An Exploration of Random Jump Markov Chain Monte Carlo

1st John Gillen
Department of Statistical Science
Duke University
Durham, United States
jg583@duke.edu

Abstract

We explore random jump Markov chain Monte Carlo and compare it to standard Gibbs samplers and Metropolis-Hastings algorithms for the identification of Gaussian mixtures. We conduct a visual comparison of posterior indicators on various data, and implement a previously proposed convergence diagnostic for RJMCMC, and compare it against its fixed-component analog for MH and Gibbs.

Index Terms

RJMCMC, transdimensional MCMC

I. Introduction

Gaussian mixture models (GMMs) are powerful tools for modeling data with underlying multimodal distributions. They are ubiquitous in applications such as clustering, density estimation, and image segmentation. Methods like Expectation-Maximization (EM) and K-means have been used to estimate the parameters of GMMs, but they face challenges when dealing with low sample sizes or irregular clusters.

Markov chain Monte Carlo (MCMC) methods offer a probabilistic approach to parameter estimation and often perform better in exploring complex, high-dimensional parameter spaces. However, as we have seen this semester, conventional methods face challenges in dealing with GMMs, such as label switching and mode trapping. There are ways around these issues, of course, but they tend to revolve around the idea of choosing a fixed number of components. However, the number of components may not always be known or may not be easy to guess with confidence.

Reversible Jump MCMC (RJMCMC), first proposed by Peter Green [1], addresses these issues by treating the number of components k as unknown and allowing it to vary. The algorithm then proceeds as normal (mixture) Metropolis-Hastings, but includes an additional proposal for a new k. Waagepetersen and Sorensen [2] provide a very in-depth derivation of the algorithm. We will merely give a brief overview:

Setup

We have data $\mathbf{x}$ and a set of mixture models indexed by k, where each model has parameters θ_k. A k-component mixture model:

$$p(\mathbf{x} \mid k, \theta_k) = \sum_{j=1}^{k} \pi_j f(x_i \mid \phi_j),$$

with $\theta_k = (\pi_1, \ldots, \pi_k, \phi_1, \ldots, \phi_k)$, has joint posterior:

$$\pi(k, \theta_k \mid \mathbf{x}) \propto p(\mathbf{x} \mid k, \theta_k) p(k, \theta_k).$$

Within-Model Moves (Metropolis–Hastings)

Propose θ_k' from $q(\theta_k' \mid \theta_k)$ and accept with probability:

$$\alpha_{\text{within}} = \min\left\{1, \frac{\pi(k, \theta_k' \mid \mathbf{x}) q(\theta_k \mid \theta_k')}{\pi(k, \theta_k \mid \mathbf{x}) q(\theta_k' \mid \theta_k)}\right\}.$$

Between-Model Moves (Reversible Jump)

Introduce an auxiliary variable u and a bijection $(\theta_k, u) \leftrightarrow \theta_{k+1}$. Propose $(k+1, \theta_{k+1})$ from (k, θ_k) and accept with probability:

$$\alpha_{\text{between}} = \min\left\{1, \frac{\pi(k+1, \theta_{k+1} \mid \mathbf{x}) q(k, u \mid k+1, \theta_{k+1})}{\pi(k, \theta_k \mid \mathbf{x}) q(k+1, u \mid k, \theta_k)} \left|\frac{\partial(\theta_{k+1})}{\partial(\theta_k, u)}\right|\right\}.$$

This ensures correct sampling over both models and parameters (note that we may also propose $(k-1, \theta_{k-1})$ if $k > 1$).

We hope to explore here the viability of RJMCMC for GMMs by comparing it to Gibbs and Metropolis-Hastings approaches in terms of posterior predictive accuracy (where applicable), convergence speed, and runtime. We use the Acidity [3], Galaxies [4][5], and Fish [6][7] datasets. Acidity contains 155 observations of lake acidity in Wisconsin, Galaxies has 82 observations of velocities (in 1000 km/s) of galaxies diverging from the Milky Way, and Fish contains 256 observations of lengths of fish. Additionally, we compare these methods on randomly generated R data using the 'rnormmix' function with sample size 200 unless otherwise specified.

II. Previous Work

GMMs seem like a natural use case for this technique. Green and Richardson [8] used RJMCMC for a fully Bayesian framework on GMMs in an unknown-k setting with weak priors, with promising results.

