Bayesian Analysis (2016) **11**, Number 3, pp. 929–943

Contributed Discussion on Article by Pratola*

Comment by Oksana A. Chkrebtii[1]

Abstract. Pratola (2016) introduces a novel proposal mechanism for the Metropolis–Hastings step of a Markov chain Monte Carlo (MCMC) sampler that allows efficient traversal of the space of latent stochastic partitions defined by binary regression trees. Here we discuss two considerations: the first is the use of the new proposal mechanism within a population Markov chain Monte Carlo sampler (Geyer, 1991) to further increase sampling efficiency in the presence of greatly separated posterior modes, the second is a prior model that favors parsimony for the problem of variable selection.

Keywords: population Markov chain Monte Carlo, model selection, Bayesian treed regression.

We congratulate the author on an important contribution to sampling methodology for Bayesian treed regression. The joint posterior over the binary tree partition and model parameters is notoriously difficult to explore with existing local proposal mechanisms (Chipman et al., 2010; Wu et al., 2007; Gramacy and Lee, 2008). The rotation proposal step introduced by Pratola (2016) is important because it allows movement between disjoint high posterior density regions that arise when sampling regression tree structures. Given that this is a very challenging sampling problem, we suggest incorporating the proposal mechanism into a sampling scheme that can quickly move between posterior modes. We also note that improved ability to sample a possibly multimodal posterior allows us to consider the problem of variable selection, alluded to in the first motivating example in the paper where confounded variables tended to be added to the model together. To overcome this issue, we propose a conditional prior specification for the split variables that favors model parsimony.

1 Population MCMC with efficient tree proposals

Population MCMC methods (Geyer, 1991) allow both local and global transitions by simulating a number of auxiliary MCMC chains targeting progressively tempered posterior densities, and swapping their states with probability ξ_s. The Markov chain targeting the untempered posterior can thus explore greatly separated regions of the parameters space efficiently relative to a single chain. In Algorithm 1 we incorporate the proposal of Pratola (2016) in a parallel tempering sampler targeting the posterior distribution $[\tau, \{(v_i, c_i)\}, \sigma^2, \mu \mid y]$. Symbols with subscripts in parentheses correspond to a single chain. We use an expanded notation for the likelihood to make clear the dependence on sampled parameters at each step. The user defines the vector of temperatures $\boldsymbol{\gamma} \in (0, 1]^C$

*Main article DOI: 10.1214/16-BA999.
[1]Department of Statistics, The Ohio State University, Columbus, OH, USA, oksana@stat.osu.edu

 DOI: 10.1214/16-BA999H

Algorithm 1 Parallel tempering tree sampling algorithm using C Markov chains.

For $c = 1, \ldots, C$ construct $T_{(c)} = (\tau_{(c)}, \{(v_i, c_i)\}_{(c)})$ using the algorithm of Chipman et al. (2010), then sample $\sigma^2_{(c)}, \mu_{(c)}$ from the conditional prior;
for $m = 1 : M$ **do**
  if $\xi_s > \mathrm{U}[0,1]$ **then**
    Propose a swap between the index pair (i, j) drawn from a symmetric proposal distribution $q(i,j), 1 \le i, j \le C, i \ne j$ and compute the ratio,

$$\rho = \frac{L(y \mid \tau_{(i)}, \{(v_i,c_i)\}_{(i)}, \sigma^2_{(i)}, \mu_{(i)})^{\gamma_i}}{L(y \mid \tau_{(i)}, \{(v_i,c_i)\}_{(i)}, \sigma^2_{(i)}, \mu_{(i)})^{\gamma_j}} \frac{L(y \mid \tau_{(j)}, \{(v_i,c_i)\}_{(j)}, \sigma^2_{(j)}, \mu_{(j)})^{\gamma_j}}{L(y \mid \tau_{(j)}, \{(v_i,c_i)\}_{(j)}, \sigma^2_{(j)}, \mu_{(j)})^{\gamma_i}};$$

    if $\min(1, \rho) > \mathrm{U}[0,1]$ **then**
      Swap $(\tau_{(i)}, \{(v_i,c_i)\}_{(i)}, \sigma^2_{(i)}, \mu_{(i)}) \leftrightarrow (\tau_{(j)}, \{(v_i,c_i)\}_{(j)}, \sigma^2_{(j)}, \mu_{(j)})$;
    end if
  end if
  for $c = 1 : C$ **do**
    if $\xi_r > \mathrm{U}[0,1]$ **then**
      Construct $T'_{(c)} = (\tau'_{(c)}, \{(v'_i, c'_i)\}_{(c)})$ by performing a birth/death or rotation on $T_{(c)} = (\tau_{(c)}, \{(v_i,c_i)\}_{(c)})$ and compute the tempered ratio:

