Efficient MCMC Sampling Finite-State Markov Jump Processes and Bayesian Inference

Boqian Zhang
Department of Statistics, Purdue University
Vinayak Rao
Department of Statistics, Purdue University

November 11, 2016

Abstract

Abstrct.

Keywords: Markov jump process, MCMC, Metropolis Hasting sampler, Bayesian inference

1 Introduction

In this paper, we tackled the problem of sampling MJP parameters from the posteriors, efficiently, using Metropolis Hasting algorithm.

2 Metropolis Hasting for Bayesian Inference using FFBS within the Gibbs Sampling On MJPs

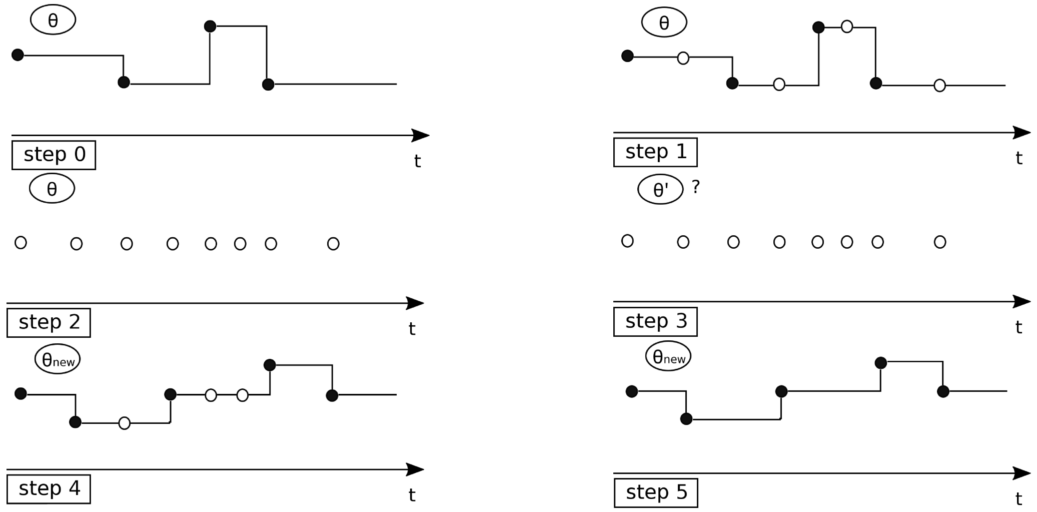

Figure 1: MH algorithm

Algorithm 1 MH In Gibbs sampling for MJPs

```
Input: A set of partial and noisy observations $y_{[t_0,t_{N+1})}$, Initial distribution over states $\pi_0$, Metropolis Hasting proposal $q(.|\theta)$.
The previous MJP path $S(t) = (S, T)$, the previous MJP parameters $\theta$.
Output: A new MJP trajectorie $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, A series of MJP parameters $\tilde{\theta}$.
0: Let $\Omega = h(\theta)$, with $\Omega > max_s|A_s|$ using some deterministic function $h$.
1: Sample virtual jumps $U \subset [t_{start}, t_{end}]$ from a Non homogeneous Poisson process with piecewise-constant rate
       $R(t) = (\Omega + A_{S(t)}).$
   Define $W = T \cup U$.
2: Propose $\theta^* \sim q(.|\theta)$.
   Accept $\theta^*$ as $\tilde{\theta}$ with probability $\alpha$.
       $\alpha = 1 \wedge \frac{P(W,\theta^*|y)}{P(W,\theta|y)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}$
       $= 1 \wedge \frac{P(y|W,\theta^*)P(W|\theta^*)p(\theta^*)}{P(y|W,\theta)P(W|\theta)p(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}.$
3: Sample a path $\tilde{V}$, from a discret-time Markov chain with $|W|+1$ steps, using FFBS algorithm. The transition matrix of the Markov chain is $B = (I + \frac{A}{\Omega})$ while the initial distribution over states is $\pi_0$. The likelihood of state $s$ at step $i$ is
       $L_i(s) = P(Y_{[w_i,w_{i+1})}|S(t) = s\ for\ t \in [w_i, w_{i+1})) = \prod_{j:t_j \in [w_i,w_{i+1})} p(y_{t_j}|S(t_j) = s).$
4: Let $\tilde{T}$ be the set of times in $W$ when the Markov chain changes state. Define $\tilde{S}$ as the corresponding set of state values. Return $(\tilde{S}, \tilde{T}, \tilde{\theta})$.
```

3 Improved Metropolis Hasting for Bayesian Inference using FFBS within the Gibbs Sampling On MJPs

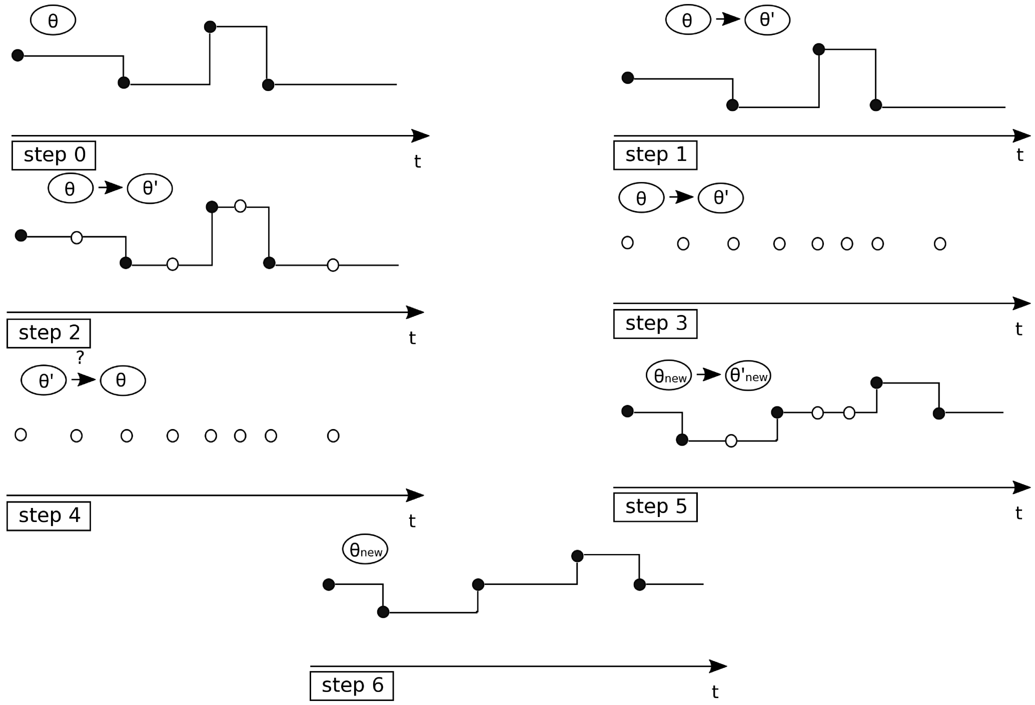

Figure 2: Improved MH algorithm

Algorithm 2 MH In Gibbs sampling for MJPs

Input: A set of partial and noisy observations $y_{[t_0,t_{N+1})}$, Initial distribution over states π_0, Metropolis Hasting proposal $q(.|\theta)$.

The previous MJP path $S(t) = (S, T)$, the previous MJP parameters (θ).

Output: A new MJP trajectorie $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, A series of MJP parameters $\tilde{\theta}$.

0: Sample $\theta^* \sim q(.|\theta)$. And let $\Omega = h(\theta)+h(\theta^*)$, with $h(\theta) > max_s|A_s(\theta)|$, $h(\theta^*) > max_s|A_s(\theta^*)|$ using some deterministic function h.

1: Sample virtual jumps $U \subset [t_{start}, t_{end}]$ from a Non homogeneous Poisson process with piecewise-constant rate

$$R(t) = (\Omega + A_{S(t)}(\theta)).$$

Define $W = T \cup U$.

