Bayesian Analysis (2016) **00**, Number 0, pp. 1

Comment on Article by Matthew T. Pratola

Reihaneh Entezari[*,†], Radu V. Craiu[*,‡] and Jeffrey S. Rosenthal[*,§]

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 from 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 by taking a weighted average of batch-draws that were generated with a minor modification to LISA (modLISA), one can produce indistinguishable posterior distributions from the full posterior distribution of BART.

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

[*]Department of Statistical Sciences, University of Toronto
[†]entezari@utstat.toronto.edu
[‡]craiu@utstat.toronto.edu, url: http://www.utstat.toronto.edu/craiu/
[§]jeff@math.toronto.edu, url: http://probability.ca/jeff/

 DOI: 0000

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.

Table 1 is comparing 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 on one single machine. 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. As seen in Table 1 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.

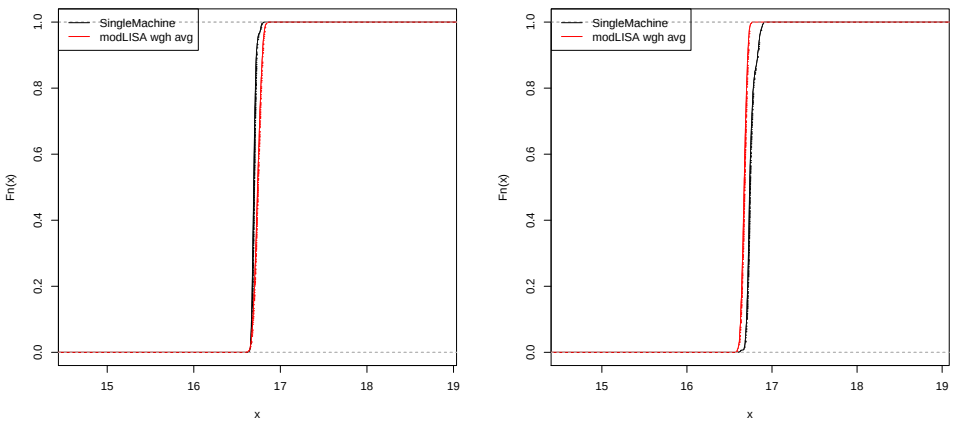
Table 1: Results of training data RMSE, test data RMSE and mean post burn-in $\hat{\sigma}$ from each method with 30% rotate proposals. There are $K = 30$ batches in total.

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

Table 2: Computational efficiency comparison between modLISA and SingleMachine

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

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. 1

— (2010). "BART: Bayesian additive regression trees." *The Annals of Applied Statistics*, 266–298. 1

Entezari, R., Craiu, R. V., and Rosenthal, J. S. (2016). "Likelihood Inflating Sampling Algorithm." *arXiv preprint arXiv:1605.02113*. 1, 2

Friedman, J. H. (1991). "Multivariate adaptive regression splines." *Annals of Statistics*, 1–67. 1

Kapelner, A. and Bleich, J. (2013). "bartMachine: Machine Learning with Bayesian Additive Regression Trees." *arXiv preprint arXiv:1312.2171*. 1

Pratola, M. T. (2016). "Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian Regression Tree Models." *Bayesian Analysis*. 1, 2