Stats 230 Final Project Report

David Mwakima and Jizhi Zhang

03/20/2023

1 Introduction

Markov Chain Monte Carlo (MCMC) is a numerical integration technique. The main motivation for this technique is that it is often not feasible both mathematically and practically to evaluate certain integrals in high dimensions. In Bayesian statistics, in particular, all inference proceeds via the posterior distribution $p(\theta|\mathbf{X})$ where θ could be a vector of multiple parameters. This distribution is given by

$$p(\theta|\mathbf{X}) = \frac{1}{C} f(\mathbf{X}|\theta)p(\theta)$$

where $f(\mathbf{X}|\theta)$ is our sampling model for $\mathbf{X}$, $p(\theta)$ is our prior model for θ and $C = \int_\Theta f(\mathbf{X}|\theta)p(\theta)$ is the normalizing constant. But computing C requires integration if we are going to evaluate this posterior and use it for inference. This is difficult, especially in cases where: (1) an expression for the integrand is not available in closed analytic form and (2) high dimensional models with 5 or more parameters.

Since Gelfand and Smith (1990) rediscovered the Metropolis-Rosenbluth-Hastings (MRH) algorithm developed by Metropolis et al. (1953), this difficulty in Bayesian analysis has been addressed. This algorithm relies on the theory of Markov stochastic processes. Assuming certain conditions (irreducibility, positive recurrence, detailed balance) are satisfied by a homogeneous Markov chain $\{X_n\}$ on state space E, then one can show that the chain possesses a stationary distribution π. One then appeals to the Ergodic Theorem (a dependent samples analogue of the Strong Law of Large Numbers for i.i.d samples) that guarantees that for any initial distribution:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} f(X_k) = \sum_{i\in E} f(i)\pi_i = E_\pi[f(X)]$$

Here's how the algorithm works for Bayesian inference. Let $p(\theta|\mathbf{X})$ be the target distribution, where θ is a vector of parameters. Then with proposal densities $q(\theta^{(j)}|\theta^{(i)}, \mathbf{X})$, the MRH acceptance ratio is:

$$a(\theta^{(j)}, \theta^{(i)}, \mathbf{X}) = \min\left\{ \frac{p(\theta^{(j)}|\mathbf{X})}{p(\theta^{(i)}|\mathbf{X})} \frac{q(\theta^{(i)}|\theta^{(j)}, \mathbf{X})}{q(\theta^{(j)}|\theta^{(i)}, \mathbf{X})}, 1 \right\}$$

And the MRH algorithm, in pseudo-code, for approximating $E_{p(\theta|\mathbf{X})}[h(\mathbf{X})]$ is:

	Metropolis-Rosenbluth-Hastings Algorithm
1:	Start with some initial value $\theta = \theta^{(0)}$
2:	**for** i = 0 to N **do**
3:	Simulate $\theta^{(i+1)}$ with $q(\theta^{(i+1)}|\theta^{(i)}, \mathbf{X})$
4:	Compute $a(\theta^{(i+1)}, \theta^{(i)}, \mathbf{X})$
5:	Generate $U \sim \text{Unif}(0, 1)$
6:	Accept $\theta^{(i+1)}$ if $U \leq a(\theta^{(i+1)}, \theta^{(i)}, \mathbf{X})$
	Otherwise $\theta^{(i+1)} = \theta^{(i)}$
7:	**end for**
8:	**return** $\frac{1}{N}\sum_{i=1}^{N} h(X_i)$

However, the use of MCMC for large datasets presents a new research frontier (Bardenet, Doucet, and Holmes (2014) and Bardenet, Doucet, and Holmes (2017)). This is because when $\mathbf{X}$ is large $n \gg 1$, as is typically the case in genomics, spatial statistics and cosmology; evaluating the likelihood ratio

$$\frac{p(\theta^{(j)}|\mathbf{X})}{p(\theta^{(i)}|\mathbf{X})} = \frac{p(\theta^{(j)})}{p(\theta^{(i)})} \frac{\prod_{k=1}^{n} f(X_k|\theta^{(j)})}{\prod_{k=1}^{n} f(X_k|\theta^{(i)})}$$

appearing in the MRH acceptance ratio is computationally intensive (Bardenet, Doucet, and Holmes (2014)).

As a consequence MCMC with the MRH algorithm cannot be considered for reasonable runtime when n is very large.

