# 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

January 22, 2017

**Abstract**

Markov jump processes (MJPs) are continuous-time stochastic processes that find wide application in a variety of fields. Inference for MJPs typically proceeds via Markov chain Monte Carlo, the state-of-the-art being a recent auxiliary variable Gibbs sampler proposed in **?**. This algorithm was designed for the situation where the MJP parameters are known, and Bayesian inference over the unknown parameter is typically carried out by incorporating it into a larger Gibbs sampler. In many situations, the MJP trajectory and parameters can exhibit strong coupling, and this strategy of alternately sampling parameters given path, and then path given parameters can result in poor mixing. In this work, we propose a simple, elegant and novel algorithm to address this problem. Our scheme shows how standard Metropolis-Hastings (MH) approaches relevant to discrete-time hidden Markov models (HMMs) can be extended to the continuous-time MJP. Our proposed solution also ties up some of the loose ends in **?**, and provides a complete and clean recipe for Bayesian inference in jump processes. In our experiments, we demonstrate superior performance over Gibbs sampling, as well as other approaches like particle Markov chain Monte Carlo **?**.

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

## 1 Introduction

Markov jump processes (MJPs) are continuous-time stochastic processes that have found wide application in fields like computational chemistry, population genetics, mathematical finance, artificial intelligence and social-network analysis. MJPs have been used to model phenomena such as the state of a chemical reaction **?**, the state of a queuing network **?**, segmentations of a strand

of DNA **?**, the state of users logging their activity on social media **?** among many others Their temporal evolution in continuous-time nature results in realistic, mechanistic, and interpretable models, often amenable to mathematical analysis. These same dynamics however raise computational challenges in statistical applications, where given partal and noisy measurements, one has to make inferences over the latent MJP trajectory as well as any system parameters. Carrying out inferences over the latent MJP trajectory, and any system parameters, is complicated by two facts: one cannot *a priori* bound the number of transitions, and state-transition times are continuous-valued. Inference for these models typically proceeds via Markov chain Monte Carlo, the state-of-the-art being a recent auxiliary variable Gibbs sampler proposed in **?**. This algorithm was designed to sample MJP paths when the MJP parameters are known, and parameter inference is typically carried out by incorporating it into a larger Gibbs sampler.

In many situations, the MJP trajectory and parameters can exhibit strong coupling, so that the Gibbs described earlier can mix poorly. In this work, we propose a Metropolis-Hastings framework to address this issue. Our proposed solution is simple, elegant and ties up some of the loose ends in the algorithm from **?**. In our experiments, we demonstrate superior performance over Gibbs sampling, as well as other approaches like particle Markov chain Monte Carlo **?**.

## 2 Markov jump processes

Formally, an MJP is a right-continuous piecewise-constant stochastic process $S(t)$ taking values in a countable, and usually finite state space (see Figure **??**). For simplicity, we will assume $N$-states, with $= \{1, \ldots, n\}$. Then, an MJP is parameterized by two quantities, a probability vector $\pi$ and a rate-matrix $A$. The former, an $n$-component vector, gives the distribution over states at the initial time (which without loss of generality, we assume is 0), while the latter is an $n \times n$-matrix governing the dynamics of the system. An off-diagonal element $A_{ij}$, for some $i \neq j$ gives the rate at which the system transitions from state $i$ to $j$. The rows of $A$ sum to 0, so that the diagonal element $A_i \equiv A_{ii} = -\sum_{j \neq i} A_{ij}$, and its absolute value gives the total rate at which the system leaves state $i$ for any other state. To simulate an MJP trajectory, one first samples an initial state $S(0)$ from the distribution $\pi$, after which we repeat the following two steps:

- Sample a wait-time $t_i$ from an exponential with rate $A_i$, where $i$ is the current state. The

MJP remains in state $i$ for time $t_i$.

- At the end of this time, jump to a new state $j \neq i$ with probability proportional to $A_{ij}$.

### 2.1 Structured rate matrices

In the general case, the rate matrix $A$ has $n(n-1)$ free parameters, corresponding to transition rates between every pair of distinct states. In typical applications, and especially when large state-spaces are involved, this $n \times n$ matrix is determined by a much smaller set of parameters. We will write these as $\theta$, and the rate matrix $A$ is a deterministic function of these parameters: $A \equiv A(\theta)$. The parameters $\theta$ are often much more interpretable than the elements of $A$, corresponding directly to physical, biological or environmental parameters of interest. Below we give three examples:

**The immigration-death process** This is a simple MJP governed by two parameters: an arrival rate $\alpha$ and a 'death'-rate $\beta$. The state space represents the size of a population or the number of jobs in a queue, and can be finite or countably infinite. New individuals enter the system according to a rate-$\alpha$ Poisson process, so that the off-diagonal elements $A_{i,i+}$ all equal to $\alpha$. On the other hand, each individual dies at a rate $\beta$, so that the system moves from state $i$ to $i-1$ with rate $i\beta$. All other transitions have rate 0, so that $\theta = (\alpha, \beta)$, and $(\theta)$ is a tri-diagonal matrix.

**Birth-death processes** This is a simple variant of the immigration where the system moves from state $i$ to $i+1$ with rate $i\alpha$, so that the population grows at a rate proportional to the population size. Once again, $\theta = (\alpha, \beta)$.

**Jules-Cantor model** This is popular model in genetics used to characterize the transition rates between amino-acids at a locus over evolutionary time. In this model, transitions are categorized into two types: synonymous transitions that encode

## 3 Bayesian inference for MJPs

In realistic situations, one only observes the MJP trajectory at a finite set of times, and typically, these observations themselves are noisy. There are then two challenges than the practitioner faces:

- What is the MJP trajectory underlying the observations?
- What are the unknown parameters governing the MJP dynamics?