2: Propose (θ^*, θ) and accept θ^* as $\tilde{\theta}$ with probability α.

$$\begin{aligned}\alpha &= 1 \wedge \frac{P(W, (\theta^*, \theta)|y)}{P(W, (\theta, \theta^*)|y)} \\ &= 1 \wedge \frac{P(y|W, \theta^*, \theta)P(W|(\theta^*, \theta))p((\theta^*, \theta))}{P(y|W, (\theta, \theta^*))P(W|(\theta, \theta^*))p((\theta, \theta^*))} \\ &= 1 \wedge \frac{P(y|W, \theta^*, \theta)p((\theta^*, \theta))}{P(y|W, (\theta, \theta^*))p((\theta, \theta^*))}.\end{aligned}$$

3: Sample a path $\tilde{V}$, from a discret-time Markov chain with $|W|+1$ steps, using FFBS algorithm. The transition matrix of the Markov chain is $B = (I + \frac{A(\tilde{\theta})}{\Omega})$ while the initial distribution over states is π_0. The likelihood of state s at step i is

$$L_i(s) = P(Y_{[w_i,w_{i+1})}|S(t) = s \; for \; t \in [w_i, w_{i+1})) = \prod_{j:t_j \in [w_i,w_{i+1})} p(y_{t_j}|S(t_j) = s).$$

4: Let $\tilde{T}$ be the set of times in W when the Markov chain changes state. Define $\tilde{S}$ as the corresponding set of state values. Return $(\tilde{S}, \tilde{T}, \tilde{\theta})$.

4 Verifications of Algorithm 1

Proof of Algorithm 1:

Assume: $S = [S_0, S_1, ..., S_N]$,$T = [T_0, T_1, ..., T_N, T_{N+1}(T_{end})]$, and y as observations.

In JMLR-2013 Fast MCMC Sampling for MJP and Extensions, the FFBS frame contains α_t as follows.

Since after uniformization, the virtual jumps are added. Then the state process of the trajectory with virtual jumps is just a discrete time markov jump process. The key point is that we need to have W be conditioned, to get the marginal probability $P(y_{[T_0,T_{N+1})}|\theta,W)$ from FFBS algorithm.

$$\alpha_t^\theta(s) = P(S_t = s\ , y_{[T_0,T_t)}, U, T).$$

$$P(y_{[T_0,T_{N+1})}|\theta,W) = \sum_{s=0}^{N-1} \alpha_N^\theta(s) \cdot P(y_{[T_N,T_{N+1})}|S_N = s).$$

$$P(\theta,W|y) \propto P(\theta,W,y) = P(y|W,\theta)P(W|\theta)P(\theta).$$

$P(y|W,\theta)$ is the marginal probability we get after Forward Filtering Algorithm and the $P(W|\theta)$ is the probability density for the $poisson(\Omega)$, because of the uniformization procedure. Let denote the kernel for (a), (b) and (c) as $\kappa_1(\theta^*|\theta,W,T,S,y)$, $\kappa_2(S^*,T^*|S,T,W,\theta^*,y)$ and $\kappa_3(W^*|S^*,T^*,\theta^*,y)$.

For Step (a) $\kappa_1(\theta^*|\theta,W,T,S)$:

$$\begin{aligned} P((W,T,S,\theta) \to (W,T,S,\theta^*))P(\theta,W|y) &= P(\theta^*,W|y)q(\theta|\theta^*) \wedge P(\theta,W|y)q(\theta^*|\theta) \\ &= P((W,T,S,\theta^*) \to (W,T,S,\theta))P(\theta^*,W|y). \end{aligned}$$

Thus, $\int ab\kappa_1(\theta^*|\theta)P(\theta,W|y)d\theta = P(\theta^*,W|y)$.

So the stationary distribution of κ_1 is $P(\theta,W|y)$.

Step (b) $\kappa_2(S^*,T^*|S,T,W,\theta^*,y)$:

Step(b) is the same as Fast MJPs Gibbs sampling scheme.

$$((S,T,\theta,W) \to (S^*,T^*,\theta,W)|y) = P(V^*|W,\theta,y) = P(V^*|W,\theta,y)/P(W,\theta,y)$$

$$\begin{aligned} P((S,T) \to (S^*,T^*)|W,\theta,y)P(S,T|W,\theta,y) &= P(V^*|W,\theta,y)P(V|W,\theta,y) \\ &= P((S^*,T^*) \to (S,T)|W,\theta,y)P(S^*,T^*|W,\theta,y) \end{aligned}$$

So the stationary distribution of $\kappa_2(S^*, T^*|S, T, W, y)$ is $P(S, T|W, \theta, y)$. Now, let's consider $\kappa_2 \circ \kappa_1(S^*, T^*, \theta^*|S, T, \theta, y, W)$.

$$((S, T, \theta, W) \rightarrow (S^*, T^*, \theta^*, W)|y) = P((W, T, S, \theta) \rightarrow (W, T, S, \theta^*))P((S, T, \theta^*.W) \rightarrow (S^*, T^*, \theta^*, W)|y).$$

The stationary distribution of $\kappa_1(S^*, T^*, U^*|S, T, U)$ is $P(S, T, U|\theta, y)$. And the stationary distribution of $\kappa_2(U^*|U)$ is $P(U|S, T, \theta, y)$.

$$\begin{aligned}
&P((S, T, \theta, W) \rightarrow (S^*, T^*, \theta^*, W)|y)P(S, T, \theta|W, y)\\
&= P((W, T, S, \theta) \rightarrow (W, T, S, \theta^*)) \cdot P(\theta|W, y) \cdot P((S, T, \theta^*.W) \rightarrow (S^*, T^*, \theta^*, W)|y)P(S, T|\theta, W, y)\\
&= P((W, T, S, \theta^*) \rightarrow (W, T, S, \theta)) \cdot P(\theta^*|W, y) \cdot P((S^*, T^*, \theta^*.W) \rightarrow (S, T, \theta^*, W)|y)P(S^*, T^*|\theta, W, y)\\
&= P((S^*, T^*, \theta^*, W) \rightarrow (S, T, \theta, W)|y)P(S, T, \theta|W, y).
\end{aligned}$$

So the stationary distribution of $\kappa_2 \circ \kappa_1$ is $P(S, T, \theta|W, y)$.
Obviously, $\kappa_3(W^*|W, S^*, T^*, \theta^*, y)$ has $P(W|S^*, T^*, \theta^*, y)$ as stationary distribution.
Therefore, $\int \kappa_3(W^*|W, S^*, T^*, \theta^*, y)P(W, S^*, T^*, \theta^*|y)dW = P(W^*, S^*, T^*, \theta^*|y)$.
Thus, $\int \kappa_3 \cdot (\int \kappa_2 \circ \kappa_1 \cdot P(W, S, T, \theta|y)d\theta dS dT)dW = \int \kappa_3 P(W, S^*, T^*, \theta^*|y)dW = P(W^*, S^*, T^*, \theta^*|y)$.
So the stationary distribution of $\kappa_3 \circ \kappa_2 \circ \kappa_1$ is $P(W^*, S^*, T^*, \theta^*|y)$.

5 Verifications of Algorithm 2

Proof: Our state is $(W, S, T, \theta, \theta^*)$.

$$\begin{aligned}
p(y, W, S, T, \theta, \theta^*) &= p(\theta)q(\theta^*|\theta)P(S, T|\theta, \theta^*)P(W|S, T, \theta, \theta^*)P(y|S, T, \theta, \theta^*)\\
&= p(\theta)q(\theta^*|\theta)P(S, T|\theta)P(W|S, T, \theta, \theta^*)P(y|S, T).
\end{aligned}$$

The marginal distribution of $(y, S, T, \theta, \theta^*)$ and (y, S, T, θ) as follows.

$$\begin{aligned}
p(y, S, T, \theta, \theta^*) &= p(\theta)q(\theta^*|\theta)P(S, T|\theta, \theta^*)P(y|S, T, \theta, \theta^*)\\
&= P(y, S, T, \theta)q(\theta^*|\theta).
\end{aligned}$$

$$p(y, S, T, \theta) = p(\theta)P(S, T|\theta)P(y|S, T, \theta).$$

So the conditional distribution over θ^* given (y, S, T, θ) is $q(\theta^*|\theta)$. And the conditional distribution over W given $(y, S, T, \theta, \theta^*)$ is $P(W|S, T, \theta, \theta^*)$, which is actually the distribution of Non Homogeneous Poisson Process with piecewise constant rate $h(\theta) + h(\theta^*) - A_{S(t)}(\theta)$.