In our report we consider a paper by Maire, Friel, and Alquier (2019) that addresses the problem of using MCMC for large datasets. This paper proposes a new methodology, which the authors call *Informed Sub-Sampling MCMC* (ISS-MCMC), for doing Bayesian MCMC approximation of the posterior distribution. This is a scalable version of the Metropolis-Hastings algorithm designed for situations when n is so big that to approximate the posterior distribution takes a very long time. ISS-MCMC is "informed" because it makes use of a measure of similarity with respect to the full dataset through summary statistics. It is "sub-sampling" because it uses this measure to select a subset of the dataset that will be used by the Markov transition kernel at the k-th iteration of the algorithm. In this way, the Markov chain transition kernel uses only a fraction n/N of the entire dataset. They show using examples that choosing $n \ll N$ can lead to significant reductions in computational run-times while still retaining the simplicity of the standard Metropolis-Hastings algorithm. Moreover, unlike other approaches (discussed below), their method can be applied to virtually any model (involving i.i.d. data or not) and it does not require any assumption on the likelihood function nor on the prior distribution.

2 Comparison with other approaches

The authors note that sub-sampling of the full dataset strategy has been proposed elsewhere, in particular, by Korattikara, Chen, and Welling (2014), Bardenet, Doucet, and Holmes (2014) and Maclaurin and Adams (2015). The key difference between their approach and these is that the Markov chain transition kernel only uses a fraction n/N of the available data which is by construction held constant throughout the algorithm. It is the subset variable that is randomly refreshed at each iteration according to the similarity measure.

Other similar approaches to solve these big data problems are Quiroz et al. (2018) and the *Confidence Sampler* in Bardenet, Doucet, and Holmes (2017). Both of these approaches use sophisticated "control variates" to get positive unbiased estimators (based on a subset of data) for the likelihoods in the MRH acceptance ratio. The authors note that these control variates are still computationally intensive.

The authors also find some affinity to their work in *"noisy"* or *"inexact"* approaches to MCMC due to Korattikara, Chen, and Welling (2014) and Alquier et al. (2016), where an approximate MRH rule based on a sequential hypothesis test is used accept or reject samples with high confidence using only a fraction of the data required for the exact MH rule. The cost here is that the number of likelihood evaluations is adaptively set by the algorithm at each iteration. When the chain reaches equilibrium, the computational complexity is of order $O(n)$. This number can be brought down if an accurate proxy of the log-likelihood ratio, acting as control variates, is available, as demonstrated in Bardenet, Doucet, and Holmes (2017).

Another approach which the authors compare their approach with is an approach based on continuous time Markov processes (Langevin diffusion, Zig-Zag process) in Fearnhead et al. (2018) and Bierkens, Fearnhead, and Roberts (2019). Here, the authors note that the computational hurdle involves calculation of the gradient of the log-likelihood, which may not always be unbiased. Moreover, these approaches depart significantly from the simplicity of the original discrete MRH algorithm.In the following sections we consider in more detail the main ideas of how this algorithm works (Section 3) and implement one of their examples of estimating the parameters in a logistic regression model with $N = 10^6$ using their algorithm.

3 Main ideas of how it works

Let $(Y_1, \ldots, Y_N)$ be a set of observed data and define $Y_U = \{Y_k, k \in U\}$ where $U \subset \{1, \ldots, N\}$. Let $\mathcal{S} : Y \to S \subseteq \mathbb{R}^s$. So $\mathcal{S}$ takes the data and returns an s dimensional function of them, $S \subseteq$

$\mathbb{R}^s$. If the model admits a sufficient statistic, then $\mathcal{S}$ maps Y or a subset of it Y_U to it, otherwise $\mathcal{S}$ maps Y or a subset of it Y_U to summary statistics $\bar{S} = S(Y_U)/n$. See Maire, Friel, and Alquier (2019) p. 452, 453, 456.

