Bayesian Analysis (2007) **2**, Number 3, pp. 495–500

Rejoinder

Sonia Jain* and Radford M. Neal†

We thank discussants Drs. MacEachern, Robert, and Dahl for their thoughtful comments. Since many of their comments are related, we will address them by topic below.

1 Creation, Deletion, Identifiability, and Tempering

Our conditionally conjugate split-merge technique belongs to the family of trans-dimensional MCMC algorithms, which includes, for example, reversible-jump MCMC (Green 1995), birth-death MCMC (Stephens 2000a), and split-merge MCMC (Jain and Neal 2004), (Dahl 2003). Trans-dimensional MCMC algorithms construct Markov chain transitions between states that vary in dimension. For Dirichlet process mixture models, this involves the creation or deletion of mixture components.

Of course, even plain Gibbs Sampling updates for this model must be able to create and delete mixture components, but they do so only in an incremental fashion, in which a new component must start off explaining only a single observation — which may be a rather unlikely state. A key strength of trans-dimensional MCMC procedures is the ability to traverse the parameter space efficiently without having to pass through such low-probability states. For simple problems, these techniques can save computation time by reducing the required burn-in, and improving sampling thereafter. For more complex and difficult problems, such as are encountered in areas such as genetics and image analysis, these techniques may be essential if the problem is to be solved in any reasonable amount of time.

The mixture components created are given arbitrary labels, which could be permuted without affecting fit to the data, or prior probability. This "non-identifiability" has been seen by some as raising issues with regard to proper interpretation of the results, as discussed, for example by Stephens (2000b). These issues are of no relevance to our paper, which is concerned only with efficiently sampling from the posterior distribution. We agree with MacEachern that forcing the Dirichlet process mixture model to be identifiable is a hindrance to efficient MCMC sampling.

In this regard, one should note that sampling of all equivalent labellings can easily be obtained by simply introducing an additional MCMC update (applied at any desired interval) that permutes the labels — though this would be pointless for most purposes, since the labelling doesn't matter. Robert demonstrates that Gibbs sampling alone may fail to move easily between modes with different labellings. In itself, this failure is of no practical significance. Lack of movement between these equivalent modes should be

*Division of Biostatistics and Bioinformatics, Department of Family and Preventive Medicine, University of California at San Diego, La Jolla, CA, mailto:sojain@ucsd.edu

†Department of Statistics and Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, http://www.cs.toronto.edu/~radford/

ba0007

worrying only to the extent that one thinks it is a sign that the MCMC method would also fail to move between non-equivalent modes (if any) that correspond to substantively different interpretations of the data. It is unclear to us that failure to move amongst equivalent modes is actually indicative of a real problem of this sort. Conversely, there is no guarantee that a method that moves amongst equivalent modes can also move easily between non-equivalent modes.

Robert suggests that perhaps global tempering would perform better than a split-merge procedure, with regard to movement between isolated modes. (We are not sure which tempering method Robert used for his example, as it is not specified.) However, his example considers the benefits of tempering only when transitions are done using Gibbs sampling, without any split-merge updates. Moreover, his example concerns a fully conjugate model of the type treated in our earlier work (Jain and Neal 2004), rather than the nonconjugate models discussed in this article.

Also, the comparison looks only at mixing amongst equivalent modes, which as mentioned above is of no importance in itself. For these reasons, this demonstration does not convince us that tempering would work better than split-merge methods. However, we do expect that for some complex problems, such as very high-dimensional clustering, split-merge may not be sufficient. We hypothesize that tempering methods may also have difficulty with such problems, but that applying tempering in conjunction with split-merge updates might allow for their solution.

2 The Role of Conditional Conjugacy in Our Algorithm

As the discussants highlight, our split-merge method applies only to models in which the prior for parameters of component distributions exhibits conditional conjugacy. Though this limits the the usefulness of our algorithm, its domain is perhaps wider than one might expect. For instance, consider MacEachern (1998), in which he describes how a nonconjugate model can be treated as a conditionally conjugate problem by using piecewise log-concavity.