Upon discovering this method, I was surprised that I had never heard of it before. However, further reading led me to discover the problem with RJMCMC methods: diagnostics. Traditional diagnostics for MCMC are hard to extend to RJ methods, as they aren't generalized to handle changes in k. Brooks and Giudici [9] proposed a two-way ANOVA decomposition to test for differences between variances of a function of model parameters $\boldsymbol{\theta}^+$ between multiple chains. This was later expanded by Castelloe and Zimmerman [10] to lessen the power of infrequently visited sets of parameters by using weighted MANOVA. Both of these were attempts to generalize the Gelman-Rubin statistic, also known as the potential scale reduction factor (PSRF) or $\hat{R}$ to the variable-k case. However, the proposed diagnostics both have difficulties with the change in or loss of parameter interpretability that RJMCMC can introduce. Sisson and Fan [11] derived a distance-based diagnostic, which we implement here. Their diagnostic uses a point-to-nearest-event distance to map the variable-component parameter vectors into one-dimensional space, defining the distance as the distance from a reference point v (sampled from the parameter space) to the nearest component of the Markov chain, at each iteration i and for each reference point v. The empirical CDF $F(x; v)$ is the distribution of the distance from v to the nearest component across all chain realizations, effectively reducing the Markov chain output to a family of univariate distribution functions for each reference point. From here, discrepancies between chains are assessed by comparing the ECDFs across chains for the same reference point. Then the PSRFv is just the usual Gelman-Rubin statistic applied at reference point v:

$$\hat{R}_v = \left(\frac{N-1}{N} W_v + \frac{1}{N} B_v \right)^{1/2},$$

where B_v and W_v are respectively the within- and between-chain variances.

III. Methods

We implement RJMCMC using the 'uvnm.rjmcmc' function from the 'miscF' R library, which is based on the method described by Richardson and Green [8]. Gibbs and Metropolis-Hastings are both implemented using custom functions to handle GMMs. These only differ from the standard implementations in that k must be pre-set, we add a weight parameter, and we sort the means in ascending order after each iteration (this was added later after noticing that label switching was making convergence almost impossible). As stated previously, we use $n = 200$ for all synthetic data. Also, we run all chains for 10,000 iterations, with burn-in periods of 500 iterations each.

We first perform a strictly visual evaluation of these methods' fits to various data, and examine posterior predictive intervals for means. For the fixed-dimensional algorithms, k is chosen based on visual inspection of a histogram of the data. We use weak priors for RJMCMC: $k_0 = 1, mu_0 = \overline{x}, \sigma_0^2 = \text{Range}_{data}$ (as opposed to starting with the same k chosen for the fixed methods and trying to visually estimate vectors for the other parameters). Priors for the other two algorithms follow the same pattern. Posterior distributions for RJMCMC fits are generated by finding the most probable k after running the chain, and then sampling from only iterations with the corresponding k, as was previously done by Richardson and Green [8]. When extracting the RJMCMC posterior, if the two most probable k are sufficiently close in probability, we choose the lower one, as earlier testing showed RJMCMC could sometimes be prone to overfitting due to the relatively small sample sizes. We compare the posterior intervals to the true parameters (indicated by black, dashed vertical lines) if they are known.

For evaluation of convergence, we run 4 chains of each type, using the multivariate PSRF for Gibbs and MH, and the distance-based PSRFv proposed by Sisson and Fan [11] for RJMCMC (all with custom implementations). To generate reference points for the latter, we implement a custom R function to sample new points at each iteration. The points are generated for each chain, and the authors [11] recommend using about 100 total, so we generate 25 points per chain. The convergence plots include a orange dotted line at $y = 1.05$ (sorry if it's hard to see!), which is a generally-accepted threshold to indicate convergence (though some may argue for 1.10, or even 1.01).

Lastly, runtime comparison is straightforward. We use the 'microbenchmark' library to simulate each type of chain on the Gravity data for 10,000 iterations, 100 times each, and then again on the S1 data for 100 reps of 10,000 iterations each. We use two datasets of different sizes to see whether the relationship between the algorithms' runtimes varies with n.

