Bayesian Analysis (2016) **11**, Number 3, pp. 945–955

Rejoinder*

Matthew T. Pratola†

1 Introduction

We would first like to sincerely thank the Editors of Bayesian Analysis for inviting this work to be discussed, and for the many discussants who provided interesting and insightful feedback on this work: Bobby Gramacy, Christopher Hans, Luca Martino, Rafael B. Stern, Francisco Louzada, Scotland C. Leman, Andrew Hoegh, Reihaneh Entezari, Radu V. Craiu, Jeffrey S. Rosenthal, A. Mohammadi, M. C. Kaptein and O. Chkrebtii. Thank you! As there were many shared themes amongst the discussants, our rejoinder is organized along the following topics: Alternatives to Metropolis–Hastings, Data Subsetting, Reversible-Jump MCMC (RJMCMC), Priors and Adaptation.

2 Alternatives to Metropolis–Hastings

Many discussants proposed alternatives to the Metropolis–Hastings approach to sampling the posterior, including Multiple Try Metropolis and Combinatorial Sequential Monte Carlo (Martino et al., 2016), the Multiset Sampler (Leman and Hoegh, 2016), Birth–Death MCMC (Mohammadi and Kaptein, 2016) and Parallel Tempering (Chkrebtii, 2016). Generally, these algorithms are "blind" in the sense that they do not exploit the structure of tree-space explicitly to try and move efficiently amongst good trees. However, since they are designed to improve the sampling of general MCMC algorithms they are more widely applicable, albeit possibly with increased computational cost. In combination with the moves proposed in this work one might reasonably expect to see further improvements, or at least faster convergence of the sampler.

2.1 Multiple Try Metropolis

We agree with the suggestions of Martino et al. (2016) to also consider introducing more general sampling algorithms designed to improve mixing, such as Multiple Try Metropolis (MTM) or Combinatorial Sequential Monte Carlo (C-SMC), or combining some of these algorithms with ideas introduced in the paper. The tradeoff of using these general algorithms to improve mixing is usually computational cost which we would prefer to avoid, but good combinations of methods could likely improve things further. In particular, the MTM method seems likely to be a good candidate, and we thank the discussant for this suggestion.

In the MTM method (Martino and Read, 2013; Liu et al., 2000), instead of a single proposal being drawn at each Metropolis–Hastings step, a sample of k states are pro-

*Main article DOI: 10.1214/16-BA999.
†Department of Statistics, The Ohio State University, mpratola@stat.osu.edu

 DOI: 10.1214/16-BA999REJ

posed and one is selected according to appropriately calculated weights that maintain the detailed balance condition. Following Liu et al. (2000), the steps are:

1. Draw k proposals $\tau_1, \ldots, \tau_k$ from $Q(\tau, \cdot)$ and compute $w(\tau_j, \tau)$ for each $j = 1, \ldots, k$.
2. Select $\tau' = \tau_j$ with probability proportional to $w(\tau_j, \tau)$.
3. Draw $k-1$ "reference points" $\tau_1^*, \ldots, \tau_{j-1}^*, \tau_{j+1}^*, \ldots, \tau_k^*$ from $Q(\tau', \cdot)$ and set $\tau_j^* = \tau$ and compute $w(\tau_j^*, \tau')$ for each $j = 1, \ldots, k$.
4. Accept the proposal τ' with probability

$$\alpha = min\left\{1, \frac{\sum_{j=1}^{k} w(\tau_j, \tau)}{\sum_{j=1}^{k} w(\tau_j^*, \tau')}\right\}.$$

In the standard MTM approach (Liu et al., 2000), $w(\tau, \tau') = \pi(\tau)Q(\tau, \tau')\lambda(\tau, \tau')$ for a suitable user-specified function $\lambda(\tau, \tau')$ (more variations are explored in Martino and Read (2013)). The idea then of the MTM is to more adequately explore the space around the current state τ in order to choose proposals τ' that have larger weight.