For all $n \leq N$, they define U_n as the set of possible combinations of n different integer numbers less than or equal to N and $\mathcal{U}_n$ as the powerset of U_n. For any subset $U \in \mathsf{U}_n$ define the vector of difference of sufficient statistics between the whole dataset and the subset Y_U to be:

$$\Delta_n(U) = \sum_{k=1}^{N} \mathcal{S}(Y_k) - N/n \sum_{k \in U} \mathcal{S}(Y_k) \quad (1)$$

If there is no sufficient statistic define an analogous difference measure using summary statistics as:

$$\bar{\Delta}_n(U) = S(Y)/N - S(Y_U)/n \quad (2)$$

3.1 Similarity through summary statistics

They then consider the distribution $\nu_{n,\epsilon}$ on the discrete space U_n defined for all $\epsilon \geq 0$ by:

$$\nu_{n,\epsilon}(U) \propto \exp(-\epsilon ||\Delta_n(U)||^2) \quad (3)$$

The distribution $\nu_{n,\epsilon}$ assigns a weight to any subset according to its representativeness with respect to the full dataset. It is a kind of tuning parameter. When $\epsilon = 0$, $\nu_{n,\epsilon}$ is uniform on U_n, while when $\epsilon \to \infty$, $\nu_{n,\epsilon}$ is uniform on the set of subsets that minimize $||\Delta_n(U)||$. The authors note that moving to general models (i.e. non i.i.d and non-exponential) amounts to relaxing the sufficient statistics existence assumption as well as the $\epsilon \to \infty$ condition. This is achieved by constructing a class of summary statistics for the model at hand and for which the following result holds. (We do not go into the details here - See Maire, Friel, and Alquier (2019) p. 454 Proposition 3)

For any $\theta \in \Theta$ and ϵ 0, there exists $M < \infty$ such that

$$\mathbb{E}_{n,\epsilon}\left\{ \frac{f(Y|\theta)}{f(Y_U|\theta)^{N/n}} \right\} < M$$

3.2 Setting up the algorithm

In the Informed Sub-Sampling method, the set of good subsamples is treated as a series of missing data (denoted by U_1, U_2, and so on) and is simulated using the ISS-MCMC algorithm. This algorithm generates a Markov chain on the extended space $\theta \times \mathsf{U}_n$. The sequence of sub samples is randomly updated in a way that favors those subsets with summary statistics that are similar to the full dataset, as suggested by the analysis in previous sections. The process is implemented using a symmetric transition kernel R on $(\mathsf{U}_n, \mathcal{U}_n)$. A transition from (θ_i, U_i) to (θ_{i+1}, U_{i+1}) involves two steps:

(i)

(a) Propose a new subset variable $U \sim R(U_i, \cdot)$

(b) Set $U_{i+1} = U$ with probability

$$\beta(U_i, U) = \min\{b(U_i, U), 1\} \quad (4)$$

, where

$$b(U_i, U) = \exp\left(\epsilon(|\Delta_n(U_i)|^2 - |\Delta_n(U)|^2)\right) \quad (5)$$

, and $U_{i+1} = U_i$ with probability $1-\beta(U_i, U)$. Here, Δ_n is defined in equation 2. The transition kernel R is selected based on the data and distribution.

(ii)

(a)propose a new parameter $\theta \sim Q(\theta_i, \cdot)$

(b)Set $\theta_{i+1} = \theta$ with probability

$$\alpha(\theta_i, \theta) = \min\{a(\theta_i, \theta), 1\} \quad (6)$$

, where

$$a(\theta_i, \theta) = \frac{\pi_n(\theta \mid Y_{U_{i+1}}) Q(\theta, \theta_i)}{\pi_n(\theta_i \mid Y_{U_{i+1}}) Q(\theta_i, \theta)} \quad (7)$$

, and $\theta_{i+1} = \theta_i$ with probability $1 - \alpha(U_i, U)$. The transition kernel Q is selected based on the data and distribution.

3.3 The Algorithm

Here is the pseudo-code for the Informed Sub-sampling Algorithm:

