EFFICIENT PATH AND PARAMETER INFERENCE FOR MARKOV JUMP PROCESSES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Boqian Zhang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Vinayak Rao, Chair

Assistant Professor of the Department of Statistics, Purdue University

Dr. Kiseop Lee

Associate Professor of the Department of Statistics, Purdue University

Dr. Raghu Pasupathy

Associate Professor of the Department of Statistics, Purdue University

Dr. Hao Zhang

Professor of the Department of Statistics, Purdue University

Approved by:

Dr. Jun Xie

Graduate Chair of the Department of Statistics, Purdue University

For my parents Zhen Zhang and Chao Yang.

ACKNOWLEDGMENTS

I have been very fortunate to be advised by my advisor Dr. Vinayak Rao during my Ph.D. career. First and foremost, I would like to thank Dr. Vinayak Rao for his help and support throughout my Ph.D. research studies. He is knowledgeable and patient. His ideas and feedback were always helpful and inspiring. His valuable advice and guidance helped me to become a qualified independent researcher, which will also have a profound influence on my future career.

I would also like to thank my thesis committee members, Dr. Kiseop Lee, Dr. Raghu Pasupathy and Dr. Hao Zhang for their precious advice to my research work.

Besides, I want to express my gratitude to Dr. Jun Xie, for her kindly advice and help in the last five years. And I really appreciate the faculty and staff in the department of statistics, for their great work and help to me.

I would like to thank everyone in Dr. Rao's weekly reading group. Thank you for presenting so many great papers. I will miss the time when we had those inspiring discussions in order to understand the papers better.

I would like to thank my dear friends I made at Purdue, especially, Cheng Li, Zhou Shen, Qi Wang, Jingyuan Chen, Zizhuang Wu, Jiasen Yang, Jincheng Bai, Botao Hao, Hanxi Sun, Jiapeng Liu, Putu Ayu Sudyanti, Eric Gerber, Ryan Murphy, Donglai Chen. I appreciate all the help you gave to me and I will never forget the time we spent together. It is you who made my time at Purdue full of joy.

I thank the Purdue Research Foundation and the National Science Foundation for the financial support.

Last but not least, I would like to thank my parents Zhen Zhang and Chao Yang for their love and support.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

CLT	Central limit theorem
CVB	Collapsed variational Bayesian
ELBO	Evidence lower bound
ESS	Effective sample size
FFBS	Forward filtering backward sampling
JC69	Jukes-Cantor
KL	Kullback-Leibler
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo
MH	Metropolis Hastings
MJP	Markov jump process
MWG	Metropolis-within-Gibbs
VB	Variational Bayes
PMCMC	Particle MCMC
TV	Total variation

ABSTRACT

Zhang, Boqian PhD, Purdue University, May 2019. Efficient Path and Parameter Inference for Markov Jump Processes. Major Professor: Vinayak Rao.

Markov jump processes are continuous-time stochastic processes widely used in a variety of applied disciplines. Inference typically proceeds via Markov chain Monte Carlo (MCMC), the state-of-the-art being a uniformization-based auxiliary variable Gibbs sampler. This was designed for situations where the process parameters are known, and Bayesian inference over unknown parameters is typically carried out by incorporating it into a larger Gibbs sampler. This strategy of sampling parameters given path, and path given parameters can result in poor Markov chain mixing.

In this thesis, we focus on the problem of path and parameter inference for Markov jump processes.

In the first part of the thesis, a simple and efficient MCMC algorithm is proposed to address the problem of path and parameter inference for Markov jump processes. Our scheme brings Metropolis-Hastings approaches for discrete-time hidden Markov models to the continuous-time setting, resulting in a complete and clean recipe for parameter and path inference in Markov jump processes. In our experiments, we demonstrate superior performance over Gibbs sampling, a more naïve Metropolis-Hastings algorithm we propose, as well as another popular approach, particle Markov chain Monte Carlo. We also show our sampler inherits geometric mixing from an 'ideal' sampler that is computationally much more expensive.

In the second part of the thesis, a novel collapsed variational inference algorithm is proposed. Our variational inference algorithm leverages ideas from discrete-time Markov chains, and exploits a connection between Markov jump processes and discrete-time Markov chains through uniformization. Our algorithm proceeds by

marginalizing out the parameters of the Markov jump process, and then approximating the distribution over the trajectory with a factored distribution over segments of a piecewise-constant function. Unlike MCMC schemes that marginalize out transition times of a piecewise-constant process, our scheme optimizes the discretization of time, resulting in significant computational savings. We apply our ideas to synthetic data as well as a dataset of check-in recordings, where we demonstrate superior performance over state-of-the-art MCMC methods.