A natural fit for the MTM idea in tree sampling is in the dimension-changing birth/death and rotate proposals. These both choose proposals uniformly at random, which is likely an inefficient choice. For instance, of the possible trees constructed by the rotation proposal, one is uniformly selected at random. Birth/Death moves are similarly simplistic in their approach to choosing tree topology in the sense that the terminal (or next-to-terminal) node to perform a Birth/Death is uniformly selected at random. Implementing MTM for these proposals would allow the choice to be done with non-uniform weight, which may further improve the acceptance of dimension-changing moves. While there would be a computational cost to pay, it may be manageable in some cases, such as BART where trees are constrained to be a reasonable size by the depth-penalizing prior.

2.2 Birth–Death MCMC

The suggestion by Mohammadi and Kaptein (2016) of a continuous-time approximation of RJMCMC is a very interesting one, and is not an approach that has been considered in any capacity by those working in Bayesian tree models as far as we know. While making use of these ideas in Bayesian regression trees may require greater modification than it initially appears, such an unexplored approach is bound to lead to interesting developments.

In the approach outlined in Mohammadi and Wit (2015), the discussants consider undirected graphical models $G = (V, E)$ where $V = 1, \ldots, p$ is the set of nodes and $E \subset V \times V$ is the set of existing edges while the set of non-existing edges is defined as $\bar{E}$. They define a Gaussian graphical model as

$$\mathcal{M}_G = \{\mathcal{N}_p(0, \Sigma) | K = \Sigma^{-1} \in \mathcal{P}_G\},$$

where $\mathcal{P}_G$ denotes the space of $p \times p$ matrices with entries $(i, j) = 0$ for all $e \in \bar{E}$. The continuous-time Birth/Death process is defined via the birth rate

$$\beta_e(K) = \frac{P(G^{+e}, K^{+e} \setminus (k_{ij}, k_{ji})|\text{data})}{P(G, K \setminus k_{jj}|\text{data})}, \text{ for each } e \in \bar{E},$$

and death rate

$$\delta_e(K) = \frac{P(G^{-e}, K^{-e} \setminus k_{jj}|\text{data})}{P(G, K \setminus (k_{ij}, k_{jj})|\text{data})}, \text{ for each } e \in E.$$

Based on these rates, the Birth–Death MCMC iterates the following steps:

1. Calculate the Birth rates $\beta_e(K)$ and $\beta(K) = \sum_{e \in \bar{E}} \beta_e(K)$
2. Calculate the Death rates $\delta_e(K)$ and $\delta(K) = \sum_{e \in E} \delta_e(K)$
3. Calculate the waiting time $W(K)$
4. Simulate the Birth/Death jump
5. Sample a new precision matrix, K

where $W(K) = \frac{1}{\beta(K)+\delta(K)}$, $P(\text{birth at edge e}) = W(K)\beta_e(K)$ and $P(\text{death at edge e}) = W(K)\delta_e(K)$.

The difficulty would seem to lie in particular with step (1) in the algorithm as this calculation involves a sum over all edges not currently in the tree. In the Gaussian graphical model defined by Mohammadi and Wit (2015), the size of this space is finite since p is assumed fixed, known. But in Bayesian regression trees, the dimensionality of tree-space is potentially infinite.

Another aspect is how one chooses which edges to birth/death. In the graphical model described, birth/death can occur at any edge but for trees we require connectedness to the existing tree structure. Such a constraint may be able to be incorporated into the general framework described in Mohammadi and Wit (2015), but does not appear to be a straightforward implementation of the method described which assumes birth/death of an edge occurs independently of the others. Nonetheless, we do agree that this would be a very unique and fresh perspective on the sampling problem.

2.3 The Multiset Sampler

Another good suggestion is the Multiset Sampler (MSS) recommendation of Leman and Hoegh (2016). In this approach, instead of sampling directly from the distribution of interest, one samples instead from the equally-weighted mixture

$$\pi^*(f|x, y) = C \sum_{(t,m) \in f} p(t, m|x, y),$$

where k denotes how many elements are in the Multiset in the MSS. The user selected parameter k denotes how many simultaneous states exist in the MSS. Thus, k in some sense allows for one to account for more multi-modal posteriors, with increasing k able to handle posteriors of greater complexity. Expectations of interest can then be extracted from $\pi^*(f|x, y)$ as discussed in Kim and MacEachern (2015). A limitation is the increased computations required. For instance, calculating posterior expectations requires the calculation of weights for the entire collection of trees – for BART this would be $m \times k$ weights for each posterior draw if the same multiset size is used for each tree in BART. Nonetheless, we agree that the MSS is another good choice for improving mixing. For the single-tree case, Leman et al. (2009) demonstrated improvements over the Bayesian CART algorithm of Chipman et al. (1998). Furthermore, since the MSS algorithm also depends on a proposal distribution, we agree that combining the MSS with the proposals discussed in the paper may lead to further improvements.