	Informed Sub-Sampling MCMC Algorithm
1:	**Input**: **initial state** $(\tilde{\theta}_0, U_0)$ **and summary statistics** $S_0 = \bar{S}(Y_{U_0})$ $S^* = \bar{S}(Y)$
2:	**for** $i = 1, 2, \ldots$ **do**
3:	propose a new subset $U \sim R(U_{i=1}, \cdot)$ and draw $J \sim \text{Unif}(0, 1)$
4:	compute $S = \bar{S}(Y_U)$ and $b = b(U_{i-1}, U)$ defined in (4)
5:	**if** $J \leq b$ **then**
6:	set $U_i = U$ and $S_i = S$
7:	**else**
8:	set $U_i = U_{i-1}$ and $S_i = S_{i-1}$
9:	**end if**
10:	propose a new parameter $\tilde{\theta} \sim Q(\tilde{\theta}_{i-1}; \cdot)$ and draw $I \sim \text{Unif}(0, 1)$
11:	compute $\tilde{\pi}(\tilde{\theta}_{i-1}\|Y_{U_i})$, $\tilde{\pi}_n(\tilde{\theta}\|Y_{U_i})$ and $\tilde{a}(\tilde{\theta}_{i-1}\|U_i)$ defined in (7)
12:	**if** $I \leq \tilde{a}$ **then**
13:	set $\tilde{\theta}_i = \tilde{\theta}$
14:	**else**
15:	set $\tilde{\theta}_i = \tilde{\theta_{i-1}}$
16:	**end if**
17:	**end for**
18:	**return: Markov Chain** $\{(\tilde{\theta}_i, U_i), i \in \mathbb{N}\}$

4 Example with logistic regression

Here we implement one example from their paper. Their example involves simulated logistic regression data with $N = 10^6$ observations. However given our limited computational memory, we could not implement the full $N = 10^6$ simulated example. Instead, we ran our simulation for $N = 10^5$ using and $n = 100, 500$ and 1000 and compared our results to the MRH without sub sampling.

Consider the logistic regression model:

$$P(Y = 1) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)} \quad (8)$$

We simulate N times with $\beta_0 = -1$, $\beta_1 = 1$ and $\beta_2 = 0$

We define the prior as a multivariate normal distribution with mean $\mu = (0, 0, 0)^T$ and variance $\Sigma = 100I$, where I is the identity matrix.

For the summary statistic S is the sum of Y_U since our data is binomial. The transition kernel of subsets $R(U_i, U)$ is defined as a random sample from the set U_n. The transition kernel of the parameter $Q(\beta_i, \beta)$ is defined as a uniform distribution in $\mathbb{R}^3$ with mean β_i and width $\delta = 0.2$. We chose this proposal density because it simplifies the calculation of the acceptance ratio in the ISS-MCMC algorithm.

Finally, in implementing the logistic regression example, we used two different tuning parameter values for ϵ; $\epsilon = 0.00005$ and $\epsilon = 1$. The authors of the paper are not clear on how they chose their ϵ values. See section 6.3 of Maire, Friel, and Alquier (2019). In some parts of the paper it seems like ϵ is 5×10^4, while in other parts it appears that ϵ is 5×10^{-6}. So we made this choice because for large values of $\epsilon > 1$ we did not see any change in the acceptance rate of a new proposal U_{i+1}.

4.1 Compare computational time of ISS-MCMC with M-H.

Finally, we present tables that compare the computational time of using ISS-MCMC. See Table 1.

5 Conclusion and Future Work

Since we could not handle simulated data of the order 10^6, future work can include using a computer with more memory to handle such a large

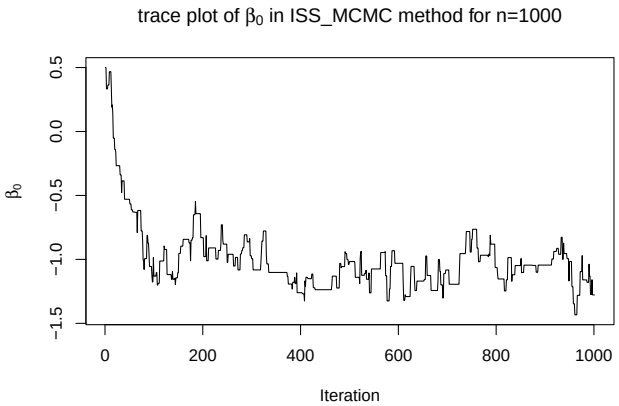
Table 1: Comparing Computational Time of MRH to ISS MCMC