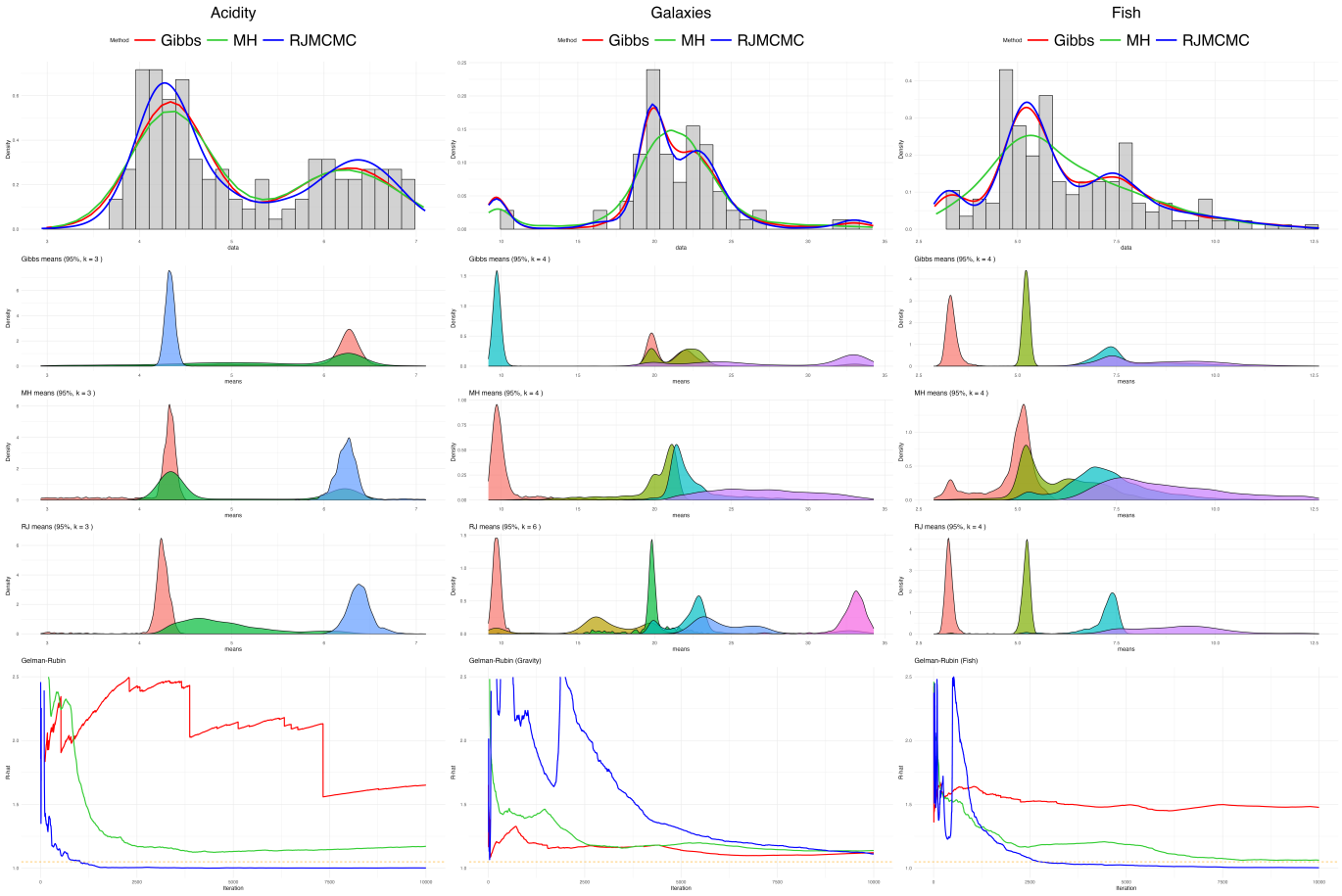

Fig. 1. Performance on Real Data (x-axes of posterior plots are identical to the data, vertical order is Gibbs-MH-RJ, and range is 0-10,000 for x in the Gelman-Rubin plots). Also, Gibbs is red, MH is green, RJ is blue

IV. Results

In general, the posterior parameter distributions tend to be similar, especially for the RJ and Gibbs chains. MH appears to be averse to 'believing' in true multimodality without lots of evidence, whereas RJ tends to be easily convinced to add another component (coming up with a dubious-looking 6 as the most-probable k for the Galaxies data). Perhaps due to this phenomenon, the posterior intervals for MH tend to have lots of overlap. There is also a clear issue of label switching for all chains, but especially the fixed-dimensional ones, and especially whenever posterior component means are close together.