2.4 Parallel Tempering

Finally, Chkrebtii (2016) suggests applying the parallel tempering algorithm of Geyer (1991). We agree that this is also a good suggestion. Since the parallel tempering algorithm also depends on proposals, we can combine the results of this paper with this general algorithm to realize further improvements in mixing. We thank the discussant for presenting this algorithm in the context of Bayesian regression trees. We also note that parallel tempering has been explored in other tree models, as mentioned by Gramacy (2016) who previously devised a related importance tempering approach (Gramacy et al., 2010).

3 Data Subsetting

Another alternative to the Metropolis–Hastings sampler falls under data-subsetting parallel algorithms, such as the LISA method recently proposed in Entezari et al. (2016) which builds on ideas introduced in the Consensus Monte Carlo algorithm of Scott et al. (2016). These approaches split the dataset into equal-sized subsets, independently run an MCMC algorithm to learn the posterior on each subset, and then recombine the subset posteriors in some way to form an overall estimate of the posterior. Such data subsetting methods often do in fact make use of standard MCMC algorithms in each batch, but gain computing efficiency from their "embarrassingly-parallel" construction, assuming the recombining of posteriors after the MCMC algorithms have completed can be performed efficiently and accurately. As a side effect, because each subset is analyzed independently, better exploration of the model space may be achieved compared to fitting the model using a single MCMC run on all the data.

The LISA algorithm partitions the entire dataset of size N into K batches of approximately equal size and then samples the $j = 1, \ldots, K$ sub-posteriors,

$$\pi_{j,LISA}(\vec{\theta}|\vec{y}^{(j)}) \propto \left(f(\vec{y}^{(j)}|\vec{\theta})\right)^K p(\vec{\theta}),$$

which are then recombined in some fashion. Entezari et al. (2016) demonstrate similar statistical performance as BART on the Friedman test function with $N = 20{,}000$ observations and $K = 30$ batches, while achieving a 10-fold computational speedup. We wholeheartedly welcome such incredibly useful (and practical!) developments for analyzing big datasets, and as with some of the other alternatives suggested by other discussants, one would expect that combining methods would lead to further improvements.

A possible challenge with such batch-wise data-parallel methods is how one should choose the number of batches, K. For the example given, the jth sub-posterior is trained using a dataset size of ~ 666 observations for the 5-dimensional Freidman regression problem. A priori, it is unclear if this is a "good" choice or not (or if it even matters). Some alternatives avoid this issue, such as the single-chain data-parallel algorithm of Pratola et al. (2014), which relies on low-dimensional sufficient statistics to limit the communication overhead in parallel computations.

4 Reversible-Jump MCMC

Gramacy (2016) correctly points out that the defacto assumption in the paper is of terminal-node conjugacy, which greatly simplifies the sampling of tree-space. While this is a common, widely-used trick in the Bayesian tree literature, it is not always a feasible assumption. A particular example is the popular treed Gaussian Process (GP) model of Gramacy and Lee (2008). In their approach to birth/death proposals, Gramacy and Lee (2008) make use of a simple dimension-changing RJMCMC proposal mechanism to jointly sample the tree structure and the smoothness parameters of the GP correlation function. Given a GP correlation function $R(\mathbf{x}_i, \mathbf{x}_j|\mathbf{d})$ for hyperparameters $\mathbf{d} = (d_1, \ldots, d_D)$ they perform birth proposals by choosing one of the new nodes to randomly inherit the parents' correlation parameters, while the other child draws from the prior, i.e.