1. INTRODUCTION

1.1 Inference for Markov jump processes

Discrete-time Markov chains are one of the most popular models in statistics and machine learning, widely used for modeling sequences from fields such as speech and video processing (Rabiner and Juang, 1986; Rabiner, 1989), genetics (Yoon, 2009) and social-network analysis (Sarkar and Moore, 2006). However, often one is interested in modeling a system whose evolution is asynchronous, with multiple time scales. In such a situation, working directly in continuous time is a more natural approach, since there is no natural discretization time scale. Further, it is sometimes useful to make continuous approximations to discrete-time systems. For example in genetics, base-position along a strand of DNA is sometimes treated as a real number. Continuous-time modeling often results in easier theoretical analysis, and usually arises naturally from the science of the problem. Markov jump processes (MJPs) (Çinlar, 1975) are continuous-time extensions of discrete-time Markov chains and form one of the simplest continuous-time processes. Markov jump processes are continuous-time piecewise constant stochastic processes (Figure 1.1), and are widely used in fields like computational chemistry (Gillespie, 1977), molecular genetics (Fearnhead and Sherlock, 2006), mathematical finance (Elliott and Osakwe, 2006), queuing theory (Breuer, 2003), artificial intelligence (Xu and Shelton, 2010) and social-network analysis (Pan et al., 2016, 2017). MJPs have been used as realistic, mechanistic and interpretable models of a wide variety of phenomena, among others, the references above have used them to model temporal evolution of the state of a chemical reaction or queuing network, segmentation of a strand of DNA, and user activity on social media.

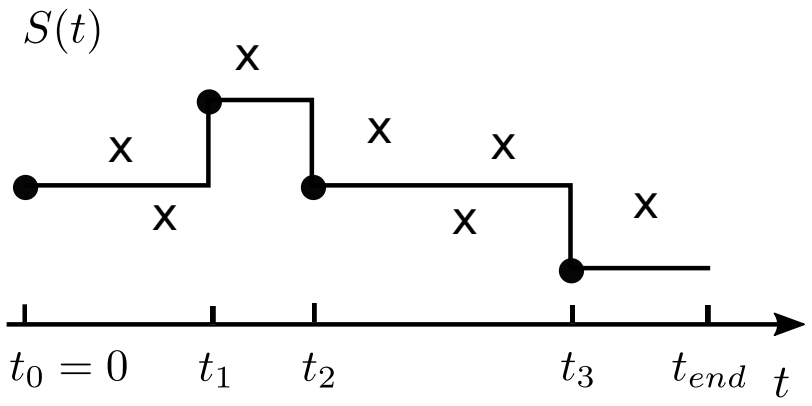

Figure 1.1. An example of an MJP path with noisy observations (crosses). Empty circles are the thinned events.

In the applications mentioned above, the MJP trajectory and the corresponding parameters are usually not observed completely. Instead, one often has noisy observations at a discrete set of times. Figure 1.1 is an example of an MJP trajectory with crosses representing noisy observations. The statistical problem is then to understand the conditional distribution over the MJP trajectory and parameters given these noisy observations. This forms a continuous-time hidden Markov model problem, which is the focus of this thesis. We list a few example applications below.

Queuing theory Computer servers typically process many simultaneous jobs. Only limited number jobs can be processed at a time, and all other jobs must wait in a queue for their turn. An MJP can model such a situation. The state can represent the number of pending jobs in a queue (Asmussen, 2003; Breuer, 2003), with the arrival and processing of jobs treated as independent events. Instead of directly observing the number of jobs, one usually observes the server load or the CPU usage. Based on these noisy observations, one may want to understand the dynamics of the system.

Genetics Genome segmentation is one of the approaches to understand the biological processes, such as mutation and recombination of DNA. An MJP trajectory

can represent a segmentation of a strand of DNA, with different regions corresponding to different mutation rates (Fearnhead and Sherlock, 2006; Philippe et al., 2007). Here, 'Time' actually represents position along a strand of DNA. Again, instead of being directly observed, only the occurrence of a particular DNA motif can be observed. Given these noisy observations, one wants to understand the evolution of the DNA sequence.

Social-network analysis Mobile and social network check-in data (Gao et al., 2012) are being collected with the advancement of sensing technologies. Mobile social media allows users to post their visits to interesting places online. We call such a visit as a check-in record. A check-in record typically consists of a timestamp and a location (latitude and longitude). MJPs can be used to model such check-in data. The state can represent the working state of the social media user. In a simple case, the state space only has two states, 'working' and 'traveling'. Instead of directly observing the states, we observe the check-in records, based on which, the underlying patterns of users can be better understood.