## 3.1 Trajectory inference for MJPs

This problem was addressed in **?**, and extended to a broader class of MJPs (as well as other jump processes like semi-Markov jump processes ) in **?**. Both these schemes are centered on an alternate approaches to sampling MJP trajectories by introducing auxiliary *thinned* candidate jump ideas. **?** follows a classical approach called uniformization, while in **?**, this was extended to a more general dependent thinning approach. We outline the latter below.

Recall that the diagonal elements of the rate matrix $A_i$ give the rate at which the MJP leaves state $i$ for any other state. Importantly, the system is set up so that self-transitions cannot occur. Now, for each parameter $A_i$, introduce an additional parameter $B_i \geqslant A_i$; **?** suggest setting $B_i = 2A_i$. Assuming the system is in state $i$, we sample a *candidate* transition time from an exponential, not with rate $B_i$. At this time, the system remains in its current state with probability $1 - A_i/B_i$. With the remaining probability, the system transitions to a new state, and as with Gillespie's algorithm, we move to state $j \neq i$ with probability proportional to $A_{ij}$. In **?**, it was shown that trajectories sampled this way have the same distribution as trajectories sampled according to Gillespie's algorithm.

Introducing the thinned variables allowed **?** to develop a novel and efficient MCMC sampler. The algorithm proceeds as follows:

**Given the MJP trajectory** $(S, T)$**, sample a new set of thinned candidate times** $U$: **?** show that these thinned events are distributed as a piecewise-constant inhomogeneous Poisson process with intensity $B_{S(t)} - A_{S(t)}$.

**Given the thinned and actual transition times** $(T \cup U)$ **from the last iteration, sample a new trajectory**: Conditioned on the skeleton $T \cup U$, the set of candidate jump times is fixed, and trajectory inference reduces to inference for the familiar discrete-time hidden Markov model (HMM) with transition matrix $B$. Between any two consecutive time points, the system remain in a fixed state, with the likelihood for a state $i$ consisting of two parts: the likelihood of all observations falling in that interval multiplied by a term $B_i \exp(-B_i \Delta t)$.

**?** show that the resulting Markov chain targets the desired posterior distribution over trajectories, and is ergodic for any choice of $B$ with $B_i$ strictly greater than $A_i$.

# 4 Parameter inference for MJPs

In practice, the MJP parameters themselves are unknown: often, these are the quantities of primary interest. A Bayesian approach places a prior over these unknown variables, and follows a Gibbs sampling approach to draw samples from the posterior: for an arbitrary initialization, sample a trajectory from the conditional $p(S(t)|X, \theta)$, and then, sample a new $\theta$ from the conditional $p(\theta|X, S(t))$. This distribution depends on a set of sufficient statistics of the MJP trajectory: how much time is spent in each state, and the number of transitions between each pair of states. Given these, a new parameter set can be sampled using any Markov kernel such as a Metropolis-Hastings update, a Hamiltonian Monte Carlo update, or in special circumstances, $\theta$ can be directly sampled from its conditional distribution.

Such conditional updates however come with a well known limitations: when the paths and parameters are stongly coupled, the resulting Gibbs sampler can be very inefficient, exploring parameter and path space very sluggishly. In Figure **??** we show both the distribution of a component of $\theta$ conditioned only on the observations, and conditioned on both the observations as well as a realization of the MJP trajectory: observe how much more concentrated the latter is compared to the former. The coupling is strengthened for longer and longer trajectories, so that the Gibbs sampler can mix very poorly for situations with long observation periods, even if the observations themselves are sparse and uninformative.

For the discrete-time situation, this problem of parameter-trajectory can be circumvented by marginalizing out the MJP trajectory and directly carrying out inference over the parameters using a Metropolis-Hastings algorithm. This scheme exploits the fact that the forward-backward algorithm used to sample a new trajectory retures the marginal probability of the observations $p(X|\theta)$ as a by product. This resulting algorithm them proceeds as follows:

- Propose a new parameter $\theta^*$ from some proposal distribution $q(\theta * |\theta)$
- Run the forward pass of the forward-backward algorithm to obtain the marginal likelihood of the observations, $p(X|\theta^*)$

- Accept this according to the usual Metropolis-Hastings acceptance probability.
- If desired, as new trajectory sample can be obtained by completing the backward pass of the forward-backward algorithm.

Naively calculating this marginal probability for the continuous-time situation is not straightforward: this requires matrix-exponentials to integrate out the now infinite number of paths, and this operation is expensive, loses sparsity in the structure of the rate matrix, and involves expensive computations that scale with the number of observations rather than the actual dynamics of the system of interest. In **?**, the authors demonstrate the benefits of the uniformization approach over such matrix exponential-based approaches, and it is important to develop similar approaches for parameter inference.

The the key idea of the dependent-thinning approach of **?** is to alternate a discrete-time HMM sampling step with a step that samples a new random grid. This naturally suggests incorporating the MH update step outlined earlier in a similar algorithm that also updates parameters. As we will show, this approach does not quite work, despite marginalizing out the MJP state the dependence of the random grid on the MJP parameters can still slow down mixing.