$$\mathbf{d}_{(1)} = u\mathbf{d} + (1-u)\mathbf{d}_*$$

$$\mathbf{d}_{(2)} = (1-u)\mathbf{d} + u\mathbf{d}_*$$

$$u \sim Bernoulli(0.5)$$

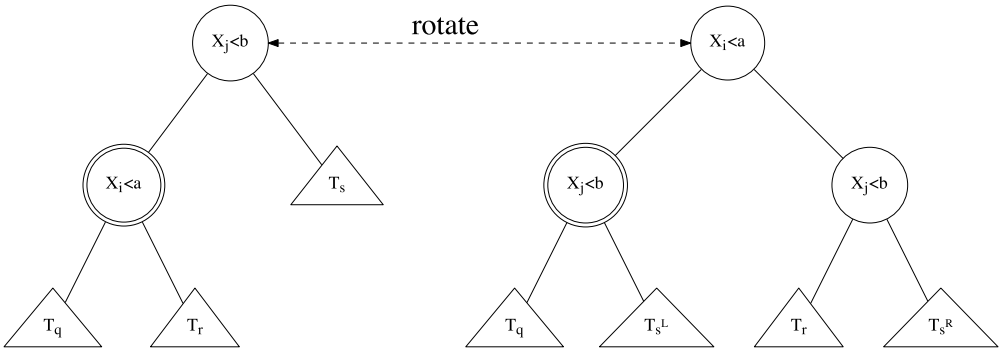
$$\mathbf{d}_* \sim \pi(\mathbf{d}),$$

which they show has unit Jacobian in the RJ-MCMC accept/reject calculation. Analogously, in a death proposal, one of the child-nodes' correlation parameters is uniformly chosen at random to be retained in the lower-dimensional model while the other is discarded.

In the case of rotate, extending this simple framework may be challenging for a few reasons. First, arriving at a good RJ-MCMC framework can be non-trivial and, unfortunately, depend on the type of bottom-node model being explored. Second, when the change of dimension is greater than 1 we necessarily have a greater number of parameters to jointly propose, which is increasingly more difficult. In some sense, this

problem is also present in Gramacy and Lee (2008) as there are D correlation length-scale parameters to handle in dimension-changing moves. Gramacy and Lee (2008) simplify this by avoiding scalar proposals in such moves.

The notion of locality can be important in getting RJ-MCMC proposals to work well. In the case of rotate, there are a few ways one might exploit the structure of a rotate move to implement RJ-MCMC. First, looking back at Figure 3 in the paper (which is reproduced below), it is clear that after the first rotate, T_q and T_r are unchanged unless a merge along X_j occurs. Therefore, deterministically mapping terminal node parameters of T_q, T_r in rotate proposals is often possible. One should also note that, of all the merge types that are possible, only one (type 3) reduces the effective number of terminal nodes in the merged tree. In other words, there may be a large degree of overlap between terminal nodes of the original and rotated tree, effectively reducing the number of "new" parameters to be handled. This suggests a similar approach to handling rotate proposals as the RJ-MCMC algorithm of Gramacy and Lee (2008).



The parts that need more careful handling are T_s^L and T_s^R that arise as a result of splitting. Note that if the rules in T_s do not include the variable X_i (top-right of Figure 3) then the number of terminal nodes in T_s^L and T_s^R equals the number of terminal nodes in the original T_s, so the dimension has increased by a possibly large amount (ignoring possible merges with T_q, T_r along X_j). On the other hand, if the rules in T_s consist of only the variable X_i then the total number of terminal nodes in $T_s^L \cup T_s^R$ may not be much larger than T_s alone. In fact, the dimension may increase by as little as a single terminal node, which would suggest a procedure similar to the birth case of Gramacy and Lee (2008).

Given these insights on the rotate proposal, a general procedure along the following lines may be feasible:

- For all terminal nodes in the rotated tree, say $\{\eta'_a, \eta'_b, \ldots\}$, repeat the following steps:

    1. Find the nearest terminal node $\{\eta_a, \eta_b, \ldots\}$ in the original tree (we suggest a procedure to do this below).

2. Propose a new parameter value m'_j according to some mapping g. Typically, g may involve a deterministic assignment, a draw from the prior, or perhaps a draw from some other distribution (e.g. Brooks et al. (2011) outline many advanced approaches such as mixtures and regression-based ideas).