$$\rho = \frac{\pi(T_{(c)}) p_r(T_{(c)})\, p^1_m\, p^2_m\, L(y \mid \tau_{(c)}, \{(v_i,c_i)\}_{(c)}, \sigma^2_{(c)}, \mu_{(c)})^{\gamma_c}}{\pi(T'_{(c)}) p_r(T'_{(c)})\, p^1_s\, p^2_s\, L(y \mid \tau'_{(c)}, \{(v'_i,c'_i)\}_{(c)}, \sigma^2_{(c)}, \mu_{(c)})^{\gamma_c}};$$

      if $\min(1, \rho) > \mathrm{U}[0,1]$ **then**
        Update $T_{(c)} \leftarrow T'_{(c)}$;
      end if
    end if
    if $\xi_p > \mathrm{U}[0,1]$ **then**
      For $i = 1, \ldots, |T_{(c)}|$, with probability proportional to equation (7), propose $(v'_i, c'_i)_{(c)}$ by a performing a perturb or perturb within change-of-variable proposal q in equation (6), and compute the tempered ratio:

$$\rho = \frac{\pi(\{(v_i,c_i)\}_{(c)})\, q(v'_i, c'_i \mid v_i, c_i)\, L(y \mid \tau_{(c)}, \{(v_i,c_i)\}_{(c)}, \sigma^2_{(c)}, \mu_{(c)})^{\gamma_c}}{\pi(\{(v'_i,c'_i)\}_{(c)})\, q(v_i, c_i \mid v'_i, c'_i)\, L(y \mid \tau_{(c)}, \{(v'_i,c'_i)\}_{(c)}, \sigma^2_{(c)}, \mu_{(c)})^{\gamma_c}};$$

      if $\min(1, \rho) > \mathrm{U}[0,1]$ **then**
        Update $\{(v_i,c_i)\}_{(c)} \leftarrow \{(v'_i, c'_i)\}_{(c)}$;
      end if
    end if
    For $j = 1, \ldots, |M_{(c)}|$, draw $\mu_{j(c)}$ from $[\mu_j \mid y, \tau_{(c)}, \{(v_i,c_i)\}_{(c)}, \sigma^2_{(c)}]$
    Draw $\sigma^2_{(c)}$ from $[\sigma^2 \mid y, \tau_{(c)}, \{(v_i,c_i)\}_{(c)}, \mu_{(c)}]$
  end for
  Save the state, $(\tau_{(C)}, \{(v_i,c_i)\}_{(C)}, \sigma^2_{(C)}, \mu_{(C)})$, of the final chain.
end for

with the last element equal to one, and the probabilities ξ_r and ξ_p of rotating a single tree or perturbing a single variable, respectively. Note that the sampling over all non-swap candidate chains can be performed in parallel at every iteration.

2 Prior for model selection

In the context of estimation, the prior of Chipman et al. (2010) acts against over-fitting the data by penalizing the depth of internal nodes while putting uniform weights on the possible indices, $\{1, \ldots, d\}$, of the split variables v. For the problem of model selection we instead suggest a prior that encourages parsimony, penalizing the *number of distinct variables* along which splits are made. This can be accomplished by introducing prior dependence among the split variables v. We may define the prior on v_i conditionally such that its distribution should put most of its mass on the unique values of all the less deep nodes $v_1, \ldots, v_{i-1}$ with the remaining prior mass uniformly distributed among all variables on which splits have not yet been made. As with the prior of Chipman et al. (2010), draws from the proposed model selection prior can be constructed sequentially and have a convenient closed form. As suggested in the first motivating example, this prior choice may result in more sharply defined posterior modes in cases when variables are confounded. However, the very efficient proposal strategy of Pratola (2016), combined with particle MCMC may nevertheless be able to identify these modes.

References

Chipman, H., George, E., and McCulloch, R. (2010). "BART: Bayesian Additive Regression Trees." *The Annals of Applied Statistics*, 4(1): 266–298. MR2758172. doi: http://dx.doi.org/10.1214/09-AOAS285. 929, 930, 931

Geyer, C. (1991). "Markov chain Monte Carlo maximum likelihood." In *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface, 156*. American Statistical Association. 929

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. 929

Pratola, M. T. (2016). "Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian Regression Tree Models." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/16-BA999. 929, 931

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. 929

Comment by Scotland Leman[1] and Andrew Hoegh[2]

We commend Professor Pratola on the well written article "Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian Regression Tree Models". This paper nicely addresses the benefits of Bayesian regression tree models, outlines their typically cumbersome MCMC convergence properties under basic proposal structures (Chipman et al., 1998; Gramacy and Lee, 2008), and provides insights for constructing proposal mechanisms for improved mixing.