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

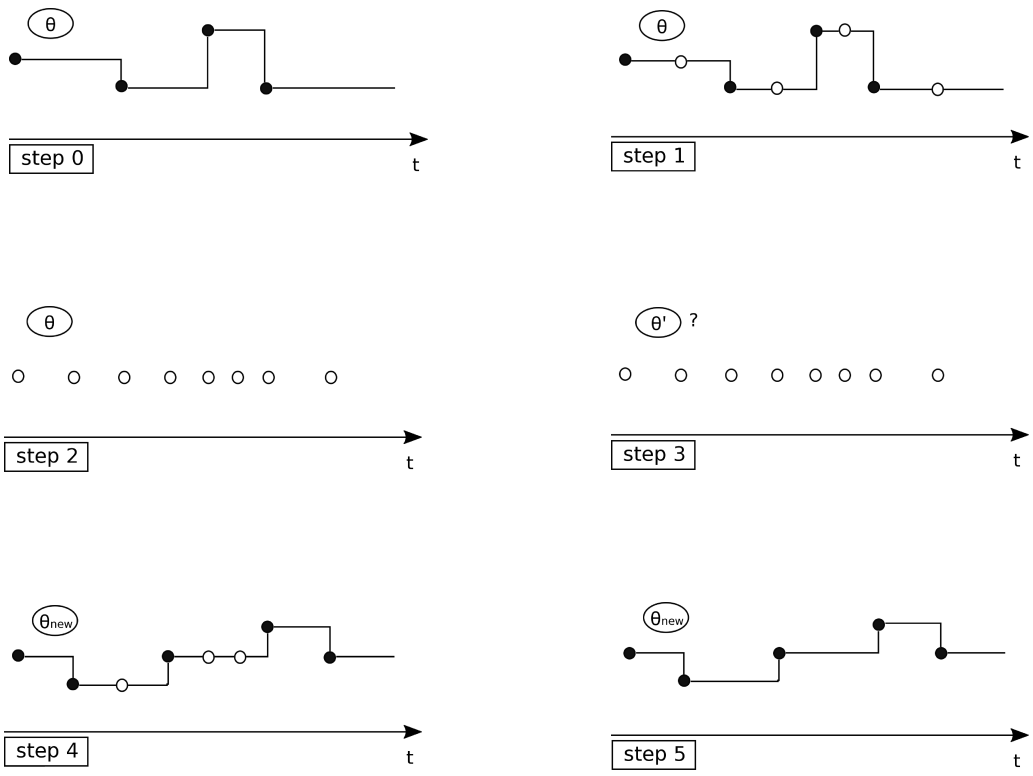


Figure 1: MH algorithm

**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 trajectory $\tilde{S}(t) = (\tilde{S}, \tilde{T})$, A series of MJP parameters $\tilde{\theta}$.

1: Let $\Omega = h(\theta)$, with $\Omega > \max_s |A_s|$ using some deterministic function $h$.

2: 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 $W = T \cup U$.

3: Propose $\theta^* \sim q(.|\theta)$.

Accept $\theta^*$ as $\tilde{\theta}$ with probability $\alpha$.

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

4: 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

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

5: 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 $(\tilde{S}, \tilde{T}, \tilde{\theta})$.