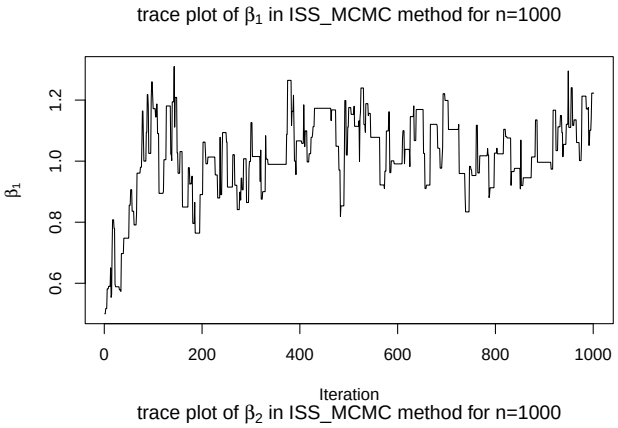
	System Time (s)
MRH	16.134
ISS MCMC (n = 100)	14.917
ISS MCMC (n = 500)	11.317
ISS MCMC (n = 1000)	13.702

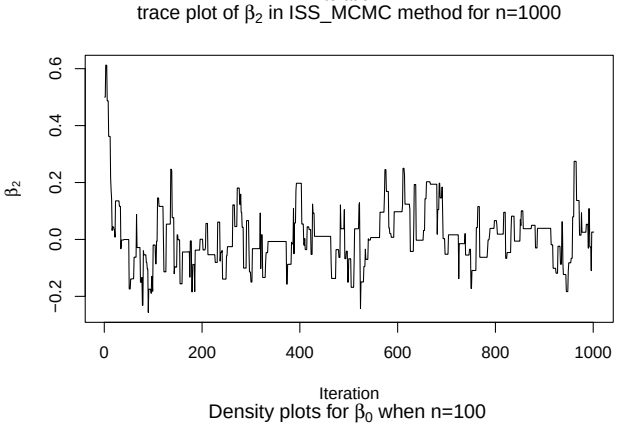
dataset in order to fully reproduce their example and check for any significant gains in reducing computational time. Also future work can include using different tuning parameters ϵ to investigate the quality the estimates for different choices of ϵ and to clarify what the optimal choices for ϵ actually are since the authors of the paper are not clear about this. Finally, because in theory we should expect a significant decrease in computational time when using ISS MCMC compared to using MRH, future work could also investigate why for the case of $n = 100$, we do not observe such a reduction.

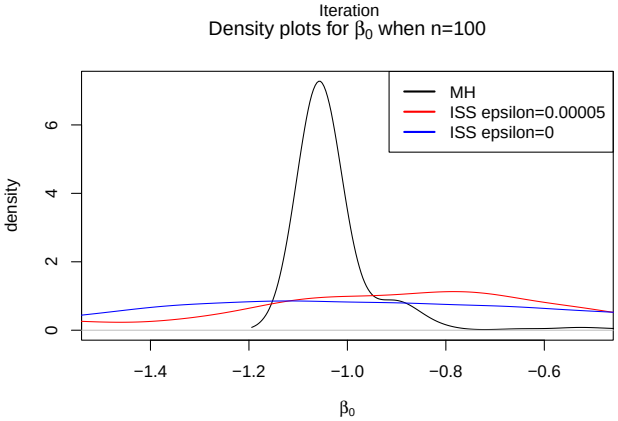
6 Appendix

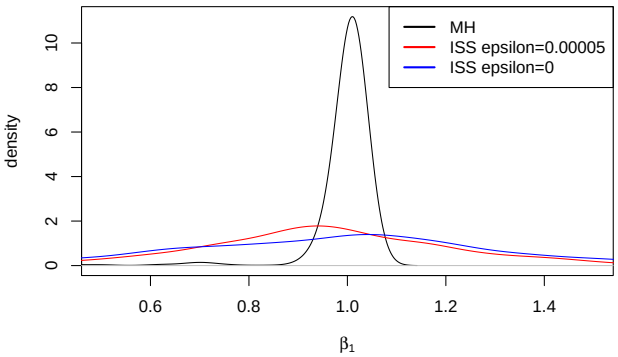
Here are marginal distributions and traceplots obtained using the ISS-MCMC algorithm that we used to check for convergence of our MCMC.