Convergence is clearly both faster and better in RJMCMC compared to the other two, with Galaxies being a strange exception, likely due to the very low sample size leading to relatively high probabilities for $k = 4, 5$, and 6, which in turn leads to diverging chains. Even so, it is clear that good convergence would likely have been achieved within another 1,000 reps or so (and this was an unlucky seed). There is also a very strange pattern with the Gibbs sampler's PSRF for the Acidity data. We observed on numerous occasions that the Gibbs sampler would either converge slightly faster than RJ, diverge outside of the range of the plot (see S1 in 2), or exhibit a pattern like this. We believe this is due to the Gibbs sampler's vulnerability to getting stuck in a single mode, and the sudden decreases correspond to the time at which a chain 'escaped' from that mode. Conversely, the sometimes-instant convergence could be attributed to all chains following similar paths (however, even in those cases, the chains would sometimes 'un-converge').

Examining the synthetic data is much more interesting, because in these cases the true parameters are known. Starting with S1, we see that RJ manages to identify the fourth component, and appears to capture the other three with marginally higher certainty (though this is difficult to discern). It again manages to identify that there is a 5th component in S2, but fails to capture said component's mean with any certainty. Interestingly, both Gibbs and RJ are 'fooled' by the small mode in the right tail, while MH gives a more conservative posterior estimate (although it also has a bimodal posterior on the leftmost component). S3 and S4 expose some weakness of RJMCMC. In the former, we see that although it appears to do a better job distinguishing between the three true components (which wasn't easy), it identifies a fourth one due to a single outlier (I

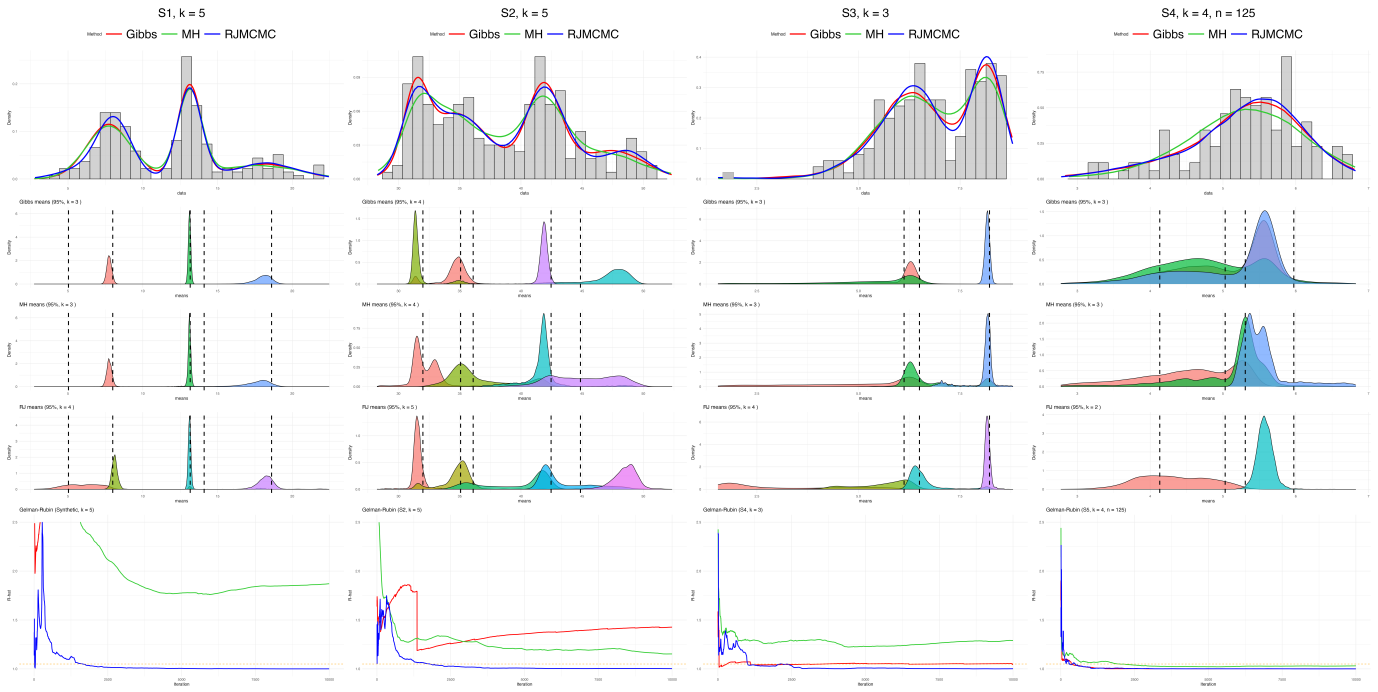

Fig. 2. Performance on Synthetic Data

had to add this to the histogram because it wouldn't appear for some reason). The S4 data, which has a sample size of 125 as opposed to the 200 for the other three, has some interesting results. Firstly, RJMCMC only identifies two components. It gets fooled by the overlap between the two rightmost components and ultimately generates a posterior interval for the mean of the discerned component that almost completely misses both true means.