Thus the Step 1 + Step 2 is actually equivalent to sampling from the conditional distribution $P(\theta^*, W|S, T, \theta, y)$.

The Step 3 + Step 4 satisfy the detailed balance condition. The reason is as follows.

$$\begin{aligned}
&P((W, S, T, (\theta, \theta^*)) \to (W, S^*, T^*, (\theta^*, \theta)))P(S, T, (\theta, \theta^*)|W, y) \\
&= (1 \wedge \frac{P((\theta^*, \theta)|W, y)}{P((\theta, \theta^*)|W, y)})P(S^*, T^*|W, (\theta^*, \theta), y)P(S, T|W, (\theta, \theta^*), y)P((\theta, \theta^*)|W, y) \\
&= P((W, S^*, T^*, (\theta^*, \theta)) \to (W, S, T, (\theta, \theta^*)))P(S^*, T^*, (\theta^*, \theta)|W, y)
\end{aligned}$$

Therefore the stationary distribution of this MCMC sampler is $P(W, S, T, (\theta, \theta^*)|y)$. Thus the stationary distribution of (S, T, θ) is the corresponding marginal distribution $P(S, T, \theta|y)$.

6 Generic Exponential Model

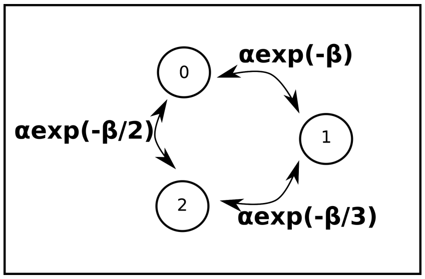

Figure 3: exp model

Assume: $S = [S_0, S_1, ..., S_N]$,$T = [t_0(t_{start}), t_1, ..., t_N, t_{N+1}(t_{end})]$, and y as observations.

We consider a specific structure of rate matrix A. $A_{ij} = \alpha f_{ij}(\beta)$, $i \neq j$. $A_{ii} = -\sum_{j \neq i} A_{ij}$. $0 \leqslant f_{ij} \leqslant 1$. Denote $F_i(\beta) = \sum_{j \neq i} f_{ij}(\beta)$.

Algorithm 3 Generic Gibbs sampling for MJPs for Gamma priors

```
**Input:** observations $y_{[t_0,t_{k+1})}$
Initialize, $i = 0$
(a) Set $\alpha(0), \beta(0)$ arbitrarily and set current trajectory $[S,T](0)$ arbitrarily.
(b) Uniformize $[S,T](0)$, to get virtual jumps $U$.
**repeat**
  **for** $i = 1$ **to** $N$ **do**
    (a) Sample $U(i) \sim P(U|\beta(i-1), \alpha(i-1), S(i-1), T(i-1), y)$.
    (b) Use FFBS algorithm to sample states given all the jump times(both true jumps and virtual jumps). (i.e. $V(i) \sim P(V|\beta(i-1), \alpha(i-1), W(i), y)$.) Then delete all the virtual jumps to get $S(i), T(i)$.
    (c) Propose $\beta^* \sim q(.|\beta(i-1))$.
    Set $\beta(i) = \beta^*$, with probability $P_{acc} = 1 \wedge \frac{P(\beta^*|S(i),T(i))}{P(\beta(i-1)|S(i),T(i))} \frac{q(\beta(i-1)|\beta^*)}{q(\beta^*|\beta(i-1))}$;
    Otherwise set $\beta(i) = \beta(i-1)$.
    (d) Sample $\alpha(i) \sim P(.|\beta(i), S(i), T(i), y)$.
    It is a $Gamma(\mu + N, \lambda + \sum_0^N F_{S_i}(\beta)(t_{i+1} - t_i))$ distribution actually.
  **end for**
**until** $i = N$
```

$$
\begin{aligned}
P(s_0, S, T|\alpha, \beta) &= \pi_0(s_0) \prod_{i=1}^{N} A_{S_{i-1}S_i} \exp(-\int_{t_{start}}^{t_{end}} |A_{S(t)}| dt) \\
&= \pi_0(s_0)\alpha^N \prod_{i=1}^{N} f_{S_{i-1}S_i} \exp(-\alpha \sum_{i=0}^{N} F_{S_i}(\beta)(t_{i+1} - t_i))
\end{aligned}
$$

We can assume the prior distributions of α, β are $p_1(\alpha)$ and $p_2(\beta)$.

Then the posterior distribution of parameters α, β will be as follows.

$$
P(\alpha, \beta|s_0, S, T) \propto \alpha^N \prod_{i=1}^{N} f_{S_{i-1}S_i} \exp(-\alpha \sum_{i=0}^{N} F_{S_i}(\beta)(t_{i+1} - t_i)) p_1(\alpha) p_2(\beta)
$$

If we assume the priors of α, β are $Gamma(\mu, \lambda)$, $Gamma(\omega, \theta)$, then the posterior will have a simper form as follows.

$$P(\alpha, \beta | s_0, S, T) = C\alpha^{\mu+N-1} \exp(-\alpha(\lambda + \sum_{i=0}^{N} F_{S_i}(\beta)(t_{i+1} - t_i))) \prod_{i=1}^{N} f_{S_{i-1}S_i} \beta^{\omega-1} \exp(-\theta\beta)$$

We notice that given β, S, T, α is distributed as a $Gamma$ distribution.
$\alpha|\beta, S, T, y = \alpha|\beta, S, T \sim Gamma(\mu + N, \lambda + \sum_0^N F_{S_i}(\beta)(t_{i+1} - t_i))$.
There is no conjugate distribution to sample $\beta \sim P(\beta|s_0, S, T)$. We will have to use Metropolis Hasting within Gibbs to sample β. The target distribution is the following one.

$$P(\beta|S, T) = C \frac{\prod_{i=1}^{N} f_{S_{i-1}S_i}(\beta)\beta^{\omega-1} \exp(-\theta\beta)}{(\lambda + \sum_{i=0}^{N} F_{S_i}(\beta)(t_{i+1} - t_i))^{\mu+N}}.$$

Such doubling might slow the mixing of the Markov chain. We can apply our Metropolis Hasting algorithm on this model.

