 89: 958–966.

Lo, A. Y. (1984). "On a Class of Bayesian Nonparametric Estimates: I. Density Estimates." *The Annals of Statistics*, 12: 351–357.

MacEachern, S. N. (1994). "Estimating Normal Means with a Conjugate Style Dirichlet Process Prior." *Communications in Statistics: Simulation and Computation*, 23: 727–741.

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

Neal, R. (1991). "Bayesian mixture modelling." In C.R. Smith, G. E. and Neudorfer, P. (eds.), *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, 197–211. Kluwer Academic Publishers.

Papaspiliopoulos, O. and Roberts, G. (2008). "Retrospective MCMC for Dirichlet process hierarchical models." *Biometrika(to appear)*.

Ramsey, F. L. (1972). "A Bayesian Approach to Bioassay (Com: V29 P225-226, V29 P830)." *Biometrics*, 28: 841–858.

Acknowledgments

This research was funded by the NSF under grant numbers SES-0437251 and DMS-0605041 and by the NSA under award number H98230-05-1-0065.