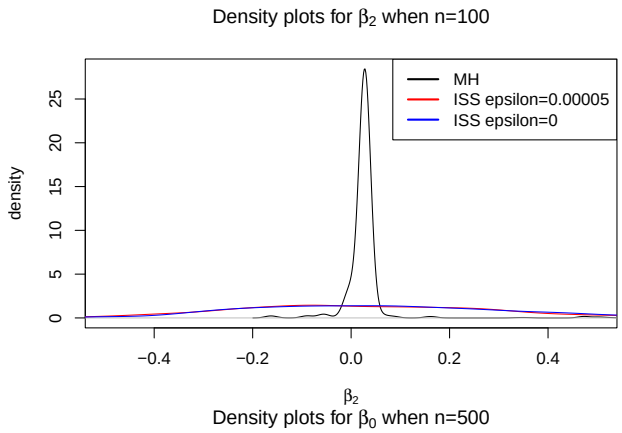


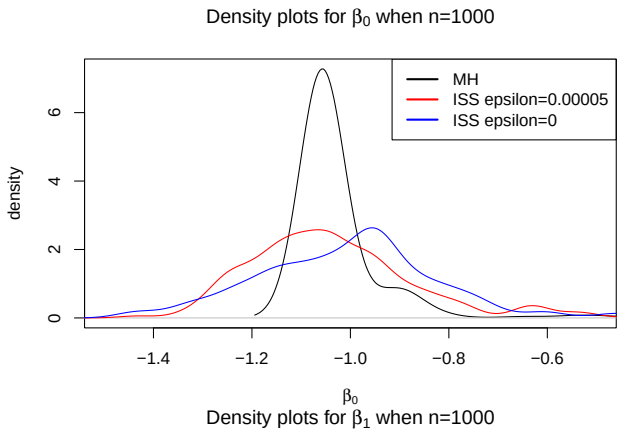


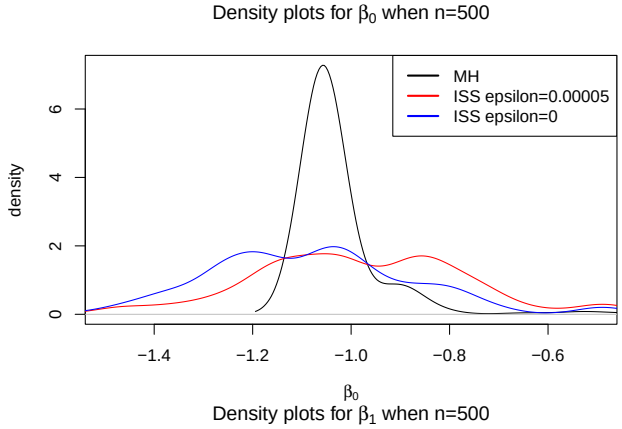


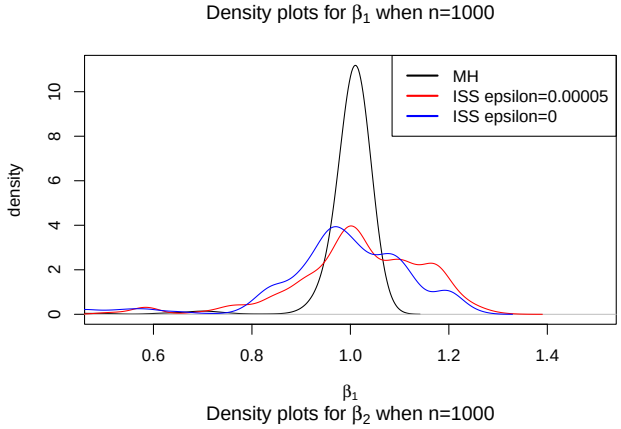


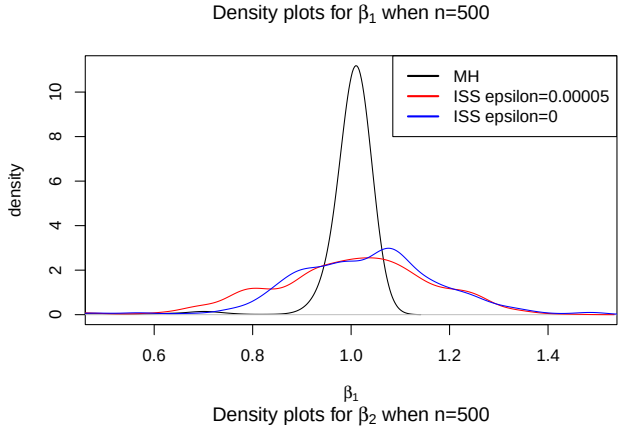


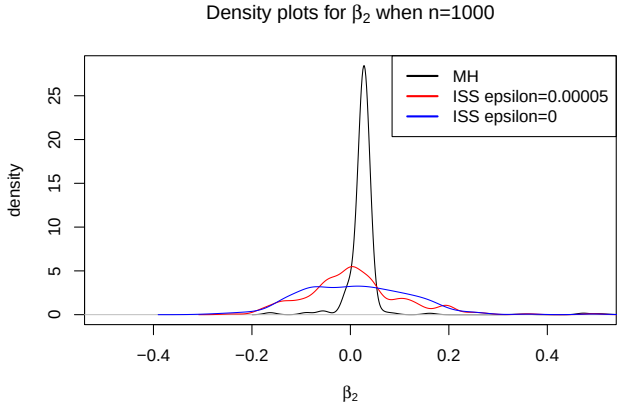

Density plots for β_2 when n=100
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_2

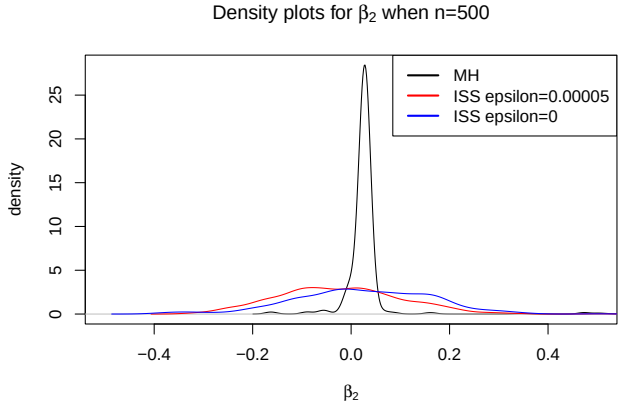

Density plots for β_0 when n=1000
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_0


Density plots for β_0 when n=500
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_0


Density plots for β_1 when n=1000
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_1


Density plots for β_1 when n=500
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_1


Density plots for β_2 when n=1000
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_2


Density plots for β_2 when n=500
MH
ISS epsilon=0.00005
ISS epsilon=0
density
β_2

References

Alquier, Pierre, Nial Friel, Richard Everitt, and Aidan Boland. 2016. "Noisy Monte Carlo: Convergence of Markov Chains with Approximate Transition Kernels." *Statistics and Computing* 26 (1-2): 29–47.

Bardenet, Rémi, Arnaud Doucet, and Chris Holmes. 2014. "Towards Scaling up Markov Chain Monte Carlo: An Adaptive Subsampling Approach." In *International Conference on Machine Learning*, 405–13.

Bardenet, Rémi, Arnaud Doucet, and Christopher C Holmes. 2017. "On Markov Chain Monte Carlo Methods for Tall Data." *Journal of Machine Learning Research* 18 (47).