RJMCMC also wins massively in runtime (though this is possibly due to suboptimal code in my custom MH and Gibbs functions), with an even greater margin on larger samples ($n = 82$ for Galaxies and 200 for S1), with the difference in times implying that RJMCMC runs in a factor of $O(\log(n))$ time, while Gibbs and MH are around $O(n)$ 3. However, it is worth noting that if we consider the full process, including generating reference points and computing PSRFv, MH and Gibbs both run quite a bit faster overall (sadly there isn't time for a thorough comparison due to how long it would take to execute, but my estimate is that the complete RJ/PSRFv process with 25 reference points and 4 chains has about triple the combined total runtime of the analogous MH and Gibbs procedures). However, given the vast difference in convergence rate and consistency, and the overall superior performance in identifying components in mixture models (to say nothing of classifying observations), I think the extra runtime is well worth it.

V. Conclusions

Metropolis-Hastings appears to perform the worst out of the three methods. In the visual tests, it seems to have great difficulty conforming to a shape with more than two modes (though this also makes it the least prone to overfitting), and its posterior predictive distributions are almost always substantially overlapping (both of these flaws were even worse before the label switching correction). When evaluating convergence, it almost never reaches the desired threshold of 1.05 within 10,000 iterations, although it is admittedly more consistent than the Gibbs sampler in this regard. The Gibbs sampler exhibits quite similar behavior to RJMCMC in terms of posterior predictive fit and estimation, but has difficulty with classification, leading to the same distributional overlap we mentioned with MH. However, in some cases this wouldn't be a problem, as identifying the component parameters can be more important than classifying observations (the latter can be done separately). Gibbs has very inconsistent convergence, but this may again not be the worst thing in the world, as the Gibbs samplers were consistently able to come up with reasonable posterior distributions. RJMCMC appears to dominate these two methods in almost every sense: it always converges better (and almost always converges faster) while allowing for variability in an additional dimension, which enables it to usually better identify component parameters. It runs more than twice as fast (and even scales better with larger samples, diagnostics notwithstanding). The only apparent weaknesses of the random jump method are its tendency to be biased by small sample sizes into either adding components that don't (or may not) exist or missing components that are close together and the runtime of the reference point/PSRFv process.