In the discrete-time situation, there exists a variety of computational tools for hidden Markov models. A standard approach for inference characterizes the conditional distribution over path and parameters using Monte Carlo samples. Sampling a trajectory of a finite state discrete-time Markov chain given noisy observations can be done efficiently using the forward-filtering backward-sampling (FFBS) algorithm (Frühwirth-Schnatter, 1994; Carter and Kohn, 1996). The complexity of the FFBS algorithm is $\mathcal{O}(TN^2)$, where T is the length of the chain and N is the number of states in the state space. The continuous-time dynamics of MJP raise computational challenges. In contrast to discrete-time hidden Markov models, one cannot *a priori* bound the number of trajectory state transitions, and the transition times themselves are continuous-valued. As a result, algorithms like FFBS can not be easily applied.

1.2 Our contributions

In this thesis, we consider the problem of path and parameter inference for Markov jump processes. Given noisy observations, we want to make inferences over the latent MJP trajectory as well as any process parameters. We provide new computational tools to address this problem. If the parameters are known, then the inference problem becomes a trajectory inference problem, which has been well studied. The state-of-the-art approach is an auxiliary variable Gibbs sampler from Rao and Teh (2013), we will refer to this as the Rao-Teh algorithm. This Markov chain Monte Carlo (MCMC) algorithm was designed to simulate paths when the MJP parameters are known. In practice, the MJP parameters are unknown. Inference for MJP parameters can be carried out by incorporating it into a Gibbs sampler that also conditionally simulates parameters given the currently sampled trajectory (see section 3.4). This is a straightforward extension of the Rao-Teh algorithm for the path and parameter inference problem. However, in many situations, the Markov jump processes trajectory and parameters exhibit strong coupling, so that alternately sampling path given parameters, and parameters given path can result in poor MCMC mixing.

In order to address this issue, we propose a novel MCMC algorithm (Zhang and Rao, 2018) as well as a novel variational Bayes algorithm (Pan et al., 2017). Both methods are based on the idea of *uniformization* (Jensen, 1953), which is fundamental for our proposed algorithms. This allows us to borrow ideas from discrete-time hidden Markov models to continuous-time hidden Markov models.

In the first part of the thesis, we propose a novel efficient MCMC algorithm for inference for Markov jump processes (Zhang and Rao, 2018). Based on uniformization, our algorithm brings Metropolis-Hastings approaches (Metropolis et al., 1953; Hastings, 1970) from discrete-time hidden Markov models to the continuous-time setting, for parameter and path inference in Markov jump processes. Our algorithm reduces the coupling between the MJP path and parameters by marginalizing out the path information and thereby accelerates the MCMC convergence. We perform empirical

studies in order to demonstrate the superior performance of our proposed MCMC algorithm. We also prove that under relatively mild conditions, our sampler inherits geometric ergodicity from an 'ideal' sampler that is computationally much more expensive. This part of work is included in our paper Zhang and Rao (2018).

In the second part of the thesis, we propose an alternative to MCMC sampling algorithms. We propose a novel variational Bayes (VB) algorithm for inference for Markov jump processes (Pan et al., 2017). Our algorithm marginalizes out the MJP parameters, thereby addressing the issue of slow mixing. Unlike the MCMC algorithm that marginalizes out the MJP transition times, our variational Bayes algorithm optimizes the transition times, resulting in significant computational savings. This part of work is published in our paper Pan et al. (2017).

We organize the rest of the thesis as follows. Chapter 2 provides a review of Markov jump processes and the properties. It introduces the key idea of uniformization (Jensen, 1953). This characterizes a Markov jump process as a discrete Markov chain on a random discretization of time. Given such a representation, we proceed to develop our novel MCMC sampler and our variational Bayes algorithm.

Chapter 3 first gives a brief review of the forward-filtering backward-sampling algorithm for discrete time hidden Markov models. It then describes the state-of-the-art approach for trajectory inference for MJPs when the parameters are known, which is an auxiliary variable Gibbs sampler from (Rao and Teh, 2013). For the case when the parameters are unknown, we introduce a Gibbs sampler which is based on the Rao-Teh algorithm, in order to make inference over both trajectory and parameters. However, the Gibbs sampler can mix very poorly because of coupling between path and parameters.