Bierkens, Joris, Paul Fearnhead, and Gareth Roberts. 2019. "The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data." *The Annals of Statistics* 47 (3): 1288–1320.

Fearnhead, Paul, Joris Bierkens, Murray Pollock, and Gareth O Roberts. 2018. "Piecewise Deterministic Markov Processes for Continuous-Time Monte Carlo." *Statistical Science* 33 (3): 386–412.

Gelfand, Alan E, and Adrian FM Smith. 1990. "Sampling-Based Approaches to Calculating Marginal Densities." *Journal of the American Statistical Association* 85 (410): 398–409.

Korattikara, Anoop, Yutian Chen, and Max Welling. 2014. "Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget." In *International Conference on Machine Learning*, 181–89.

Maclaurin, Dougal, and Ryan P. Adams. 2015. "Firefly Monte Carlo: Exact MCMC with Subsets of Data." In *Proceedings of the 24th International Conference on Artificial Intelligence*, 4289–95. AAAI Press.

Maire, Florian, Nial Friel, and Pierre Alquier. 2019. "Informed Sub-Sampling MCMC: Approximate Bayesian Inference for Large Datasets." *Statistics and Computing* 29: 449–82.

Metropolis, Nicholas, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics* 21 (6): 1087–92.

Quiroz, Matias, Robert Kohn, Mattias Villani, and Minh-Ngoc Tran. 2018. "Speeding up MCMC by Efficient Data Subsampling." *Journal of the American Statistical Association*.