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

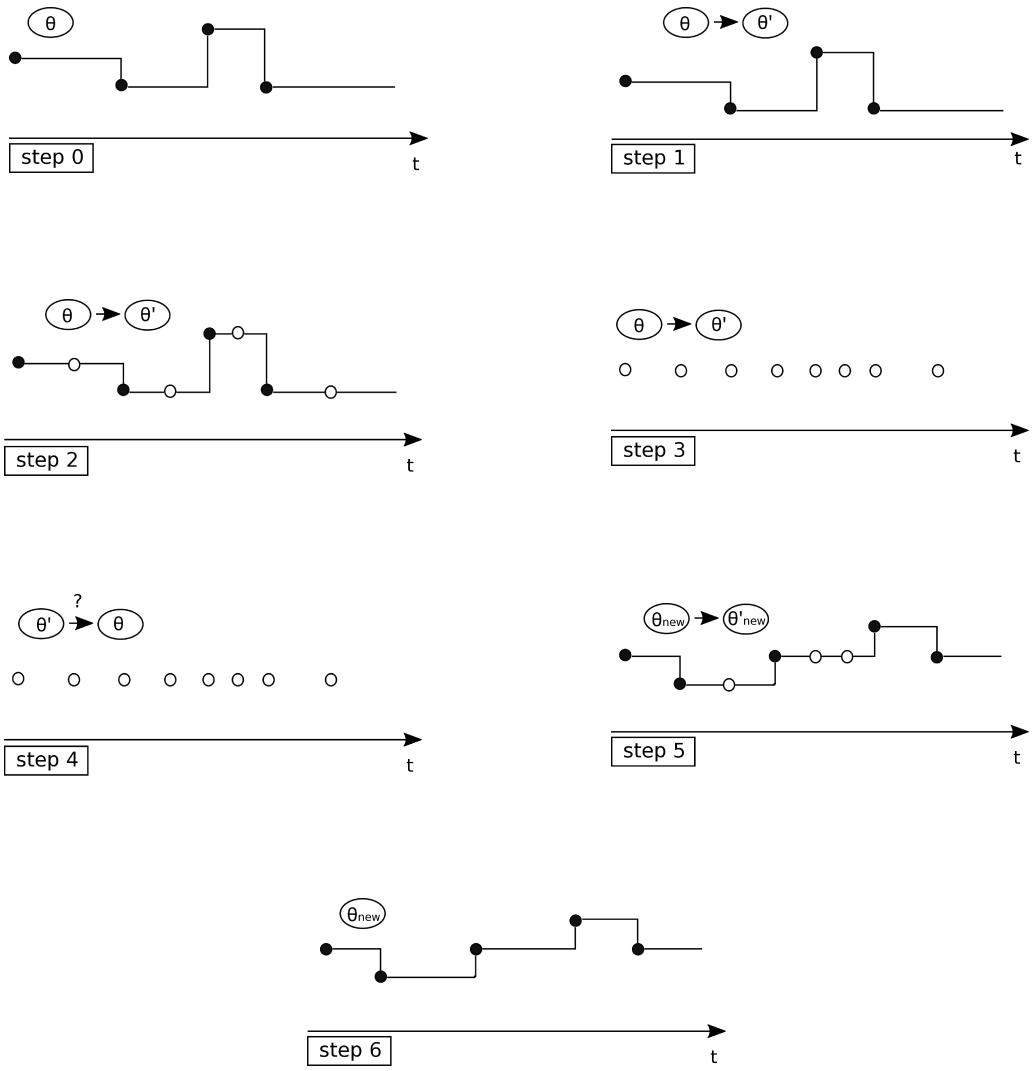


Figure 2: Improved MH algorithm

**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 $\pi_0$, 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: 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 $(\theta^*, \theta)$ and accept $\theta^*$ as $\tilde{\theta}$ with probability $\alpha$.

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

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 $\pi_0$. 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 $(\tilde{S}, \tilde{T}, \tilde{\theta})$.

# 7 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 $\alpha_t$ 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 $\kappa_1$ 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.

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

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

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

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

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

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

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, $\int\kappa_3\cdot(\int\kappa_2\circ\kappa_1\cdot P(W,S,T,\theta|y)d\theta dSdT)dW = \int\kappa_3 P(W,S^*,T^*,\theta^*|y)dW = P(W^*,S^*,T^*,\theta^*|y)$.
So the stationary distribution of $\kappa_3\circ\kappa_2\circ\kappa_1$ is $P(W^*,S^*,T^*,\theta^*|y)$.

# 8 Verifications of Algorithm 2

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

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

The marginal distribution of $(y, S, T, \theta, \theta^*)$ and $(y, S, T, \theta)$ 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 $\theta^*$ given $(y, S, T, \theta)$ 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 $(S, T, \theta)$ is the corresponding marginal distribution $P(S, T, \theta|y)$.

# 9 Generic Exponential Model

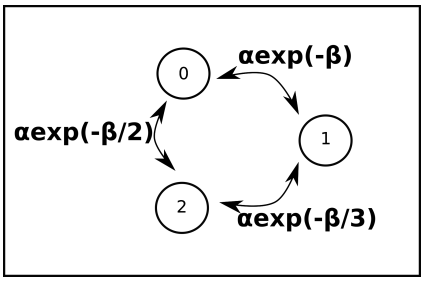


Figure 3: exp model

**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** $i = N$

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 $F_i(\beta) = \sum_{j \neq i} f_{ij}(\beta)$.

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

We can assume the prior distributions of $\alpha, \beta$ are $p_1(\alpha)$ and $p_2(\beta)$.
Then the posterior distribution of parameters $\alpha, \beta$ 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 $\alpha$, $\beta$ 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 $\beta$, $S$, $T$, $\alpha$ 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 $\beta$. 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.

## 9.1 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 $\alpha$, $\beta$ 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 parameter$\sigma$. 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

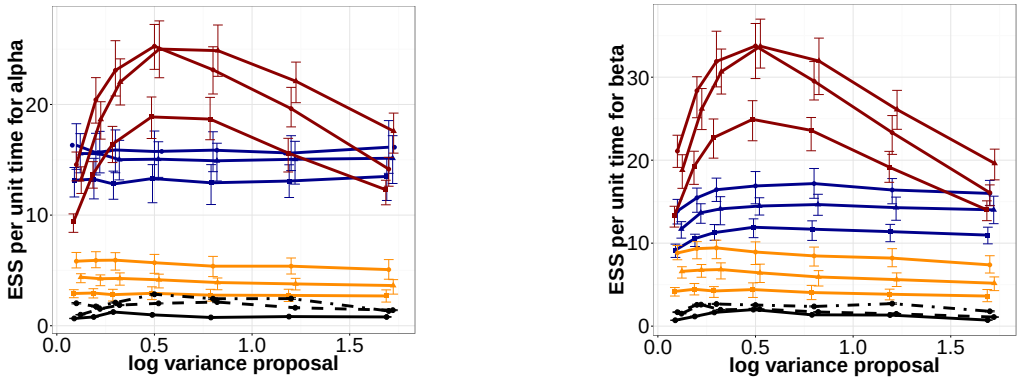


Figure 4: ESS/sec for exp model (dim 3). The left is for $\alpha$, and the right is for $\beta$.

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.

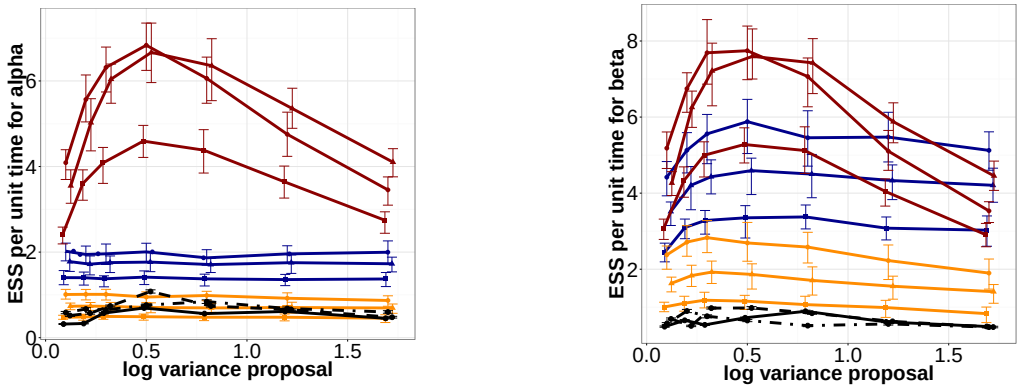


Figure 6: ESS/sec for exp model (dim 5)