In chapter 4, we propose a naïve Metropolis-Hastings algorithm (algorithm 6) for Bayesian inference in Markov jump processes, when the parameters are unknown. This approach uses a Metropolis-Hastings scheme to update the MJP parameters, conditioning on a random grid, with the state-values marginalized out. This aims to reduce the path-parameter coupling. However, it still conditions on a random

Poisson grid, whose distribution depends on the MJP parameters. We show that this significantly slows down MCMC mixing.

Chapter 5 describes our first main contribution: a symmetrized Metropolis-Hastings algorithm (algorithm 7) to get around the dependency between the random grid and the MJP parameters. Our main idea is to symmetrize the probability of the random discretization of time under the old and proposed parameters, so that the dependency between the random grid and the MJP parameters disappears when computing the MH acceptance ratio. This improves our earlier proposed naïve MH algorithm (algorithm 6) and significantly accelerates the MCMC mixing.

In Chapter 6, we evaluate Python implementations of a number of algorithms, focusing our contribution, the symmetrized MH algorithm (algorithm 7), and as well as the naïve MH algorithm (algorithm 6). We evaluate different variants of these algorithms, corresponding to different settings. We also evaluate two other baselines: Gibbs sampling (algorithm 4), and particle Markov chain Monte Carlo (Andrieu et al., 2010, see also Appendix). For each run of each MCMC algorithm, we calculated the effective sample size (ESS) of the posterior samples of the MJP parameters using the R package `rcoda` (Plummer et al., 2006). Our experiments demonstrate the superior performance of our symmetrized MH algorithm over Gibbs sampling, the proposed naïve Metropolis-Hastings algorithm (algorithm 6), as well as another popular approach, particle MCMC.

Chapter 7 provides a theoretical analysis of our symmetrized MH algorithm. It starts with a brief review of geometric ergodicity. It then describes our second main contribution, a theorem showing our symmetrized MH sampler inherits geometric mixing from an ideal sampler that is computationally much more expensive under some necessary assumptions.

Chapter 8 gives a review of variational inference, which is a technique to approximate intractable posterior probability densities. It is an alternative to MCMC, and has recently been growing popular in statistics and machine learning (Blei et al., 2017; Wang and Blunsom, 2013; Opper and Sanguinetti, 2007). Chapter 9 shows

our third main contribution. We propose a novel collapsed variational inference algorithm. This work is published in our paper (Pan et al., 2017). Our algorithm exploits a uniformized representation of the Markov jump processes, that views it as Markov chain on a random grid. We describe a prior specification of an MJP using this representation, and by marginalizing out the MJP parameters, avoid some of the parameter-trajectory coupling issues that plague standard MCMC samplers. By maintaining a point estimate of the discretization of time, we improve interpretability and allow our inference algorithm to adaptively determine which times intervals have large transition activity and which are stable.

Finally, we end with a summary, and a discussion of possible future research works in Chapter 10.

2. MARKOV JUMP PROCESSES

2.1 Introduction

In this chapter, we introduce Markov jump processes (MJPs). MJPs are one of the simplest continuous time stochastic processes, widely used in many fields (see section 1.1). In these applications, MJPs serve as a prior distribution over piecewise-constant trajectories. In practice, this trajectory is usually observed with noise through some likelihood function. Together, prior and likelihood define a posterior distribution which summarizes all information about the trajectory. However, for MJPs, this is an intractable quantity. Thus, in order to characterize it, we must use Monte Carlo or Markov chain Monte Carlo methods to draw samples from the posterior distribution. The challenge is to efficiently sample from the posterior distribution over trajectories and the MJP parameters.

We start with a review of Markov jump processes in section 2.2. Then we introduce the idea of *uniformization* in section 2.3. In section 2.4, we introduce the structured rate matrices. Finally, in section 2.5, we set up our Bayesian model for the MJPs with noisy observations.

2.2 Markov jump processes

A Markov jump process (Çinlar, 1975) is a right-continuous piecewise-constant stochastic process $S(t)$ taking values in a state space $\mathcal{S}$. We assume a finite number of states N, with $\mathcal{S} = \{1, \ldots, N\}$. Then, the MJP is parameterized by two quantities, an N-component probability vector π_0 and a rate-matrix A. The former gives the distribution over states at the initial time (we assume this is 0), while the latter is an $N \times N$-matrix governing the dynamics of the system. An off-diagonal element A_{ij}

gives the rate of transitioning from state i to j. The rows of A sum to 0, so that $A_{ii} = -\sum_{j \neq i} A_{ij}$. We write A_i for the negative of the ith diagonal element A_{ii}, so that $A_i = -A_{ii}$ gives the total rate at which the system leaves state i for any other state. We have the following properties for an MJP with rate matrix A. For any $t, t' \geq 0$ and states $s, s' \in \mathcal{S}$, write

$$P(S(t+t') = s | s(t') = s', \{s(u), u < t'\}) = P(S(t+t') = s | s(t') = s') = P^t_{s',s},$$

for some stochastic transition matrix P^t, depending on time t. We have

$$A = \lim_{t \to 0+} \frac{P_t - I}{t},$$

where I is the identity matrix. A is actually the derivative of P^t at $t = 0$.