While the novel tree *rotation* and improved *rule perturbation* proposals presented in the main article lead to improved MCMC mixing behavior, we discuss an alternative approach relying on a the multiset sampler (MSS), which is a novel data augmentation sampler that can improve Markov chain Monte Carlo (MCMC) efficiencies even under typically inefficient proposals (Leman et al., 2009). Using a similar notation to the author's, we let (T, M) define the space of tree *topologies* and *maps*, respectively. We define a random multiset F, of size k, on (T, M) having density:

$$q(f|x,y) = q((t_1,m_1),(t_2,m_2),\ldots,(t_k,m_k)|x,y) \equiv C \sum_{(t,m)\in f} p(t,m|x,y), \qquad (1)$$

where (x, y) are data, $p(\cdot|\cdot)$ is the posterior distribution on trees, and C is a normalizing constant. We note that the trees (k in total) in Equation (1) are arbitrarily ordered, and each may appear with multiplicity. The induced normalizing constant is $C = \binom{|T|+k-1}{|T|}^{-1}$, where $|T| < \infty$ denotes the number of possible tree topologies (see Leman et al. (2009) for details).

The multiset augmentation defined in Equation (1) defines a sum-of-trees representation, but differs from that imposed by the BART algorithm (Chipman et al., 2010). The MSS proceeds through a Metropolis-within-Gibbs sampler that: 1) selects a tree (t, m) at random from multiset f, 2) from (t, m), proposes (t^*, m^*), 3) replaces (t, m) in f with (t^*, m^*) to form f^*, and 4) applies a typical Metropolis–Hastings decision rule to move from f to f^*. While relatively simple to implement, the MSS has the ability to explore tree spaces efficiently, even when the underlying proposal mechanisms are relatively local. The MSS works since $q(f^*|x, y)$ only differs from $q(f|x, y)$ by one tree, so f^* will always have a reasonable chance of being accepted. In general, at least one tree will be exploring a high probability region of the tree space at a time, while other trees are free to explore. It should be noted that because trees are in a multiset, the MSS does not converge to the target distribution $p(T, M|x, y)$, but rather a related, flatter distribution. Kim and MacEachern (2015) show how resampling from $q(f|x, y)$ into $p(t, m|x, y)$ can easily recover the true tree probabilities.

We conclude by presenting an example that first appeared in Chipman et al. (1998) which follows the model $y \sim N(\mu(x_1, x_2), 2^2)$, where:

[1]Department of Statistics, Virginia Tech, leman@vt.edu
[2]Department of Mathematical Sciences, Montana State University, andrew.hoegh@montana.edu

$$\mu(x_1, x_2) = \begin{cases} 8.0 & \text{if } x_1 \leq 5.0, \text{ and } x_2 \in \{A, B\} \\ 2.0 & \text{if } x_1 > 5.0, \text{ and } x_2 \in \{A, B\} \\ 1.0 & \text{if } x_1 \leq 3.0, \text{ and } x_2 \in \{C, D\} \\ 5.0 & \text{if } 3.0 < x_1 < 7.0, \text{ and } x_2 \in \{C, D\} \\ 8.0 & \text{if } x_1 \geq 7.0, \text{ and } x_2 \in \{C, D\}. \end{cases}$$

Identical to Chipman et al. (1998), we generate 800 points by taking $x_1 \sim Unif(0, 10)$, and $x_2 \sim Unif\{A, B, C, D\}$. We apply their original algorithm (equivalent to an MSS with $k = 1$) with their proposals (Grow/Prune/Change/Swap), and compare to the MSS with $k = 2$ ($k = 2$ often provides an ample trade off between exploration and exploitation). Figure 1 illustrates the integrated likelihoods found at each of 50,000 iterations.

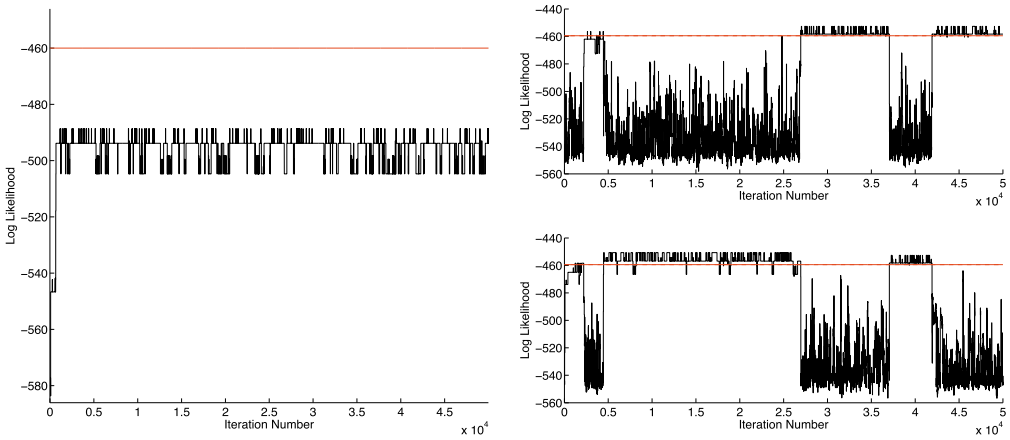


Figure 1: The dashed line represents the integrated likelihood of the true model. Panel a: the integrated likelihood at each iteration, using $k = 1$. Panel b: the integrated likelihood at each iteration, using $k = 2$.