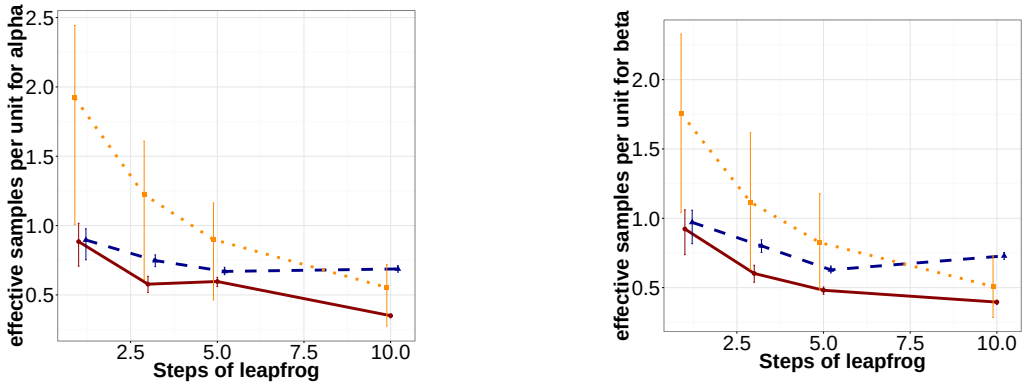


Figure 5: HMC for dim 3

# 10 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 $\alpha$ according to a Poisson process and move the process from state $i$ to $i+1$. Service times have an exponential distribution with parameter $\beta$ in the $M/M/N/N$

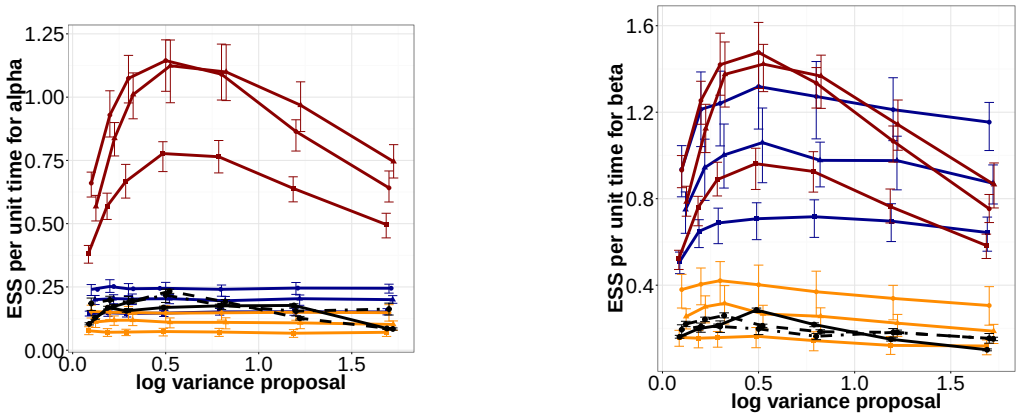


Figure 7: ESS/sec for exp model (dim 10)

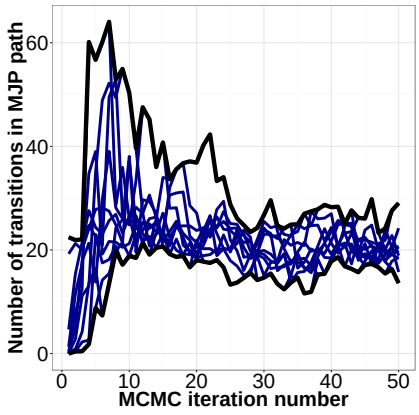


Figure 8: Trace plot of the number of MJP transitions for different initializatoins for exponential model.