Robert wonders whether it might be possible to extend the algorithm beyond conditionally conjugate models. Conditional conjugacy is needed so that we can do a Gibbs sampling scan from the launch state, and also compute the probability density for choosing the value chosen at each stage of this scan. The underlying requirement is that we have a way of proposing a new parameter vector based on the launch state such that (a) the distribution of the proposed state is similar to the posterior distribution of parameter values, and (b) we can for this proposed state compute the probability density of its having been proposed. This allows us to implement efficient Metropolis-Hastings updates, with a particular update being chosen randomly by the procedure for selecting a launch state. As an aside, note that from its very origins (Metropolis et al. 1953) the Metropolis algorithm has commonly been used with proposals that change only a subset of the variables. It is not necessary to justify such partial Metropolis-Hastings updates in a special way (as Robert suggests), or to refer to such updates as anything other than Metropolis-Hastings updates.

An MCMC transition that leaves the posterior distribution invariant is a natural way of trying to get a proposal for parameters of split/merged components that comes from close to the posterior distribution. More than one such transition would be better, but would make computing the density for a proposal impossible, as that would require integrating over intermediate states. (Such an integral is avoided in our algorithm by treating the intermediate Gibbs sampling updates not as part of the proposal distribution but rather as a procedure for choosing a launch state.) One could certainly imagine using MCMC transitions other than Gibbs sampling for this purpose. One could, for example, use a series of Metropolis-Hastings updates applied to each parameter in turn. The probability density for proposing a state that differs in all components from the launch state would then be easily computed, as the product of all the proposal densities and all the acceptance probabilities. Unfortunately, the probability density for a state in which any of these Metropolis-Hastings updates was rejected (so that at least component is the same in the launch state and in the proposal) will be infinite, which will result in a zero acceptance probability for the split-merge update.

So, although one can imagine such variations, they may not be useful in practice. One possibility that would be worth investigating is using some approximation to the posterior distribution (for the model restricted to two components), such as a Gaussian. The conditional distributions from this approximation could be used as proposals (resulting in Gibbs sampling in the limit as the approximation becomes perfect). If the rejection rate is small enough, this might work well. Alternatively, the approximation could be used directly — the validity of our algorithm does not depend on the transition from the launch state (or the intermediate transitions) leaving the actual posterior distribution invariant, though use of a bad approximation will of course lead to a low acceptance rate for the split-merge updates.

3 The Usefulness of Incremental Markov Chain Updates Together with Split-Merge

Our split-merge algorithm has four tuning parameters, controlling the number of intermediate restricted Gibbs sampling scans for splits proposals and merge proposals, and the number of split-merge updates and incremental Markov chain updates (e.g. Gibbs sampling scans) done as part of a full iteration. Both MacEachern and Dahl remark on the importance of a final incremental Gibbs sampling scan. We agree with MacEachern that the inclusion of such a step is important to facilitate mixing, as we have demonstrated in the article. However, though MacEachern emphasizes the role of such updates in mixing for small clusters, we believe that they are at least as important for moving observations back and forth between large clusters, as this cannot be done efficiently with split-merge updates.

Indeed, as we have described in our article and as Dahl observes, the “jitter” that is observed in the trace plots of the beetle example can be attributed to the final auxiliary Gibbs sampling scan. However, we disagree with Dahl’s conclusion from this that the CPU time spent on split-merge updates is “wasted” when these moves are not accepted. How can we know *a priori* that these moves will not be accepted unless they are proposed? Since split-merge updates are required to obtain a correct solution (in a reasonable amount of time) for some problems, it is necessary to perform them for all problems in order to determine if they are actually needed, and hence ensure that the answer obtained is correct.

Further, Dahl's demonstration with only auxiliary Gibbs sampling (no split-merge updates) is not entirely convincing. On close inspection, the lower plots of his Figure 1 show that auxiliary Gibbs sampling is not actually performing that well! For several thousand iterations, a number of observations seem to have been incorrectly allocated to small clusters, with the Gibbs sampler making only slow progress in correcting this. It is possible that just a few split-merge iterations could take care of these orphan clusters. By performing both incremental and non-incremental split-merge updates, one can take advantage of both large-scale changes to the cluster configuration via split-merge moves and small-scale adjustments that move a few observations between clusters, as is necessary depending on the problem.