Assuming the move from $(T, M) \to (T', M')$ increases dimension by $\delta = |M'| - |M| > 0$, the accept-reject calculation for such a procedure must be modified from the original equation (4) of the paper to:

$$min\left(1, \frac{\pi(T')\pi(M'|T')p_r(T')p_s^1 p_s^2 L(T', M')}{\pi(T)\pi(M|T)p_r(T)p_m^1 p_m^2 L(T, M)q_\delta(M_*)} \left|\frac{\partial g(M, M_*)}{\partial(M, M_*)}\right|\right),$$

where the new state $M' = g(M, M^*)$ where M^* is generated from known δ-dimensional distribution $q_\delta(M^*)$ (the reverse move would be generated deterministically with inverse acceptance probability). Here, $L(T, M)$ may simply be the likelihood, or it may be a partially integrated likelihood when some, but not all, terminal node parameters can be analytically integrated out.

In step 1, we mean only terminal nodes that are within the subtree being affected by the rotate move (other nodes would be updated by the usual MH step for the parameters). Of these nodes, by nearest terminal node we mean nodes that are arrived at by similar mappings – i.e. the sequence of rules from the tree root to terminal node η'_j defines a subregion in covariate space that is (nearly) equivalent to the sequence of rules from tree root to some terminal node, say η_i, in the original tree. This is most easily accommodated by applying labels, say $a, b, \ldots$, in terminal nodes before rotation and then matching nodes according to labels.

For instance, if two terminal nodes labelled a and b, say η_a, η_b with parameters m_a, m_b are merged in the rotate proposal, the label of the merged terminal node is ab. Calling this node in the merged tree η'_{ab} with parameter m'_{ab}, one could deterministically determine m'_{ab} by randomly choosing to assign it m_a or m_b from the "nearest" nodes in the original tree, which are η_a, η_b. Similarly, if a terminal node η_a is duplicated in the rotate proposal, the label of the duplicated nodes may be $a1, a2$. Then one can generate the proposals for parameters m'_{a1}, m'_{a2} of nodes η'_{a1}, η'_{a2} in the rotated tree by randomly selecting one of the parameters to be assigned m_a and the other to be generated from some distribution q, i.e.

$$\begin{aligned} m'_{a1} &= um_a + (1-u)m_* \\ m'_{a2} &= (1-u)m_a + um_* \\ u &\sim Bernoulli(0.5) \\ m_* &\sim q. \end{aligned}$$

Therefore, in the typical case there is a clear analogy to the birth/death RJ-MCMC proposal of Gramacy and Lee (2008). However, in the worst case one must more carefully track the labelled nodes. For instance, it is possible for node η_a in the original tree to be duplicated and one copy to be merged with either node b or c in the rotated tree, giving either nodes η'_a, η'_{ab} or η'_a, η'_{ac} in the rotated tree (although unlikely, such a situation

could occur if $T_s^L = T_s^R = T_s$ in Figure 3 and then a non-trivial merge occurs along one of the $X_j < b$ subtrees). The easiest approach in the first case may be to deterministically assign $m'_a = m_a$ and $m'_{ab} = m_b$ (similarly for the second case), although more elaborate schemes might be preferred.

Besides the situation just described, one may desire more elaborate proposal schemes in general as a means to improve performance. Without elaborating, we make some suggestions that may be helpful to consider. First, if no data maps to η'_j, this suggests drawing m'_j from the prior. If the number of observations mapping to η'_j, say n'_j, is the same as in the original tree, say node η_i with n_i observations, then a deterministic mapping (or something very near the original tree's value m_i) is suggested. Another alternative is to consider a mixture between m_i and the prior distribution, with weights that are adjusted by the relative proportion of observations remaining in the rotated terminal node.

One might also like to leverage the tree's recursive dividing of covariate space in an alternative nearest-neighbour approach by finding the sibling terminal node of η'_j in the rotated tree, say η'_k, that shares the same parent (i.e. $p(\eta'_j) = p(\eta'_k)$). If node η'_k corresponded to node η_i in the original tree according to our labelling scheme, this might suggest a mixture involving m_i. In the Gaussian case, mixture proposals in RJ-MCMC have been previously explored, see for instance Richardson and Green (1997).