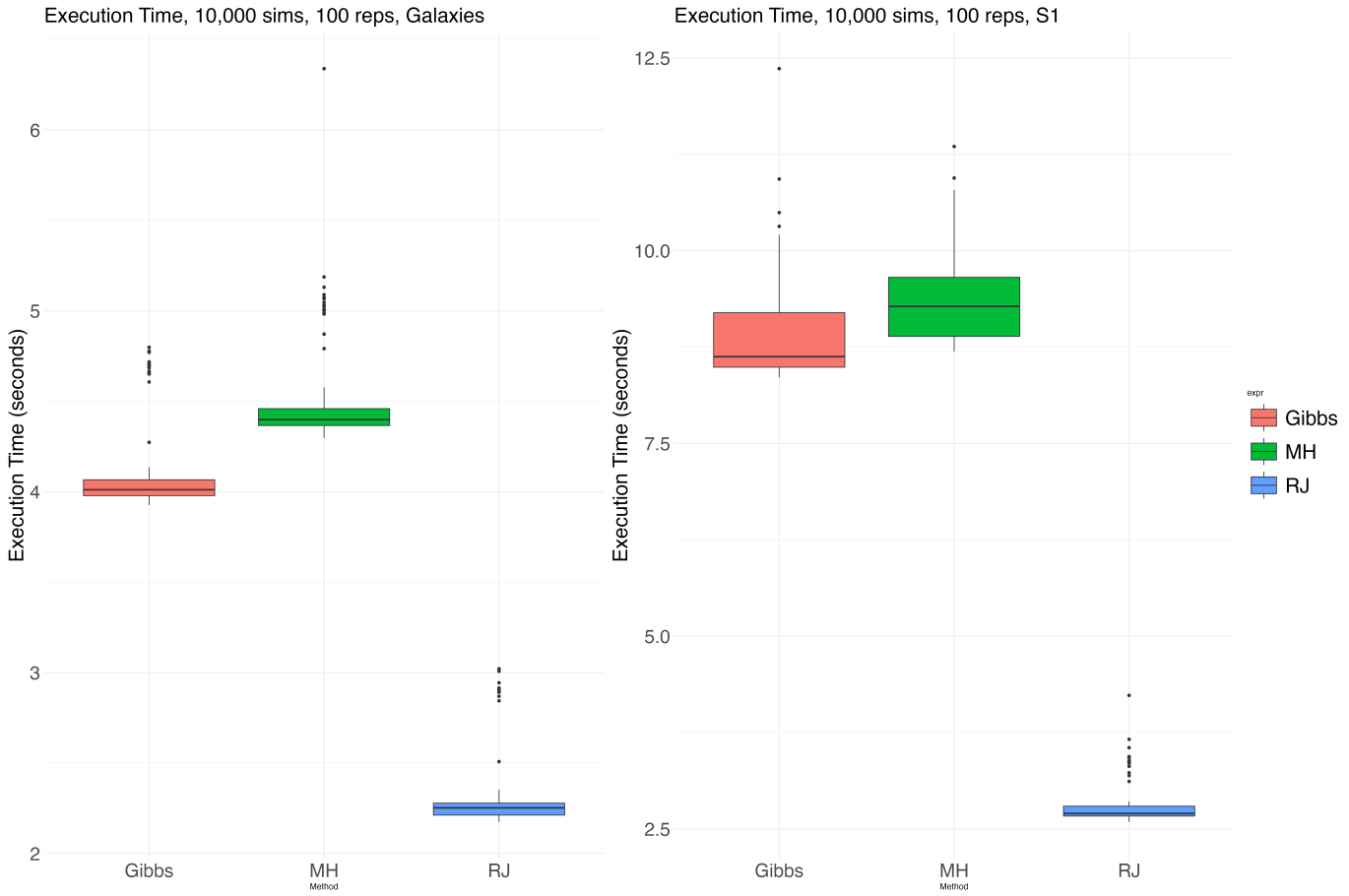

Fig. 3. Runtime Comparison

A. *Future Work*

Much of the difficulty of this project came with figuring out how to implement the convergence diagnostics in a way that would allow for proper comparison between these two different types of chains, and also with implementing and fine-tuning functions to make initializing, running, visualizing, and evaluating each algorithm a smoother process. Some work could definitely be done with respect to optimizing the code. More compelling though would be work on coming up with better ways of comparing trans- and finite-dimensional methods, and thoroughly assessing the validity of my own method. In something between a choice and a logistical disaster, other conventional diagnostics like traceplots and ESS were not used in this project. I wonder how these could be generalized to variable dimensions. RJMCMC appears to be a rather unexplored topic, at least for GMMs, especially considering how well performs. It would be interesting to explore more of the theory behind this, as the method I used to measure convergence is nearly 20 years old. Even if there is no further work (which I very much doubt), I would like to spend some time experimenting with different prior parameterizations to see if some of the small-sample weaknesses can be shored up.

REFERENCES

[1] Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[2] Rasmus Waagepetersen and Daniel Sorensen. A tutorial on reversible jump mcmc with a view toward applications in qtl-mapping. *International Statistical Review*, 69(1):49–61, 2001.

[3] Susan L. Crawford, Morris H. DeGroot, Joseph B. Kadane, and Mitchell J. Small. Modeling lake chemistry distributions: approximate bayesian methods for estimating a finite mixture model. *Technometrics*, 34(4):441–453, 1992.

[4] Reza Mohammadi. *bmixture: Bayesian Estimation for Finite Mixture of Distributions*, 2021. R package version 1.7.

[5] Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. *Annals of Statistics*, 28(1):40–74, February 2000.

[6] Bettina Gruen and Martyn Plummer. *bayesmix: Bayesian Mixture Models with JAGS*, 2023. R package version 0.7-6.

[7] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, Chichester, UK, 1985.

[8] Sylvia Richardson and Peter 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.

[9] Stephen P. Brooks and Paolo Giudici. Mcmc convergence assessment via two-way anova. *Journal of Computational and Graphical Statistics*, 9(2):266–285, 2000.

[10] John M. Castelloe. Convergence assessment for reversible jump mcmc samplers. Technical report, University of North Carolina at Chapel Hill, March 2002. Technical Report.

[11] S. A. Sisson and Y. Fan. A distance-based diagnostic for trans-dimensional markov chains. *Statistics and Computing*, 17:357–367, 2007.