From Figure 1 (Panel a) we see that the original algorithm is trapped within local regions of the tree space, and has converged to sets of trees very different from the true tree (as indicated by the difference in likelihood). In contrast, from Figure 1 (Panel b) we see that the two multiset elements take turns exploring the tree space, while the non-exploring point samples through a local high probability part of the space. Around iteration 5,000, the second element finds a high probability tree and sticks around this part of the space, while the first element continues to explore. Eventually (around iteration 25,000) the first element finds a tree that significantly improves upon the tree found by the second element. In fact the first element found the true tree! After this point, the first and second elements swap their roles as local and global *searchers*. That is while one element explores an interesting region of tree space, the other element searches for a new region of high probability. Such behavior prohibits local traps and enables

an efficient search procedure. We summarize by mentioning that Professor Pratola's newly devised tree proposals are a nice addition to the Bayesian CART literature. In combination with the MSS, we would expect to see even further improvements.

References

Chipman, H., George, E., and McCulloch, R. (1998). "Bayesian CART Model Search (with discussion)." *Journal of the American Statistical Association*, 94(443): 935–948. 932, 933

Chipman, H., George, E., and McCulloch, R. (2010). "BART: Bayesian Additive Regression Trees." *The Annals of Applied Statistics*, 4(1): 266–298. MR2758172. doi: http://dx.doi.org/10.1214/09-AOAS285. 932

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. 932

Kim, H. and MacEachern, S. (2015). "The Generalized Multiset Sampler." *Journal of Computational and Graphical Statistics*, 24(4): 1134–1154. MR3432933. doi: http://dx.doi.org/10.1080/10618600.2014.962701. 932

Leman, S., Chen, Y., and Lavine, M. (2009). "The Multiset Sampler." *Journal of the American Statistical Association*, 104(487): 1029–1041. MR2750235. doi: http://dx.doi.org/10.1198/jasa.2009.tm08047. 932

Comment by Reihaneh Entezari[1], Radu V. Craiu[2], and Jeffrey S. Rosenthal[3]

Abstract. The *Likelihood Inflating Sampling Algorithm (LISA)* (Entezari et al., 2016) is a new communication-free parallel method for posterior sampling of big datasets. In a divide and conquer strategy, LISA partitions the dataset into different "batches" and runs Markov Chain Monte Carlo (MCMC) methods on each batch of data *independently* using different processors. The results from all processors are then combined. In this discussion paper, we examine the performance of LISA when applied to the Bayesian Regression Trees model with tree proposals introduced by Pratola (2016). Our results show that LISA yields empirical distribution functions which are indistinguishable from those obtained using Pratola's algorithm, even though it first divides the data into K batches and can thus be used with datasets which are too large to fit into a single machine's memory.

Keywords: Bayesian Regression Trees (BART), big data, communication-free, Markov chain Monte Carlo (MCMC).

1 Introduction

We congratulate Matthew Pratola (henceforth, MP) for his innovative algorithm designed for Bayesian Regression Tree (BART) models. The latter are often used to analyze large datasets and this can pose seriously challenges as the run time for BART can be prohibitively slow. We discuss the use of MP's novel algorithm together with a parallel and communication-free method, the *likelihood inflating sampling algorithm (LISA)* that we have recently proposed (Entezari et al., 2016) to sample from posterior distributions arising from datasets which are too large to fit into a single machine's memory.

2 Divide and conquer analysis via BART and LISA

In order to apply LISA, the data is divided into K batches and for each batch j we compute the partial posterior $\pi_j(\theta|\vec{x}^{(j)}) \propto p(\theta)[L(\theta|\vec{x}^{(j)})]^K$ where $p(\theta)$ is the model's prior and $L(\theta|\vec{x}^{(j)})$ is the likelihood for the data in the jth batch. Samples obtained from each partial posterior are combined to perform inference about $\pi(\theta)$, the full data posterior.

Previously, Entezari et al. (2016) applied LISA to BART using the methods proposed in Chipman et al. (2010, 1998), and Kapelner and Bleich (2013), and concluded that a weighted average of batch-draws that were generated with a minor modification of LISA (modLISA), produces indistinguishable posterior distributions from the full posterior distribution of BART.

[1]Department of Statistical Sciences, University of Toronto, entezari@utstat.toronto.edu
[2]Department of Statistical Sciences, University of Toronto, craiu@utstat.toronto.edu, http://www.utstat.toronto.edu/craiu/
[3]Department of Statistical Sciences, University of Toronto, jeff@math.toronto.edu, http://probability.ca/jeff/

In this discussion paper, we will apply modLISA to BART using the tree proposals presented by Pratola (2016) to examine consistency in results and time savings.