Nearest-neighbour ideas leveraging the division of covariate space induced by trees may also be useful in regular MH proposals, such as the updating of GP correlation parameters in perturb proposals as mentioned by Gramacy (2016). For RJ-MCMC, leveraging existing ideas as much as possible is likely beneficial, requiring one to only decide what information should enter the RJ-MCMC proposal motivated by the structural rearrangements induced by the rotate move. Still, such "automatic" RJ-MCMC mechanisms are likely to be tedious to implement, and may not be suitable in all problems.

5 Priors

Comments on prior distributions in tree models also received some attention by the discussants (Gramacy, 2016; Chkrebtii, 2016). We agree that this is an area where much improvement can be made, and is one we have put considerable thought into as well. Gramacy (2016) asks why the CGM prior (Chipman et al., 1998) is so popular. A better way of framing the question is to ask what the CGM prior means. Penalizing tree depth essentially implies a less variable, or less complex, response behaviour as well as a preference against high-order interactions. The sum-of-trees BART model allows for more complex response behaviours so long as the additive representation is plausible, while still preferring a low (or no) order of interactions. Clearly, this prior allows one to constrain the type of solution desired.

Further (or alternative) constraints are certainly viable. Gramacy (2016) notes that the prior of Denison et al. (1998) penalizes the number of splits, while the prior of Wu et al. (2007) additionally adds a preference for balanced trees. These approaches can

also limit the complexity of the response. We agree with Gramacy (2016) that the CGM prior seems preferred in practice. In some sense, the direct penalization of number of terminal nodes (or splits) is a very CART-like perspective while the CGM prior seems more akin to a function-space view of the world. Yet in certain single-tree situations, perhaps ones prior preference may vary (or may not include any of the standard choices).

Chkrebtii (2016) suggests to limit the total number of variables allowed to split. We agree that this is a very interesting idea, implying a variable-selection prior in the spirit of the Lasso (Tibshirani, 1996; Hans, 2009). In the regression setting of BART, it may also be preferable to specify a prior on the number of trees rather than having it fixed. One might also explore a prior on the number of unique variables each tree is allowed to split on (e.g. 90% of trees can split on ≥ 1 variable, 50% on ≥ 2 variables, 20% on ≥ 3 variables, etc), thereby allowing one to decompose variability along main effects, 2-way interactions, and so on. Clearly there is much that can be explored!

6 Adaptation

Hans (2016) points out the possibility of adapting further aspects of the proposal mechanisms developed in the paper, specifically the change-of-variable proposal. We agree this is possible and perhaps very useful in certain problems. For instance, one may want to propose a change of variable at some internal node η_j that exploits local properties of the data rather than the global covariate correlation. If we denote $\mathbf{P}$ to be the global transition proposal probabilities based on the Spearman rank correlation assumed in Equation (7) of the paper, and $\mathbf{P^{(j)}}$ to be a locally-dependent transition matrix, then one possibility would be to form

$$\widetilde{\mathbf{P}} = \rho\mathbf{P} + (1-\rho)\mathbf{P^{(j)}}$$

(subsequently suitably normalized) for some $\rho \in (0,1)$. One might then form $\mathbf{P^{(j)}}$ based on some data-free local measure of covariate correlation (for instance), and adapt the scalar weight ρ as the algorithm learns the local properties of the data at each node. The challenge is in forming such a $P^{(j)}$ at each interior node in a computationally efficient manner. Such a general and efficient procedure may be difficult to arrive at. We agree that Hans (2016)'s suggestion of the method of Smith and Kohn (2005) would also be an interesting avenue to explore.

References

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press. MR2742422. doi: http://dx.doi.org/10.1201/b10905. 951

Chipman, H., George, E., and McCulloch, R. (1998). "Bayesian CART Model Search." *Journal of the American Statistical Association*, 93(443): 935–960. MR1631325. doi: http://dx.doi.org/10.2307/2670105. 948, 952

Chkrebtii, O., Leman, S. C., Hoegh, A., Entezari, R., Craiu, R. V., Rosenthal, J. S., Mohammadi, A., Kaptein, M. C., Martino, L., Stern, R. B., and Louzada, F. (2016). "Contributed Discussion on Article by Pratola. *Bayesian Analysis*, doi: http://dx.doi.org/10.1214/16-BA999H. 945, 946, 947, 948, 949, 952, 953