6.1 Experiments

In the following, we evaluate a Python implementation of our algorithms compared to other exact samplers which include Gibbs sampler and Particle MCMC sampler. We consider one special case when $f_{ij}(\beta) = \exp(-\beta/(i + j))$. We consider three different dimensions which are 3, 5, and 10 and three different k which are 1.5, 2, and 3. We generated random parameters α, β from prior distributions $(Gamma(3, 2), Gamma(5, 2))$, and used these parameters to construct the transition matrix A. Then we generate an MJP trajectory with a uniform initial distribution over states and the transition matrix A. The state of this MJP trajectory was observed via a Normal distribution with mean equal to the value of state and variance 1, and the proposal kernel is a lognormal distribution with location parameter $\log(\theta_{old})$ and scale parameterσ. Posterior samples given the observations were produced by a Python implementation of our algorithm. 100 MCMC runs were performed, each run consisting of 10000 iterations except for Particle MCMC algorithm. For Particle MCMC, each run consists 3000 iterations while the number of particles can be 5, 10 or 20. We also explored the gradient information of the target distribution to apply Hamiltonian Monte Carlo with different step sizes and different numbers of leapfrog jumps. For HMC, each run consists 20000 iterations while the numbers of leapfrog jumps can be 1, 3, 5, or 10, and the leapfrog

stepsize can be 0.02, 0.05 and 0.1. For each run, the acceptance rates as well as the time spent was calculated, and effective sample sizes (ESSs) of MCMC sampling parameters) were calculated using R-CODA (Plummer et al., 2006). The overall ESS per unit time of a run is defined to be the mean ESS per unit time across all these ESSs per unit time.
Figure 1, 2 and 3 plot the overall ESS per unit time against the variance of the proposal kernel per run, for different methods and different scaling parameters k($k = 1.5, 2, 3$) and different dimensions($p = 3, 5, 10$), where the $\Omega = k \max(\Omega_{old}, \Omega_{new})$ when $k < 2$, or $\Omega = k(\Omega_{old} + \Omega_{new})$ when $k \leqslant 2$. We see that the improved MH algorithm is more efficient in these cases with respect to the overall ESS per unit time. We also see that increase the scaling parameter will decrease the efficiency of the improved MH algorithm respect to overall ESS per unit time, when $k > 2$. If we set $\Omega = 1.5 \max(\Omega_{old}, \Omega_{new})$, the performance of the improved MH will not be as good as the case we set $\Omega = 2(\Omega_{old} + \Omega_{new})$ when the proposal log variance is large.
Figure 4 shows the initial burn-in of a sampler with this setting for different initializations. The vertical axis shows the number of state transitions in the MJP trajectory of each iteration. This quantity quickly reaches its equilibrium value within a few iterations.

Figure 5 plots ESS per unit time a Figure 6 plots ESS per unit time as observation interval and the number of observations both increase when the dimension is 3 and the scaling parameter k is 2. Gibbs sampler decreases faster than Metropolis Hasting Method, due to the doubling of MJP paths and the parameters.

Figure 7 plots ESS per unit time as observation interval increases with number of observations fixed, when the dimension is 3 and the scaling parameter k is 2.Gibbs sampler decreases faster than Metropolis Hasting Method, due to the doubling of MJP paths and the parameters. In addition Gibbs sampler decreases even more faster than previous case when the number of observations is not fixed.

Figure 8 plots ESS plots the overall acceptance rate against the log variance of the proposal kernel per run for dimension 3.

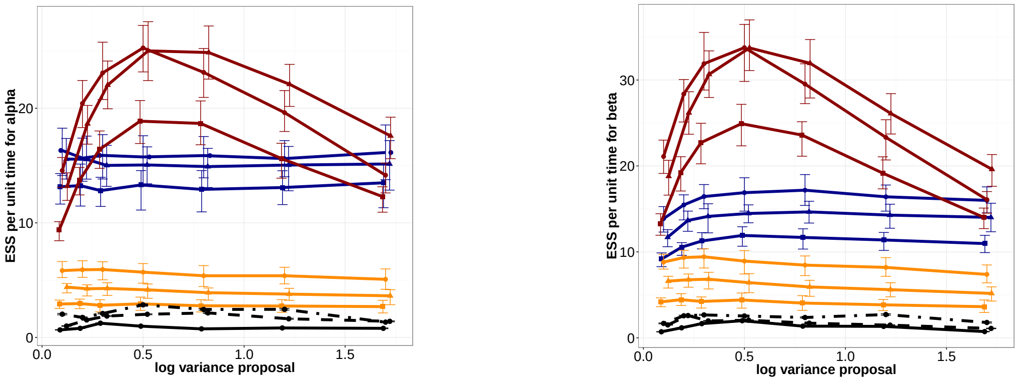

Figure 4: ESS/sec for exp model (dim 3). The left is for α, and the right is for β.

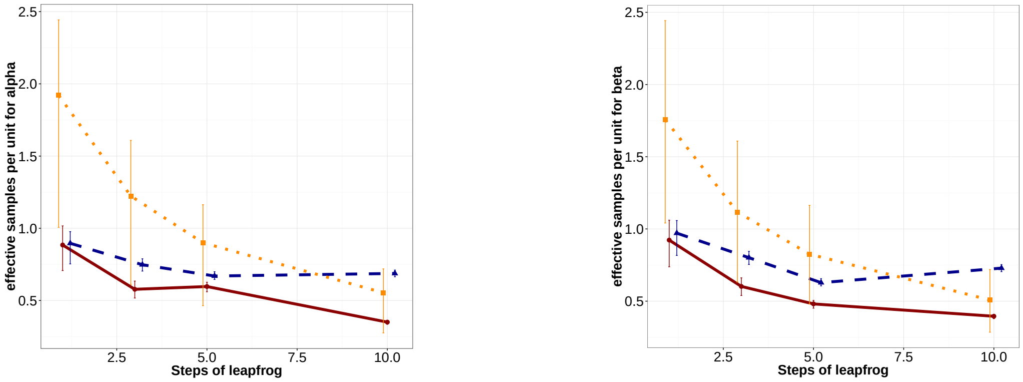

Figure 5: HMC for dim 3

7 Immigration models with capacity

An $M/M/N/N$ queue is a stochastic process whose state space is the set $\{0, 1, 2, 3, ..., N-1\}$ where the value corresponds to the number of customers in the system, including any currently in service. Arrivals occur at rate α according to a Poisson process and move the process from state i to $i+1$. Service times have an exponential distribution with parameter β in the $M/M/N/N$

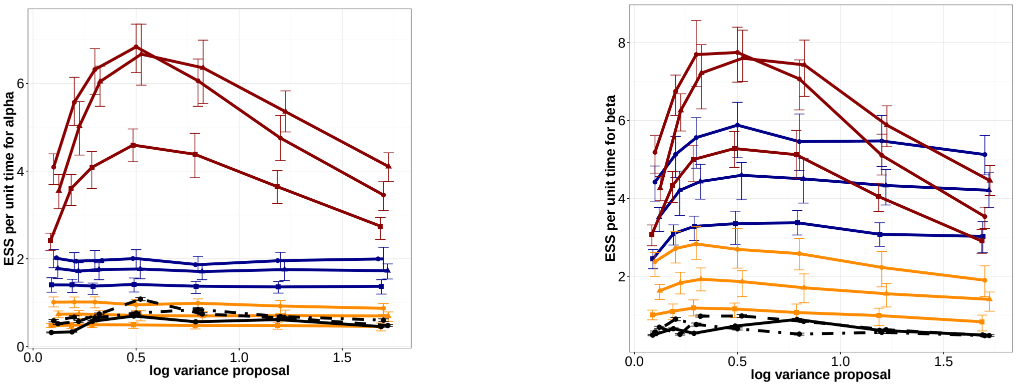

Figure 6: ESS/sec for exp model (dim 5)

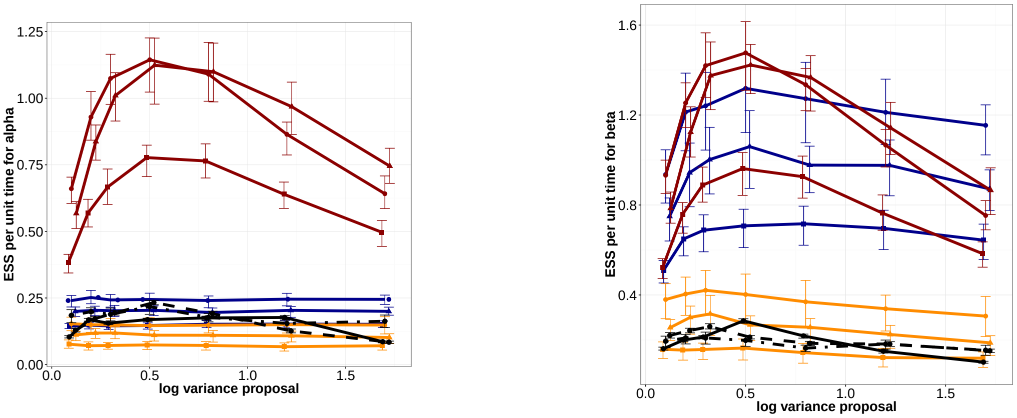

Figure 7: ESS/sec for exp model (dim 10)

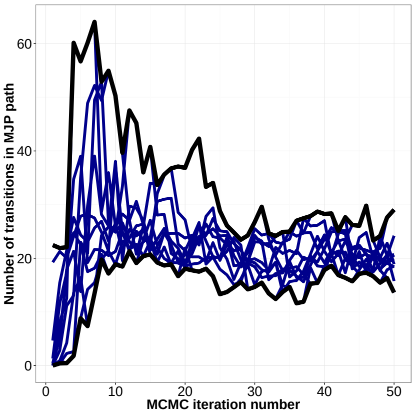

Figure 8: Trace plot of the number of MJP transitions for different initializatoins for exponential model.