We consider the Friedman's test function (Friedman, 1991):

$$f(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5,$$

and simulate 20,000 observations $y \sim N(f(x), \sigma^2)$ where $\sigma = 0.1$, and $x = (x_1, \ldots, x_{10})$ are uniformly drawn from $(0, 1)$. The sample size is chosen so that we can still run MP's algorithm to sample the full-data posterior in reasonable time. We have used the implementation of BART by Pratola (2016) to apply modLISA to this dataset with $K = 30$ batches.

Using the data presented in Table 1 one can compare the results of 1000 posterior samples generated from modLISA after 1000 burn-in iterations, to the SingleMachine which ran MP's algorithm on the full dataset. Note that we also simulated an additional 5000 observations as test data to fully compare the methods. Table 1 contains root mean squared error (RMSE) of $f(x)$ for both train and test data as well as the mean σ estimate. Both methods were performed with 30% rotate proposals without any adaptation. The results in Table 1 confirm that the parallel algorithm produces results that are very similar to the ones produced by SingleMachine. This is in line with the findings in Entezari et al. (2016). Table 2 shows, for each algorithm, the empirical test data coverage of the 90% credible interval for $f(x)$, average tree depth, total run time and the inverse product of Test RMSE and running time which can be thought of as a measure of computational efficiency. Interestingly, modLISA has higher coverage and lower average tree depth than SingleMachine. Total run time is more than 10 times faster for modLISA.

Method	Train RMSE	Test RMSE	Mean $\hat{\sigma}$
modLISA	0.137	0.147	0.176
SingleMachine	0.075	0.087	0.123

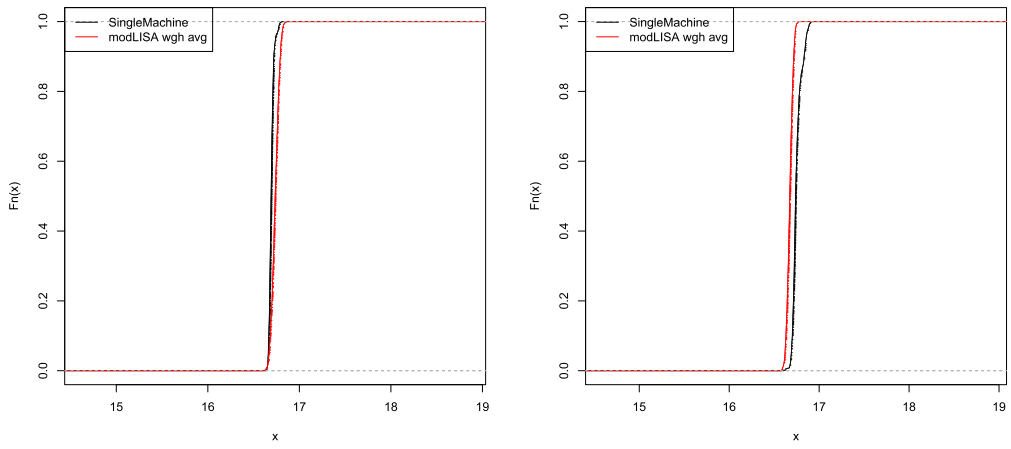
Table 1: Results of training data RMSE, test data RMSE and mean post burn-in $\hat{\sigma}$ from each method with 30% rotate proposals. There are $K = 30$ batches in total.

Method	Test Coverage	Avg tree depth	Total Run Time (secs)	1/(Test RMSE × Time)
modLISA	70.8 %	1.01	121.6	0.056
SingleMachine	63.7 %	2.07	1585.5	0.007

Table 2: Computational efficiency comparison between modLISA and SingleMachine.

Figure 1 compares the empirical distribution functions of $\hat{f}(x)$ in modLISA to SingleMachine for two different observations in the test data. As it is seen, the two empirical

distribution functions are indistinguishable. Overall, modLISA for BART with the new tree proposals introduced by MP, performs well in terms of accuracy and timing which shows consistent results with the ones found in Entezari et al. (2016). This illustrates the ability of modLISA to effectively sample from posterior distributions even when the datasets are too large and need to be divided into K batches before proceeding.



(a) observation 3300 (b) observation 4600

Figure 1: Comparing empirical distribution functions of $\hat{f}(x)$ in modLISA weighted average with $K = 30$ to SingleMachine BART for two different test observations.