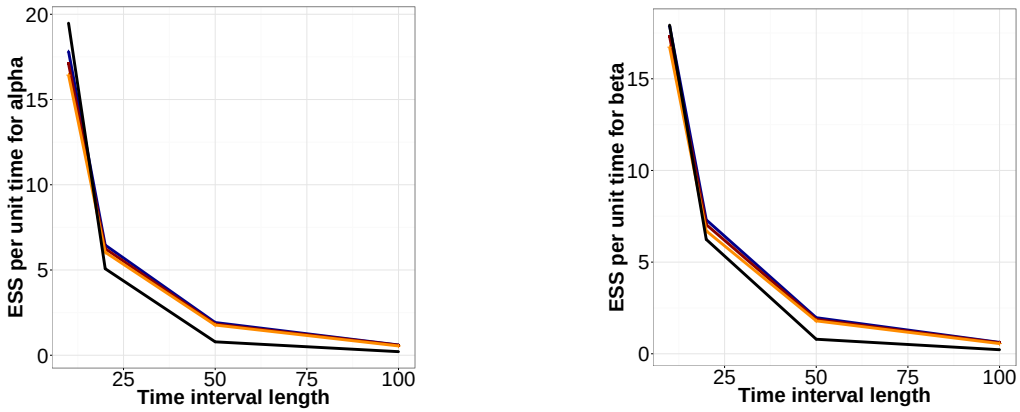


Figure 9: Time Interval vs. ESS / sec

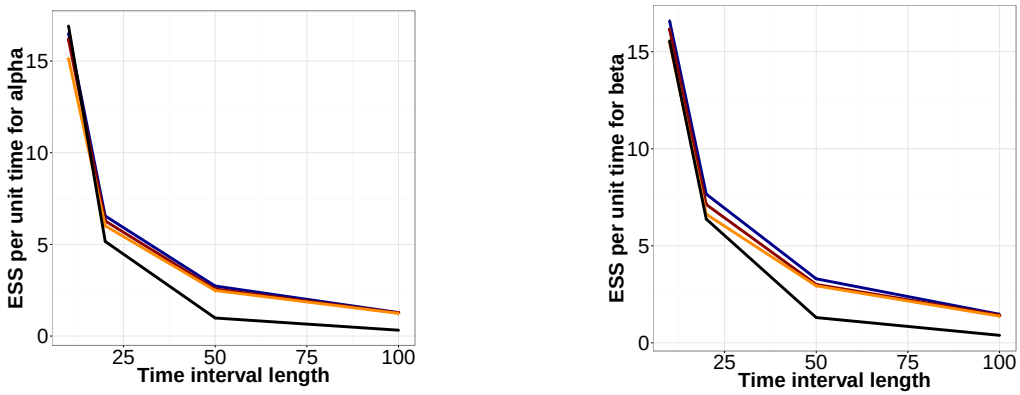


Figure 10: Time Interval vs. ESS / sec (Number of observations is fixed)

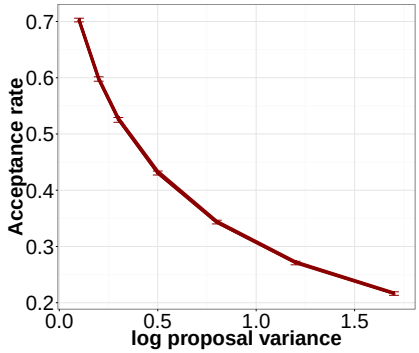


Figure 11: Acceptance rate for exp model (dim 3)

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".

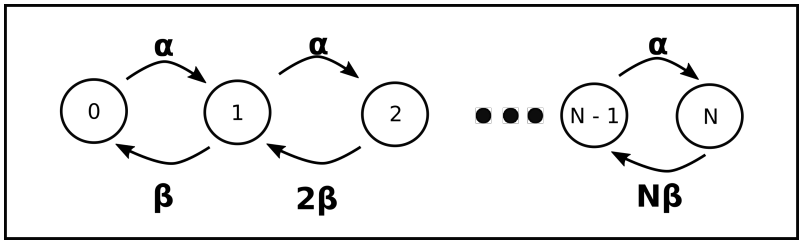


Figure 12: exp model

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 $\tau_i$, 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 $\alpha$, and $\beta$ 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 $\alpha$, $\beta$ 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 $\gamma$ priors to $\alpha$ and $\beta$. 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 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 $\alpha$, $\beta$ 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.

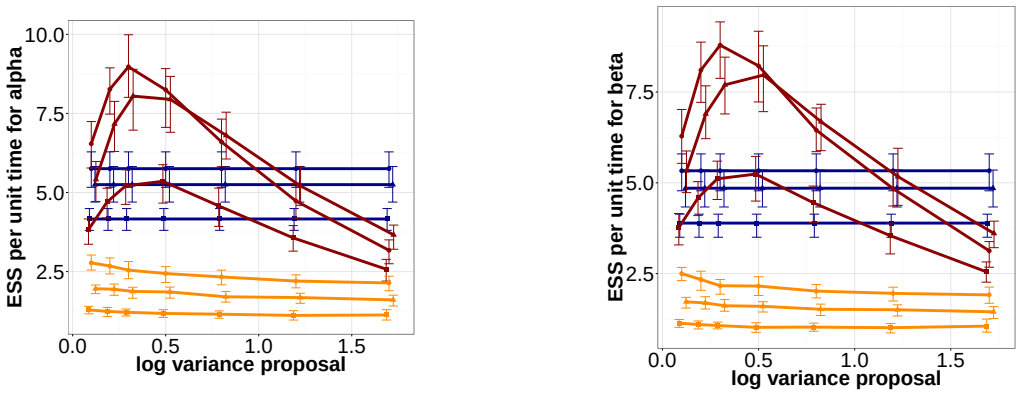


Figure 13: ESS/sec for Immigration model (dim 3)

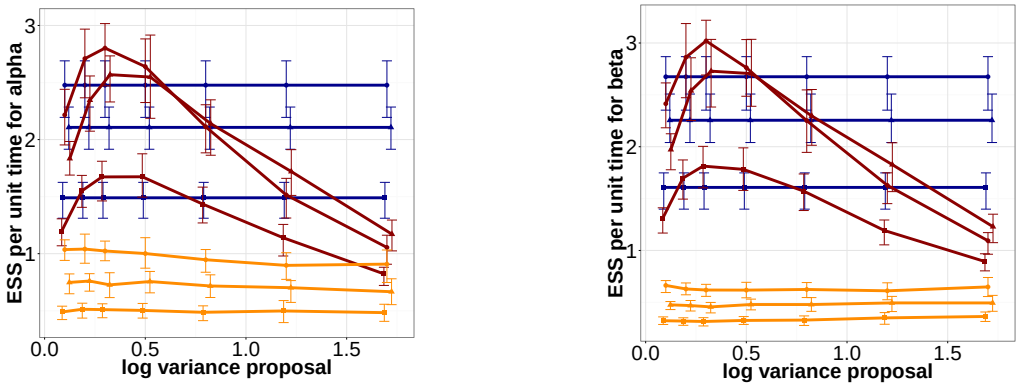


Figure 14: ESS/sec for Immigration model (dim 5)