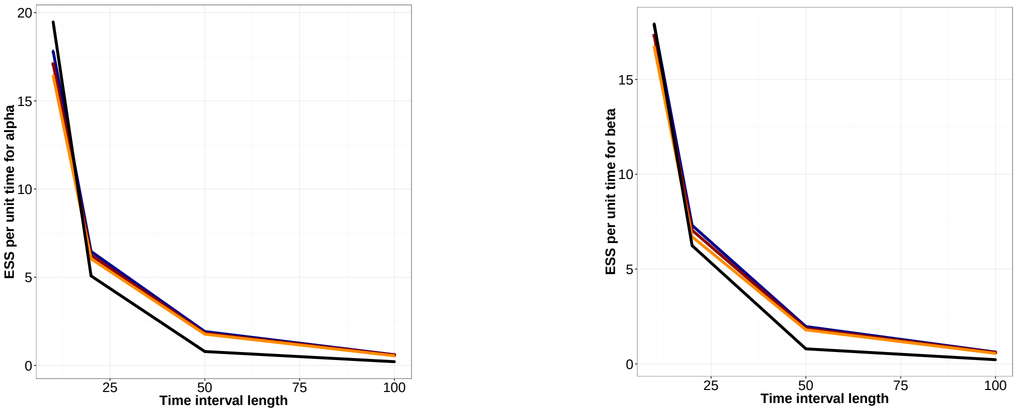

Figure 9: Time Interval vs. ESS / sec

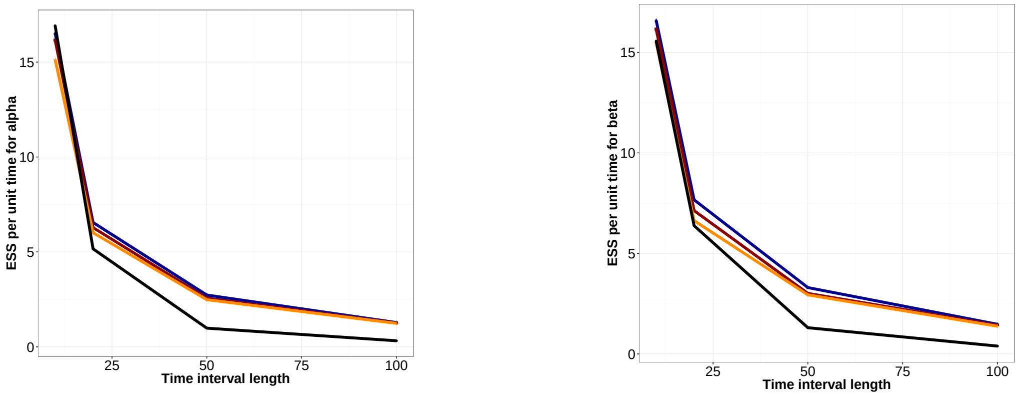

Figure 10: Time Interval vs. ESS / sec (Number of observations is fixed)

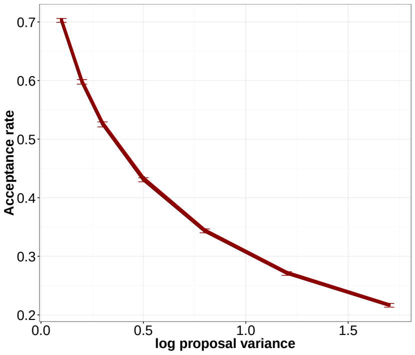

Figure 11: Acceptance rate for exp model (dim 3)

queue. There are N servers, which serve from the front of the queue. If there are less than N jobs, some of the servers will be idle. Only N customers can queue at any one time. Any further arrivals to the queue are considered "lost".

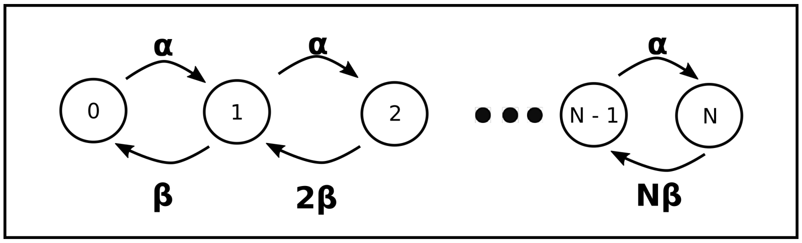

Figure 12: exp model

Assume: $S = [S_0, S_1, ..., S_N]$,$T = [t_0(t_{start}), t_1, ..., t_N, t_{N+1}(t_{end})]$, and y as observations.
Now, let's consider a immigration model as follows. State space is $\{0, 1, 2, ..., N-1\}$, representing the total population. The transition matrix is defined as follows.

$$A_i =: A_{i,i} = -(\alpha + i\beta), \quad i = 0, 1, ..., N$$

$$A_{i,i+1} = \alpha, \quad i = 0, 1, ..., N-1,$$

$$A_{i,i-1} = \beta, \quad i = 1, ..., N.$$

We already know the conditional density(given $\alpha,\ \beta$) of a MJP trajectory (s_0, S, T) in time interval $[t_{start}, t_{end}]$, with $S = (s_1, s_2, ..., s_k)$, $T = (t_1, t_2, ..., t_k)$.

$$f(s_0, S, T|\alpha, \beta) = \prod_{i=0}^{k-1} A_{s_i, s_{i+1}} \exp(\sum_{i=0}^{k} A_{s_i}(t_{i+1} - t_i)),$$

where $t_0 = t_{start}$, $t_{k+1} = t_{end}$.
Let's denote some notations here.

$$U(s_0, S, T) := \sum_{i=0}^{k-1} \mathbb{I}_{\{s_{i+1} - s_i = 1\}}.$$

$$D(s_0, S, T) := \sum_{i=0}^{k-1} \mathbb{I}_{\{s_{i+1} - s_i = -1\}}.$$

Call them U and D for short. Let's denote the total time when the trajectory state stays at state i as τ_i, i.e. $\tau_i = \sum_{j=0}^{k}(t_{j+1} - t_j)\mathbb{I}_{\{s_j = i\}}$, then $\sum_{i=0}^{k}(t_{i+1} - t_i)s_i = \sum_{i=0}^{N} \tau_i i$.

$$f(s_0, S, T|\alpha, \beta) = \exp(-\alpha(t_{end} - t_{start} - \tau_N))\alpha^U \cdot \exp((-(\sum_{i=0}^{k}(t_{i+1} - t_i)s_i)\beta)\prod_{i=1}^{N} i^{\sum_{j=0}^{k-1} \mathbb{I}_{s_{j+1}=i-1\ ,s_j=i}}\beta^D$$

If we assume the prior of α, and β are $Gamma(\mu, \lambda)$, $Gamma(\omega, \theta)$, which are independent with each other.

$$p(\alpha) = \frac{\lambda^\mu}{\Gamma(\mu)}\alpha^{\mu-1}e^{-\lambda\alpha}.$$
$$p(\beta) = \frac{\theta^\omega}{\Gamma(\omega)}\beta^{\omega-1}e^{-\theta\beta}.$$

Then we can get the posterior distribution

$$f(\alpha, \beta|s_0, S, T)$$

as follows.

$$f(\alpha, \beta|s_0, S, T) \propto \exp(-(\lambda + t_{end} - t_{start} - \tau_N)\alpha)\alpha^{\mu+U-1} \cdot \exp(-(\sum_{i=0}^{k}(t_{i+1} - t_i)s_i + \theta)\beta)\beta^{\omega+D-1}.$$

It means that the posterior distributions of α, β are still independent.
$\alpha|s_0, S, T$ is following $Gamma(\mu + U, \lambda + t_{end} - t_{start} - \tau_N)$
$\beta|s_0, S, T$ is following $Gamma(\omega + D, \theta + \sum_{i=0}^{k}(t_{i+1} - t_i)s_i)$, which is equivalent to $Gamma(\omega + D, \theta + \sum_{i=0}^{N} \tau_i i)$
Such immigration models have perfectly conjugate posterior distributions when we assign γ priors to α and β. We apply our Metropolis Hasting algorithms on such models to compare the performance with the performance of Gibbs Sampling algorithm.