References

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

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). "BART: Bayesian Additive Regression Trees." *The Annals of Applied Statistics*, 266–298. MR2758172. doi: http://dx.doi.org/10.1214/09-AOAS285. 935

Entezari, R., Craiu, R. V., and Rosenthal, J. S. (2016). "Likelihood Inflating Sampling Algorithm." arXiv:1605.02113. 935, 936, 937

Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines." *Annals of Statistics*, 1–67. MR1091842. doi: http://dx.doi.org/10.1214/aos/1176347963. 936

Kapelner, A. and Bleich, J. (2013). "bartMachine: Machine Learning with Bayesian Additive Regression Trees." arXiv:1312.2171. 935

Pratola, M. T. (2016). "Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian Regression Tree Models." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/16-BA999. 935, 936

Comment by Abdolreza Mohammadi[1] and Maurits Kaptein[2]

Abstract. The author should be commended for his outstanding contribution to the literature on Bayesian regression tree models. The author introduces three innovative sampling approaches which allow for efficient traversal of the model space. In this response, we add a fourth alternative.

Keywords: Markov chain Monte Carlo, birth–death process, continuous time Markov process, Bayesian regression tree.

1 Background

The algorithm introduced in Section 2.4 consists of a combination of birth/death and tree rotation proposals. Essentially, the *birth/death* mechanism explores trees with a different dimension, while the *rotation* mechanism explores alternative trees with the same dimensions. The specific birth/death mechanism proposed is known as reversible jump Markov chain Monte Carlo (RJ-MCMC) (Green, 1995) and is based on an ergodic discrete-time Markov chain. This algorithm is efficient only if the acceptance rate is high. As the author points out, this is not always the case.

This issue can be overcome by adopting birth–death MCMC (BD-MCMC) which is based on a *continuous*-time Markov process, as an alternative to RJ-MCMC. In this sampling scheme the algorithm explores the model space by jumping to a larger dimension (birth) or lower dimension (death) where each of these events is modeled as independent Poisson processes. The birth and death events thus occur in continuous time and their rates determine the stationary distribution of the process; see Figure 1. In BD-MCMC the moves between models are always accepted making the algorithm extremely efficient. Cappé et al. (2003) have shown, on appropriate rescaling of time, that the RJ-MCMC converges to a continuous time birth–death chain. One advantage of BD-MCMC is its ability to transit to low probability regions that can form a kind of "springboard" for the algorithm to flexibly move from one mode to another. The BD-MCMC algorithm has already been used effectively in the context of graphical models (Mohammadi and Wit, 2015; Mohammadi et al., 2017; Mohammadi and Wit, 2016) and mixture distributions (Stephens, 2000; Mohammadi et al., 2013).

2 Extension to BD-MCMC sampler

To implement the BD-MCMC mechanism we need to proof that the stationary distribution of the birth–death process converges to our target posterior distribution

$$Pr(T|data) \propto \pi(T)L(T).$$

[1]Dept. of Methodology and Statistics, Tilburg University, The Netherlands, a.mohammadi@uvt.nl
[2]Dept. of Methodology and Statistics, Tilburg University, The Netherlands, m.c.kaptein@uvt.nl

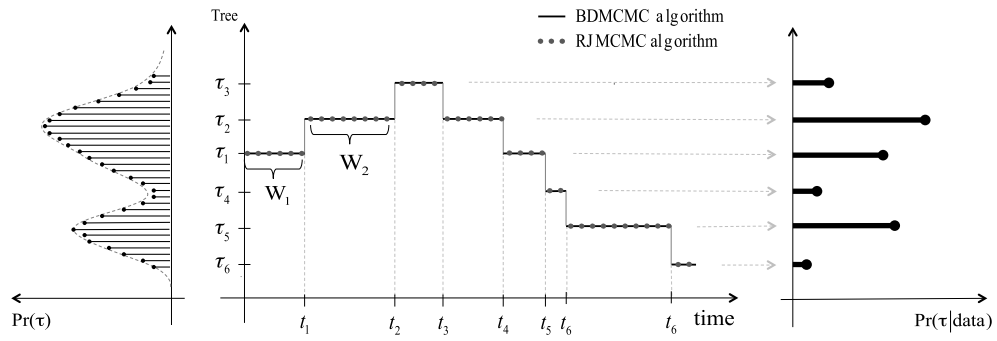


Figure 1: Visualization of the BD-MCMC algorithm. The left panel shows the true posterior distribution and the middle panel compares to progression of BD-MCMC and RJ-MCMC. $\{W_1, W_2, \ldots\}$ denote waiting times and $\{t_1, t_2, \ldots\}$ denote jumping times for the BD-MCMC algorithm, while the dots visualize the discrete RJ-MCMC samples. The right panel presents the estimated posterior probabilities of the trees which are the proportional to the waiting times (see, Cappé et al., 2003, Section 2.5).

Mohammadi and Wit (2015) show that this will be the case if the balance condition holds (Mohammadi and Wit, 2015, Appendix 1). It is easy to see that the proof of this condition for graphical models provided by Mohammadi and Wit (2015) is also valid for regression trees by noting that a tree is a special case of a graph. Let Θ_T be the tree-model space and $\Theta_{G_{max}}$ graph space in which G_{max} is a graph with the same number of nodes as a tree T with the maximum number of nodes. Clearly, $\Theta_T \subset \Theta_{G_{max}}$. This argument supports the implementation of the same birth/death mechanism as proposed by Mohammadi and Wit (2015) for trees: adding/removing a graph edge in the case of a birth/death event corresponds to adding/deleting a node of the tree (see Figure 2).