Denison, D., Mallick, B., and Smith, A. (1998). "A Bayesian CART algorithm." *Biometrika*, 85(2): 363–377. MR1649118. doi: http://dx.doi.org/10.1093/biomet/85.2.363. 952

Entezari, R., Craiu, R. V., and Rosenthal, J. S. (2016). "Likelihood Inflating Sampling Algorithm." arXiv:1605.02113. 948

Geyer, C. (1991). "Markov chain Monte Carlo maximum likelihood." In *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface*. American Statistical Association. 948

Gramacy, R. and Lee, H. (2008). "Bayesian treed Gaussian process models with an application to computer modeling." *Journal of the American Statistical Association*, 103(483): 1119–1130. MR2528830. doi: http://dx.doi.org/10.1198/016214508000000689. 949, 950, 951

Gramacy, R. B. (2016). "Comment on article by Pratola." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/16-BA999A. 948, 949, 952, 953

Gramacy, R. B., Samworth, R. J., and King, R. (2010). "Importance tempering." *Statistics and Computing*, 20: 1–7. MR2578072. doi: http://dx.doi.org/10.1007/s11222-008-9108-5. 948

Hans, C. (2009). "Bayesian Lasso regression." *Biometrika*, 96: 835–845. MR2564494. doi: http://dx.doi.org/10.1093/biomet/asp047. 953

Hans, C. M. (2016). "Comment on article by Pratola." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/16-BA999B. 953

Kim, H. and MacEachern, S. (2015). "The generalized multiset sampler." *Journal of Computational and Graphical Statistics*, 24: 1134–1154. MR3432933. doi: http://dx.doi.org/10.1080/10618600.2014.962701. 948

Leman, S., Chen, Y., and Lavine, M. (2009). "The multiset sampler." *Journal of the American Statistical Association*, 104: 1029–1041. MR2750235. doi: http://dx.doi.org/10.1198/jasa.2009.tm08047. 948

Liu, J. S., Liang, F., and Wong, W. H. (2000). "The multiple-try method and local optimization in metropolis sampling." *Journal of the American Statistical Association*, 95: 121–134. MR1803145. doi: http://dx.doi.org/10.2307/2669532. 945, 946

Martino, L. and Read, J. (2013). "On the flexibility of the design of multiple try Metropolis schemes." *Computational Statistics*, 28: 2797–2823. MR3141364. doi: http://dx.doi.org/10.1007/s00180-013-0429-2. 945, 946

Mohammadi, A. and Wit, E. C. (2015). "Bayesian structure learning in sparse Gaussian graphical models." *Bayesian Analysis*, 10: 109–138. MR3420899. doi: http://dx.doi.org/10.1214/14-BA889. 946, 947

Pratola, M., Chipman, H., Gattiker, J., Higdon, D., McCulloch, R., and Rust, W. (2014). "Parallel Bayesian additive regression trees." *Journal of Computational and Graphical Statistics*, 23: 830–852. MR3224658. doi: http://dx.doi.org/10.1080/10618600.2013.841584. 949

Richardson, S. and Green, P. J. (1997). "On Bayesian analysis of mixtures with an unknown number of components (with discussion)." *Journal of the Royal Statistical Society, Series B*, 59: 731–792. MR1483213. doi: http://dx.doi.org/10.1111/1467-9868.00095. 952

Scott, S. L., Blocker, A. W., and Bonassi, F. V. (2016). "Bayes and big data: The consensus Monte Carlo algorithm." *International Journal of Management Science and Engineering Management*, 11(2): 78–88. doi: http://dx.doi.org/10.1080/17509653.2016.1142191. 948

Smith, M. and Kohn, R. (2005). "Nonparametric regression using Bayesian variable selection." *Journal of Econometrics*, 75: 317–343. doi: http://dx.doi.org/10.1016/0304-4076(95)01763-1. 953

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society, Series B*, 58: 267–288. MR1379242. 953

Wu, Y., Tjelmeland, H., and West, M. (2007). "Bayesian CART: Prior specification and posterior simulation." *Journal of Computational and Graphical Statistics*, 16(1): 44–66. MR2345747. doi: http://dx.doi.org/10.1198/106186007X180426. 952