7.1 Experiments

In the following, we evaluate a Python implementation of our algorithms compared to other exact samplers which include Gibbs sampler and Particle MCMC sampler. We consider three different dimensions which are 3, 5, and 10 and three different k which are 1.5, 2, and 3. We generated random parameters α, β from prior distributions ($Gamma(3, 2)$, $Gamma(5, 2)$), and used this to construct the transition matrix A. Then we generate an MJP trajectory with a uniform initial

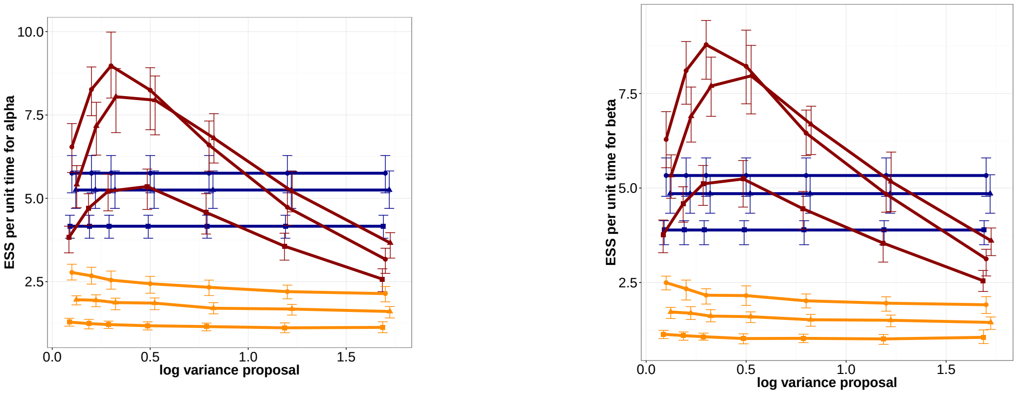

Figure 13: ESS/sec for Immigration model (dim 3)

distribution over states. The state of this MJP trajectory was observed via a Normal distribution with mean equal to the value of state and variance 1, and posterior samples given the observations were produced by a Python implementation of our algorithm. 100 MCMC runs were performed, each run consisting of 10000(Varies among different dimensions) iterations. For each run, the number of transitions as well as the time spent was calculated, and effective sample sizes (ESSs) of these statistics (the number of independent samples with the same 'information' as the correlated MCMC samples) were calculated using R-CODA (Plummer et al., 2006). The overall ESS of a run is defined to be the mean ESS across all these ESSs.

8 DNA evolution JC69 model

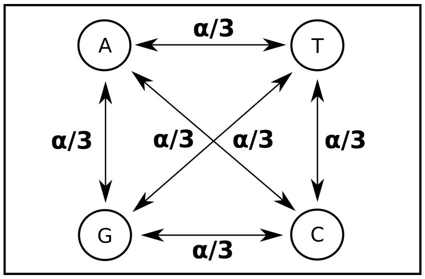

Figure 19: exp model

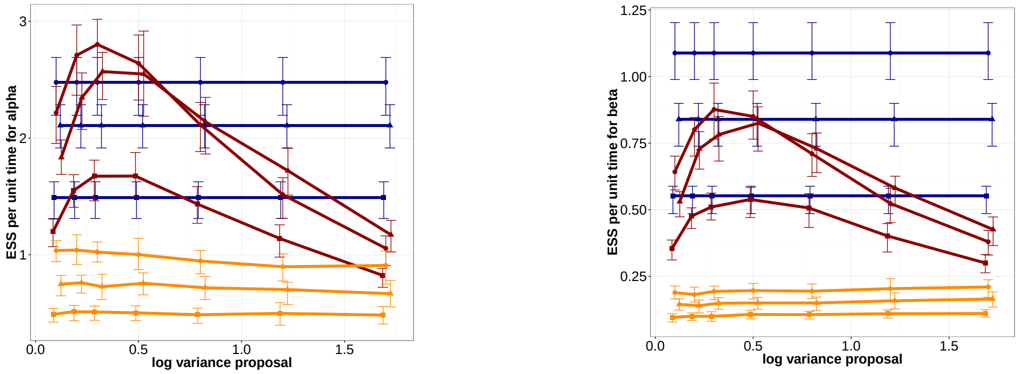

Figure 14: ESS/sec for Immigration model (dim 5)

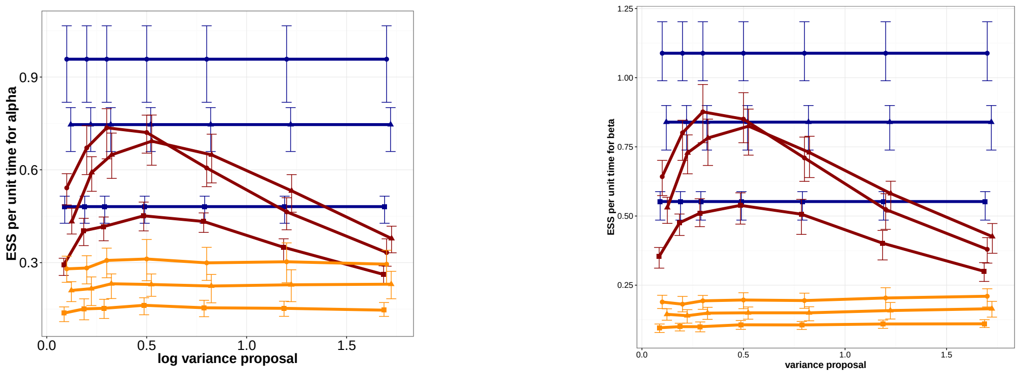

Figure 15: ESS/sec for Immigration model (dim 10)

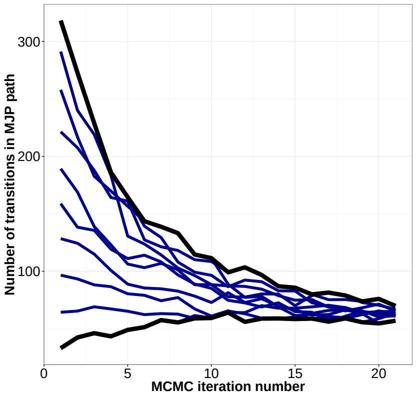

Figure 16: Trace plot of the number of MJP transitions for different initializatoins for immigration model.