Following Mohammadi and Wit (2015), the birth and death rates are

$$B_n(T) = \frac{\pi(T^{+n})L(T^{+n})}{\pi(T)L(T)},$$
$$D_n(T) = \frac{\pi(T^{-n})L(T^{-n})}{\pi(T)L(T)},$$

in which T^{+n}/T^{-n} is a tree T with one more/less node n. As birth and death follow a Poisson processes, the time between two events has an exponential distribution and the probability of birth and death events are proportional to their rates.

We thus propose to replace the accept–reject mechanism of RJ-MCMC by a continuous time birth–death mechanism. In this new birth/death scheme the births and deaths occur at a higher rate when the components explain more of the data; a desirable feature not present in the RJ-MCMC approach. We believe that *combining* the BD-MCMC with the rotation mechanism will increase the efficiency of the traversal of model space (for a performance comparison of RJ-MCMC and BD-MCMC in graphical models see Section 4.1 of Mohammadi and Wit, 2015).

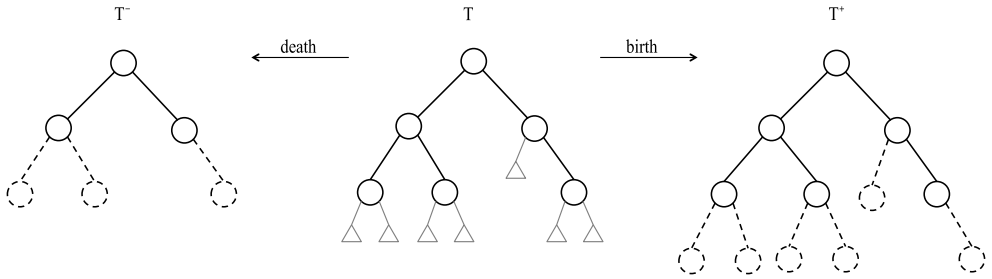


Figure 2: The birth–death mechanism for adding or deleting nodes of the tree.

References

Cappé, O., Robert, C., and Rydén, T. (2003). "Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3): 679–700. MR1998628. doi: http://dx.doi.org/10.1111/1467-9868.00409. 938, 939

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82(4): 711–732. MR1380810. doi: http://dx.doi.org/10.1093/biomet/82.4.711. 938

Mohammadi, A. and Wit, E. C. (2015). "Bayesian structure learning in sparse Gaussian graphical models." *Bayesian Analysis*, 10(1): 109–138. MR3420899. doi: http://dx.doi.org/10.1214/14-BA889. 938, 939

Mohammadi, A. and Wit, E. C. (2016). "BDgraph: An R package for Bayesian structure learning in graphical models." *Journal of Statistical Software*, in press. 938

Mohammadi, A., Salehi-Rad, M., and Wit, E. (2013). "Using mixture of Gamma distributions for Bayesian analysis in an M/G/1 queue with optional second service." *Computational Statistics*, 28(2): 683–700. MR3064474. doi: http://dx.doi.org/10.1007/s00180-012-0323-3. 938

Mohammadi, A., Abegaz Yazew, F., van den Heuvel, E., and Wit, E. C. (2017). "Bayesian modeling of Dupuytren disease using Gaussian copula graphical models." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, in press. 938

Stephens, M. (2000). "Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods." *Annals of statistics*, 40–74. MR1762903. doi: http://dx.doi.org/10.1214/aos/1016120364. 938

Comment by Luca Martino[1], Rafael B. Stern[2], and Francisco Louzada[3]

Keywords: Bayesian regression tree (BRT) models, Markov Chain Monte Carlo (MCMC), Multiple Try Metropolis algorithms, Sequential Monte Carlo methods.

We would like to congratulate the author for this contribution that provides improved proposals for the Metropolis–Hastings algorithm when the state of the chain can be mathematically represented as a tree. The paper is a very enjoyable read.

The author considers the use of the Metropolis–Hastings algorithm to simulate from the posterior in Bayesian regression tree (BRT) models. The algorithms typically used for such models suffer from poor mixing, remaining trapped in a local mode due to a small rate of acceptance of new candidate trees. The author suggests novel proposal mechanisms (to be used within a standard Metropolis–Hastings technique) in order to efficiently traverse the state space and improve the mixing of the chain. Our main consideration is that it would be worthwhile to combine the proposed method with other kinds of Monte Carlo (MC) algorithms, in order to improve the mixing of the resulting sampler.