4 MCMC Initialization

Robert suggests that sampling from the prior to initialize the intermediate restricted Gibbs sampling could lead to wasted computational effort. In higher-dimensional problems, we agree that overcoming bad initial values could be a problem — i.e. many restricted Gibbs sampling scan might be required. In the Discussion section of the paper, we had suggested alternatives to sampling from the prior to initialize the restricted Gibbs sampling, such as adapting a method used by Dahl (2003), or some other posterior estimation method.

A feature of the split-merge technique that Dahl discusses is its insensitivity to the initial value that the Markov chain is started with, whereas the Gibbs sampler is susceptible to poor choices (as illustrated in the Beetle example). We agree with this, but are puzzled by the discrepancies in the simulations. We also initialized the chain by sampling the model parameters from the prior and not by setting the initial values to the sample mean and precision. One possible explanation is differing orders of updates — we sampled the indicators first and then the model parameters (means before precisions).

5 Random versus Fixed Scan Sampling

MacEachern investigates in detail how MCMC performance differs for fixed versus random scans, in the context of Gibbs sampling. He proposes a systematic scan as an alternative to the random scan that we utilize to initiate the split-merge process (i.e. select two observations, denoted as i and j, uniformly at random). MacEachern suggests permuting the indices from 1 to n and then using successive pairs as i and j, thereby reducing randomness. This gives a feasible scan length (unlike systematically using all possible pairs of observations). We agree that this is likely to improve performance, but perhaps not by much. Partly, this is because there will still be considerable randomness in which *clusters* are chosen for split/merge operations — in particular, the same clusters might well be chosen several times in a row.

More generally, however, MacEachern may be overestimating the difference between fixed and random scans. The interpretation of his Table 2 is perhaps not obvious. The number of iterations required to reach some small total variation distance is proportional to $-1/\log(v)$, where v is the second-largest eigenvalue. So, for example, using scheme 3, with $\alpha = 1$ (which is $M = 1$ in MacEachern's notation), the fixed scan method is not better by a factor of $0.171/0.037 = 4.6$, as one might naively think, but rather by a factor of $\log(0.037)/\log(0.171) = 1.9$. As α approaches infinity (approximated by $\alpha = 100$, i.e. corresponding to $M = 100$ in the table), the second largest eigenvalue for the fixed scan approaches zero — all the variables are independent, so a single fixed Gibbs sampling scan immediately reaches equilibrium. The random scan has a non-zero second eigenvalue in the $\alpha \to \infty$ limit, reflecting the fact that after any number of iterations there is a non-zero probability that some variable could still be left unchanged. Technically, the asymptotic convergence rate of the fixed scan is infinitely better than that of the random scan, but in practice a modest number of iterations is sufficient to give the correct result with very high probability.

In this small example, Markov chain sampling is based on the prior distribution of clusterings for three data points, but the likelihood factors deriving from the data are omitted. However, in practical problems, where many split-merge proposals are likely to be made for each that is accepted, the randomness in choice of clusters to split/merge may be negligible compared to the randomness in proposing how to split or merge them, and in whether or not to accept the result. Finding ways of further improving the split/merge proposals may be a better focus for future research.

References

Dahl, D. B. (2003). "An improved merge-split sampler for conjugate Dirichlet process mixture models." Technical Report 1086, Department of Statistics, University of Wisconsin. 495, 498

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82: 711–732. 495

Jain, S. and Neal, R. M. (2004). "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model." *Journal of Computational and Graphical Statistics*, 13: 158–182. 495, 496

MacEachern, S. N. (1998). "Computational methods for mixture of Dirichlet process models." In Dey, D., Müller, P., and Sinha, D. (eds.), *Practical Nonparametric and Semiparametric Bayesian Statistics*, 23–43. New York: Springer-Verlag. 496

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equation of state calculations by fast computing machines." *Journal of Chemical Physics*, 21: 1087–1092. 496

Stephens, M. (2000a). “Bayesian analysis of mixtures with an unknown number of components – an alternative to reversible jump methods.” *Annals of Statistics*, 28: 40–74. 495

— (2000b). “Dealing with label-switching in mixture models.” *Journal of the Royal Statistical Society, Series B*, 62: 795–809. 495