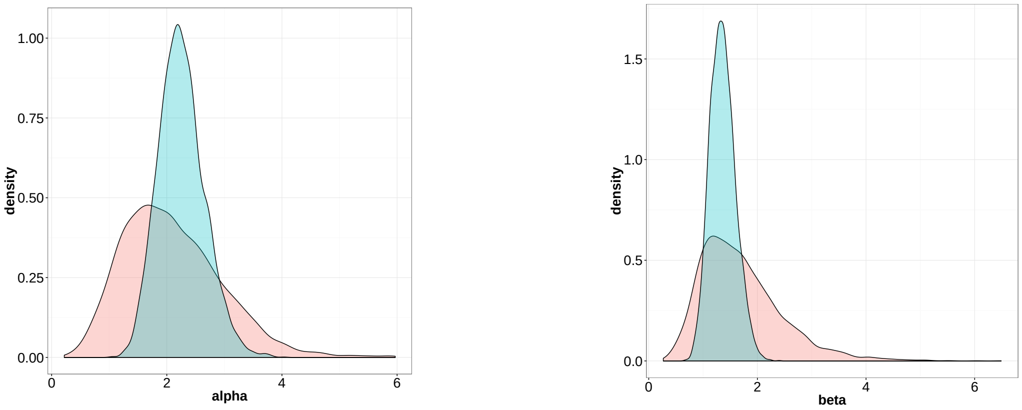

Figure 17: density

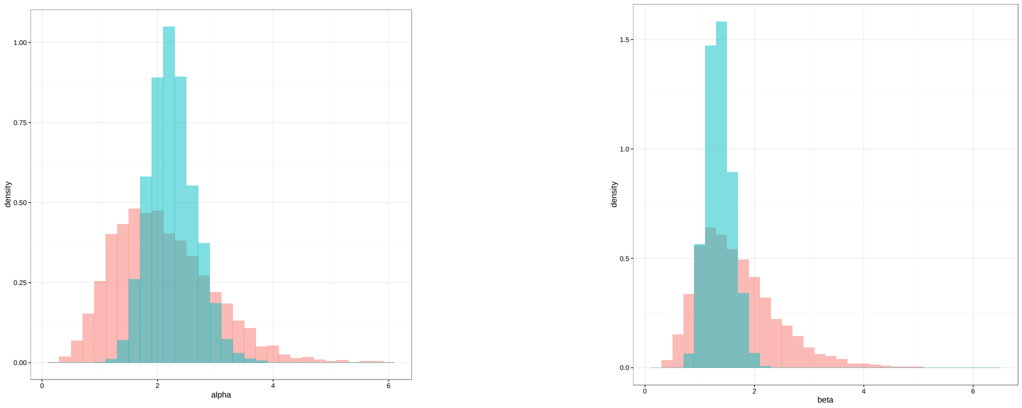

Figure 18: test

JC69 is the simplest substitution model. There are several assumptions. It assumes equal base frequencies and equal mutation rates. The only parameter of this model is α. The overall substitution rate is therefore 3α. The state space is $\{1, 2, 3, 4\}$, representing $\{A, T, C, G\}$.Assume: $S = [S_0, S_1, ..., S_N]$,$T = [t_0(t_{start}), t_1, ..., t_N, t_{N+1}(t_{end})]$, and y as observations.

$$A_i =: A_{i,i} = -3\alpha, \quad i = 0, 1, ..., N$$

$$A_{i,j} = \alpha, \quad i \neq j.$$

If we assume the prior of α is $Gamma(\mu, \lambda)$

$$p(\alpha) = \frac{\lambda^\mu}{\Gamma(\mu)} \alpha^{\mu-1} e^{-\lambda\alpha}$$

. Then we can get the posterior distribution

$$f(\alpha|s_0, S, T)$$

as follows.

$$f(\alpha|s_0, S, T) \propto \exp(-(\lambda + 3(t_{end} - t_{start}))\alpha)\alpha^{\mu+N-1}.$$

$\alpha|s_0, S, T$ is following $Gamma(\mu + N, \lambda + 3(t_{end} - t_{start}))$

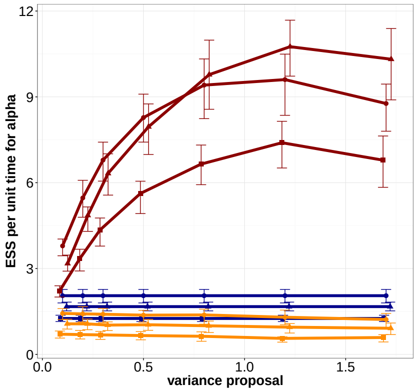

Figure 20: ESS/sec for JC69 Model

9 Experiments

In the following, we evaluate a Python implementation of our algorithms compared to other exact samplers which include Gibbs sampler and Particle MCMC sampler. We consider three different dimensions which are 3, 5, and 10 and three different k which are 1.5, 2, and 3. We generated random parameters α, β from prior distributions ($Gamma(3, 2), Gamma(5, 2)$), and used this to construct the transition matrix A. Then we generate an MJP trajectory with a uniform initial distribution over states. The state of this MJP trajectory was observed via a Normal distribution with mean equal to the value of state and variance 1, and posterior samples given the observations were produced by a Python implementation of our algorithm. 100 MCMC runs were performed, each run consisting of 10000(Varies among different dimensions) iterations. For each run, the number of transitions as well as the time spent was calculated, and effective sample sizes (ESSs) of these statistics (the number of independent samples with the same 'information' as the correlated MCMC samples) were calculated using R-CODA (Plummer et al., 2006). The overall ESS of a run is defined to be the mean ESS across all these ESSs.

10 Immigration models with capacity with piece wise constant rate

We consider the Queuing model, with piece wise constant transition rate. Assume: $S = [S_0, S_1, ..., S_N]$,$T = [t_0(t_{start}), t_1, ..., t_N, t_{N+1}(t_{end})]$, and y as observations.
Now, let's consider a immigration model as follows. State space is $\{0, 1, 2, ..., N-1\}$, representing the total population. The transition matrix is defined as follows.

$$A_i(t) =: A_{i,i}(t) = -(\alpha + i\beta)w(t), \;\; i = 0, 1, ..., N$$

$$A_{i,i+1}(t) = \alpha w(t), \;\; i = 0, 1, ..., N-1,$$

$$A_{i,i-1}(t) = \beta w(t), \;\; i = 1, ..., N.$$

$w(t)$ is a piece wise constant function. $w(t) = w_i, \; t \in [l_i, l_{i+1}), i = 1, 2, 3, ..., K.$
We already know the conditional density(given $\alpha, \; \beta$) of a MJP trajectory (s_0, S, T) in time interval $[t_{start}, t_{end}]$, with $S = (s_1, s_2, ..., s_k)$, $T = (t_1, t_2, ..., t_k)$.

$$f(s_0, S, T|\alpha, \beta) = \prod_{i=0}^{k-1} A_{s_i, s_{i+1}}(t_i) \exp(\sum_{i=0}^{k} A_{s_i}(t_i)(t_{i+1} - t_i)),$$

where $t_0 = t_{start}, \; t_{k+1} = t_{end}$.
Let's denote some notations here.

$$U(s_0, S, T) := \sum_{i=0}^{k-1} \mathbb{I}_{\{s_{i+1} - s_i = 1\}}.$$

$$D(s_0, S, T) := \sum_{i=0}^{k-1} \mathbb{I}_{\{s_{i+1} - s_i = -1\}}.$$

Call them U and D for short. Let's denote the total time when the trajectory state stays at state i as τ_i, i.e. $\tau_i = \sum_{j=0}^{k}(t_{j+1} - t_j)\mathbb{I}_{\{s_j = i\}}$, then $\sum_{i=0}^{k}(t_{i+1} - t_i)s_i = \sum_{i=0}^{N} \tau_i i$.

$$f(s_0, S, T|\alpha, \beta) \propto \exp(\sum_{r=0}^{K} -w_r \alpha(l_{r+1} - l_r - \tau_N^r))\alpha^U \cdot \exp(-\int_{t_s}^{t_e}(S(t)w(t)\beta)\beta^D$$

If we assume the prior of α, and β are $Gamma(\mu, \lambda)$, $Gamma(\omega, \theta)$, which are independent with each other.