For example, Multiple Try Metropolis (MTM) schemes (Casarin et al., 2011; Liu et al., 2000; Martino et al., 2012; Martino and Read, 2013; Martino and Louzada, 2016; Qin and Liu, 2001) can be easily applied in order to make inference in BRT models. In these methods, several trees are created and compared before they are proposed as a new state. Namely, the next tree of the chain is selected within a set of possible trees, drawn from a single or multiple proposal mechanisms (Casarin et al., 2011; Martino and Read, 2013). The main advantage of MTM is that it can quickly explore large portions of the sample space and reduce the probability of remaining trapped in a local mode. The use of different proposal functions can be also a way of combining several ideas suggested in different works. It is important to remark that, in general, the acceptance rate of an MTM converges to 1 as the number of candidates grows (Martino and Louzada, 2016; Martino and Read, 2013).

Another possible way of exploring the space of trees is through the use of Combinatorial Sequential Monte Carlo (C-SMC) (Wang, 2012). This generalization of sequential importance sampling allows one to generate tree proposals by iteratively adding nodes to smaller trees. Since this algorithm simultaneously grows several trees from the empty tree, it generally explores several of the local modes (see (Stern, 2015) and the related work (Naesseth et al., 2015) for other applications of SMC schemes for inference in tree models).

It is possible to combine all the previous approaches as proposals in a particle MH (PMH) chain (Andrieu et al., 2010). PMH is an MCMC technique that can be interpreted as an MTM scheme where the set of candidate trees is built by a particle filter

[1]Universitat de València, Spain, lukatotal@gmail.com
[2]Universidade Federal de Sã Carlos, Brazil
[3]Universidade de São Paulo (ICMC), Brazil

(a.k.a., Sequential Monte Carlo) approach (Martino et al., 2014). So the advantages of SMC and MCMC strategies are mixed in this solution (Martino et al., 2014). Finally, as a future work, the possible adaptation of the *perturbation* mentioned in the schemes of Section 4 could be investigated, designing schemes similar to the approaches proposed in (Haario et al., 2001, 2005).

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov Chain Monte Carlo Methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. MR2758115. doi: http://dx.doi.org/10.1111/j.1467-9868.2009.00736.x. 941

Casarin, R., Craiu, R. V., and Leisen, F. (2011). "Interacting Multiple Try Algorithms with Different Proposal Distributions." *Statistics and Computing* 23(2): 185–200. 941

Haario, H., Saksman, E., and Tamminen, J. (2001). "An Adaptive Metropolis Algorithm." *Bernoulli*, 7(2): 223–242. MR1828504. doi: http://dx.doi.org/10.2307/3318737. 942

Haario, H., Saksman, E., and Tamminen, J. (2005). "Componentwise Adaptation for High Dimensional MCMC." *Computational Statistics*, 20(2): 265–273. MR2323976. doi: http://dx.doi.org/10.1007/BF02789703. 942

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(449): 121–134. MR1803145. doi: http://dx.doi.org/10.2307/2669532. 941

Martino, L. and Read, J. (2013). "On the Flexibility of the Design of Multiple Try Metropolis Schemes." *Computational Statistics*, 28(6): 2797–2823. MR3141364. doi: http://dx.doi.org/10.1007/s00180-013-0429-2. 941

Martino, L. and Louzada, F. (2016). "Issues in the Multiple Try Metropolis Mixing." *Computational Statistics*, doi: http://dx.doi.org/10.1007/s00180-016-0643-9. 14 pages, in press. 941

Martino, L., Olmo, V. P. D., and Read, J. (2012). "A Multi-Point Metropolis Scheme with Generic Weight Functions." *Statistics & Probability Letters*, 82(7): 1445–1453. MR2929799. doi: http://dx.doi.org/10.1016/j.spl.2012.04.008. 941

Martino, L., Leisen, F., and Corander, J. (2014). "On Multiple Try Schemes and the Particle Metropolis–Hastings Algorithm." viXra:1409.0051. 942

Naesseth, C. A., Lindsten, F., and Schon, T. B. (2015). "Nested Sequential Monte Carlo Methods." *Proceedings of the International Conference on Machine Learning (JMLR)*. 941

Qin, Z. S. and Liu, J. S. (2001). "Multi-Point Metropolis Method with Application to Hybrid Monte Carlo." *Journal of Computational Physics*, 172: 827–840. MR1857620. doi: http://dx.doi.org/10.1006/jcph.2001.6860. 941

Stern, R. B. (2015). "A Statistical Contribution to Historical Linguistics." Ph.D. thesis, Carnegie Mellon University. 941

Wang, L. (2012). "Bayesian Phylogenetic Inference via Monte Carlo Methods." Ph.D. thesis, University of British Columbia. 941

Acknowledgments

This work has been partially supported by the European Research Council (ERC) under the ERC-CoG-2014 project 647423 (http://erc.europa.eu/), by the Grant 2014/23160-6 of São Paulo Research Foundation (FAPESP) and by the Grant 305361/2013-3 of National Council for Scientific and Technological Development (CNPq).