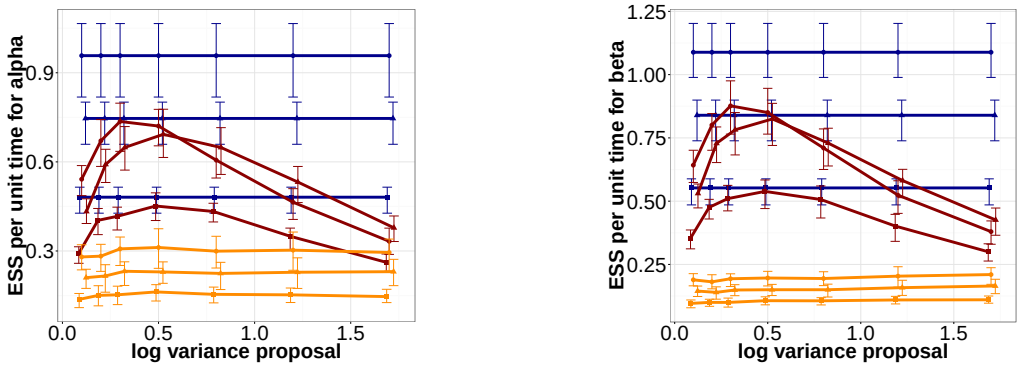


Figure 15: ESS/sec for Immigration model (dim 10)

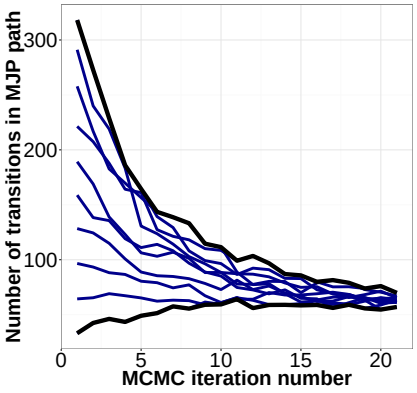


Figure 16: Trace plot of the number of MJP transitions for different initializatoins for immigration model.

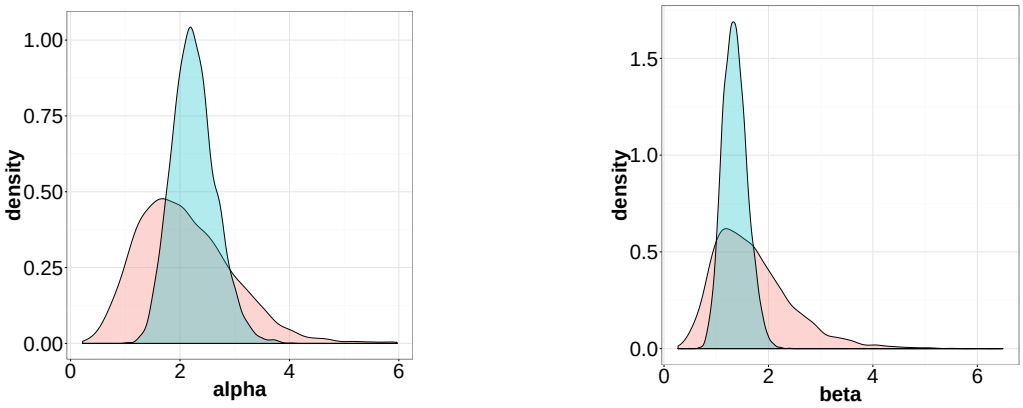


Figure 17: density

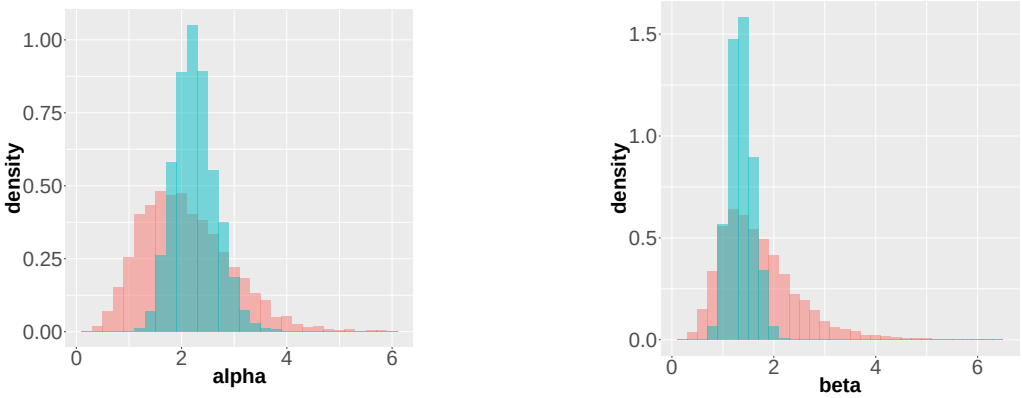


Figure 18: test

# 11 DNA evolution JC69 model

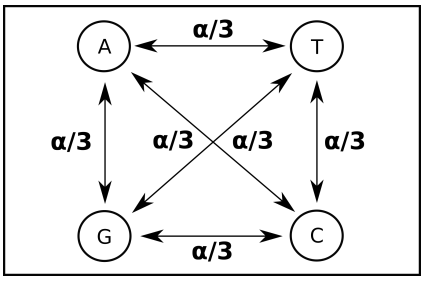


Figure 19: exp model

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 $\alpha$. The overall substitution rate is therefore $3\alpha$. 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 $\alpha$ 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 $Gamma(\mu + N, \lambda + 3(t_{end} - t_{start}))$