$$p(\alpha) = \frac{\lambda^{\mu}}{\Gamma(\mu)}\alpha^{\mu-1}e^{-\lambda\alpha}.$$
$$p(\beta) = \frac{\theta^{\omega}}{\Gamma(\omega)}\beta^{\omega-1}e^{-\theta\beta}.$$

Then we can get the posterior distribution

$$f(\alpha,\beta|s_0,S,T)$$

as follows.

$$f(\alpha,\beta|s_0,S,T) \propto \exp(-(\lambda + \sum_{r=0}^{K} w_r\alpha(l_{r+1}-l_r-\tau_N^r))\alpha)\alpha^{\mu+U-1}\cdot\exp(-(\int_{t_s}^{t_e}(S(t)w(t)+\theta)\beta)\beta^{\omega+D-1}.$$

It means that the posterior distributions of α, β are still independent.
$\alpha|s_0,S,T$ is following $Gamma(\mu+U, \lambda+\sum_{r=0}^{K} w_r\alpha(l_{r+1}-l_r-\tau_N^r))$
$\beta|s_0,S,T$ is following $Gamma(\omega+D, \int_{t_s}^{t_e}(S(t)w(t)+\theta)$.
Such immigration models have perfectly conjugate posterior distributions when we assign γ priors to α and β. We apply our Metropolis Hasting algorithms on such models to compare the performance with the performance of Gibbs Sampling algorithm.

10.1 Experiments

In the following, we evaluate a Python implementation of our algorithms compared to the Gibbs sampler. We consider three different dimensions which are 3, 5, and 10 and three different k which are 1.5, 2, and 3. We generated random parameters α, β from prior distributions $(Gamma(3,2), Gamma(5,2))$. We set w as $(1,2,3,4)$ and l as $(0,5,10,15,20)$ We used this to construct the transition matrix A. Then we generate an MJP trajectory with a uniform initial distribution over states. The state of this MJP trajectory was observed via a Normal distribution with mean equal to the value of state and variance 1, and posterior samples given the observations were produced by a Python implementation of our algorithm. 100 MCMC runs were performed, each run consisting of 5000(Varies among different dimensions) iterations. For each run, the number of transitions as well as the time spent was calculated, and effective sample sizes (ESSs) of these statistics (the number of independent samples with the same 'information' as the correlated

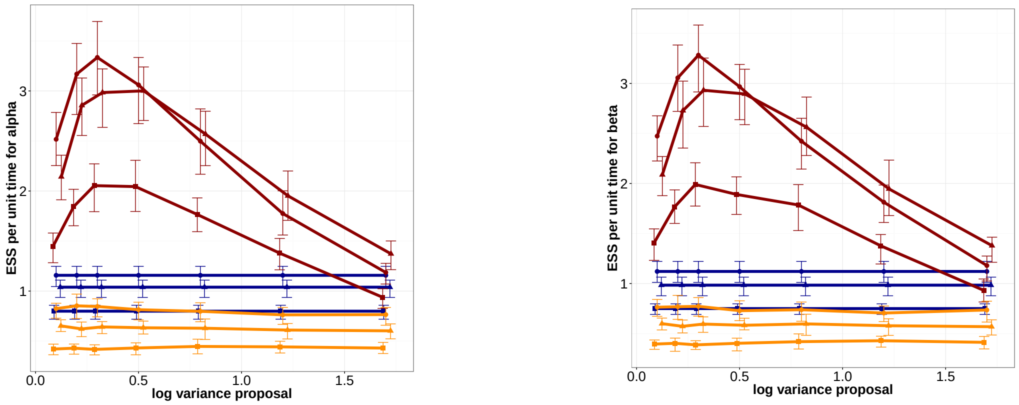

Figure 21: ESS/sec for NH Immigration model (dim 3)

MCMC samples) were calculated using R-CODA (Plummer et al., 2006). The overall ESS of a run is defined to be the mean ESS across all these ESSs.

[9] [8] [1] [2] [3] [4] [5] [7] [6]

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society*, B, 2010.

[2] C. Andrieu and G. O. ROBERTS. The pseudo-marginal approach for efficient monte carlo computations. *Annals of Statistics*, 37(2), 2009.

[3] A. Golightly, D. A. Henderson, and C. Sherlock. Delayed acceptance particle mcmc for exact inference in stochastic kinetic models. *Statistics and Computing*, 25(Issue 5), 2015.

[4] A. Golightly, D. A. Henderson, and C. Sherlock. Delayed acceptance particle mcmc for exact inference in stochastic kinetic models. *Statistics and Computing*, 25(Issue 5), 2015.

[5] Jun S. Liu. The collapsed gibbs sampler in bayesian computations with applications to gene regulation problem. *American Statistical Association*, 89(427), 1994.

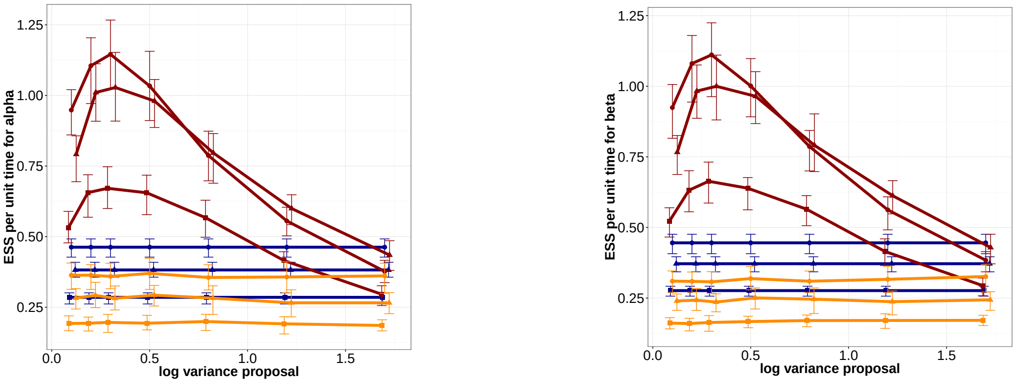

Figure 22: ESS/sec for NH Immigration model (dim 5)

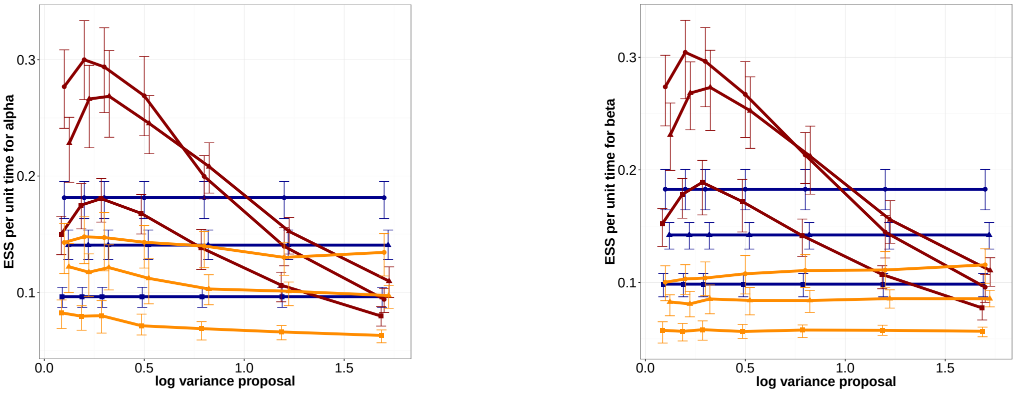

Figure 23: ESS/sec for NH Immigration model (dim 10)

[6] Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3), 2003.

[7] Radford M. Neal. Mcmc using hamiltonian dynamics. *the Handbook of Markov Chain Monte Carlo*, Chapter 5, 2011.

[8] V. Rao and Y. W. Teh. Mcmc for continuous-time discrete-state systems. *In Advances in Neural Information Processing Systems*, (24), 2012.

[9] V. Rao and Y. W. Teh. Fast mcmc sampling for markov jump processes and extensions. *Journal of Machine Learning Research*, 2013.