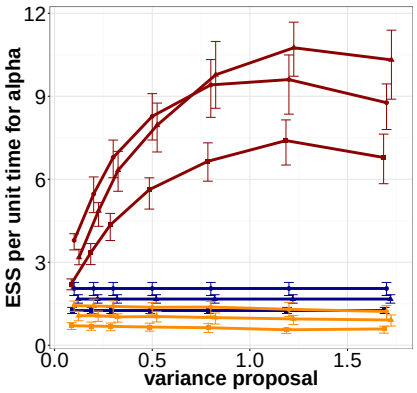


Figure 20: ESS/sec for JC69 Model

# 12 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 $\alpha$, $\beta$ 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.

# 13 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.

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

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

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

$w(t)$ 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 $\tau_i$, 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 $\alpha$, and $\beta$ 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 $\alpha$, $\beta$ 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 $\gamma$ priors to $\alpha$ and $\beta$. We apply our Metropolis Hasting algorithms on such models to compare the performance with the performance of Gibbs Sampling algorithm.

## 13.1 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 $\alpha$, $\beta$ 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

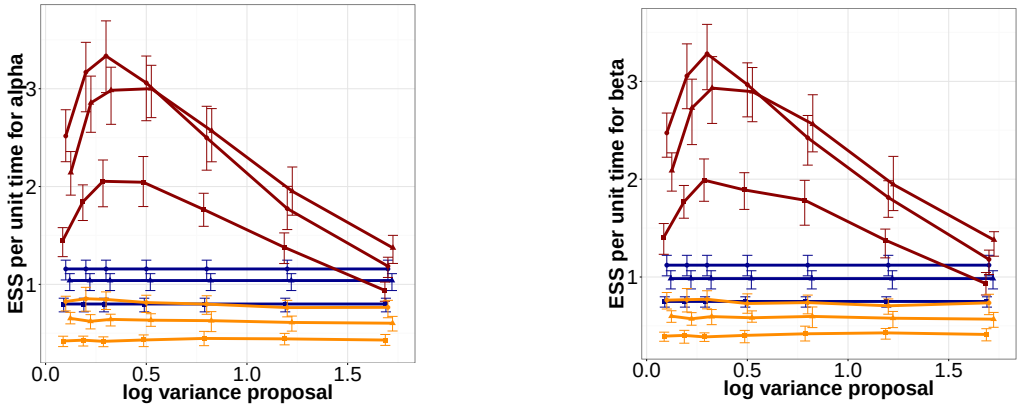


Figure 21: ESS/sec for NH Immigration model (dim 3)

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.

? ? ? ? ? ? ? ? ?

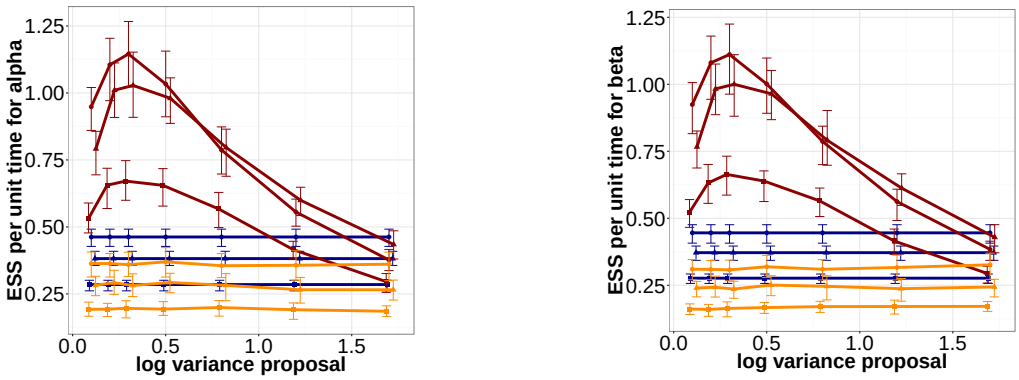


Figure 22: ESS/sec for NH Immigration model (dim 5)

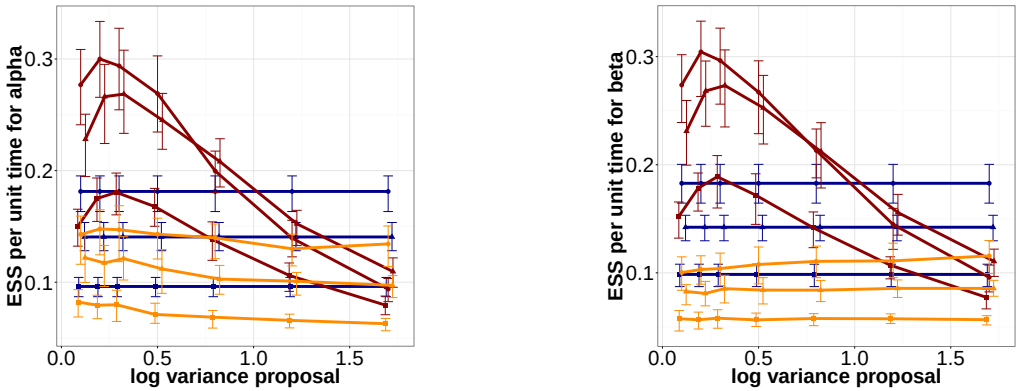


Figure 23: ESS/sec for NH Immigration model (dim 10)