$$P^t = \exp(At), \quad P(S(t+dt) = s | s(t') = s') = A_{s',s} dt.$$

To simulate an MJP over an interval $[0, t_{end})$, one follows Gillespie's algorithm (Gillespie, 1977): first sample an initial state s_0 from π_0, and defining $t_0 = t_{curr} = 0$ and $k = 0$, repeat the following while $t_{curr} < t_{end}$:

- Sample a wait-time Δt_k from an exponential distribution with rate A_{s_k}. Set $t_{k+1} = t_{curr} = t_k + \Delta t_k$. The MJP remains in state s_k until time t_{k+1}.
- Jump to a new state $s_{k+1} \neq s_k$ with probability equal to $A_{s_k s_{k+1}} / A_{s_k}$. Set $k = k + 1$.

The times $T = (t_1, \ldots, t_{k-1})$ and states $S = (s_1, \ldots, s_{k-1})$, along with the initial state s_0, define the MJP path, so that $\{S(t), t \in [0, t_{end})\} \equiv (s_0, S, T)$. See Figure 2.2 for example.

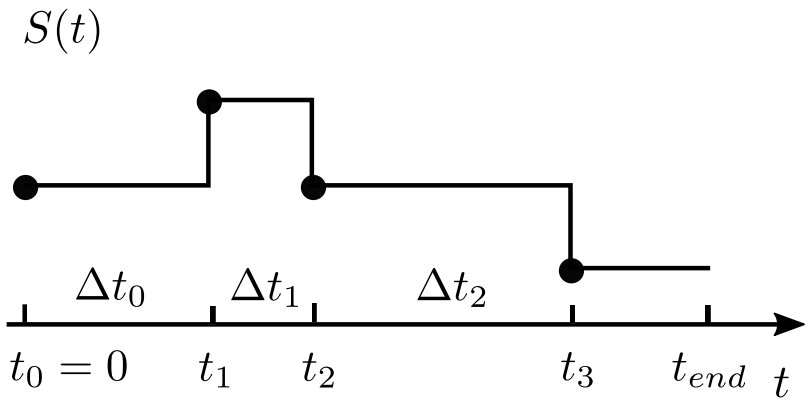

Figure 2.1. Gillespie's algorithm to sample a MJP path on $[0, t_{end}]$.

Algorithm 1 Gillespie's algorithm to sample an MJP path on the interval $[0, t_{end}]$

Input: The rate matrix A and the initial distribution over states π_0.

Output: An MJP trajectory $\{S(t), t \in [0, t_{end})\} \equiv (s_0, S, T)$.

1: Initialize the MJP starting state $s_0 \sim \pi_0$. Set $t_0 = t_{curr} = 0$ and $k = 0$

2: **while** $t_{curr} < t_{end}$ **do**

3: Sample $\Delta t_k \sim \exp(A_{s_k})$.

4: Set $t_{k+1} = t_{curr} = t_k + \Delta t_k$.

5: The MJP jump to a new state $s_{k+1} \neq s_k$ with probability equal to $A_{s_k s_{k+1}}/As_k$.

6: Set $k = k + 1$.

7: **end while**

8: Drop the last pair of (s_k, t_k).

2.3 Uniformization

Gillespie's algorithm is a straightforward way to sample a path from an MJP. In this section, we introduce the idea of *uniformization* (Jensen, 1953), which is an alternative scheme to sample an MJP trajectory. Uniformization builds a connection between the Markov jump processes, the Poisson process and the discrete-time Markov chain. For an Markov jump process with rate matrix A and initial distribu-

tion π_0, choose an $\Omega > \max|A_{ii}|$ and sample a set of times from a Poisson process with intensity Ω in the time interval $[0, t_{end}]$ (Figure 2.2 left). These form a random discretization of time of the time interval $[0, t_{end}]$, and denote it as $W = (w_1, ..., w_{|W|})$, where $0 < w_1 < ... < w_{|W|} < t_{end}$ with probability 1. Let $B = I + \frac{1}{\Omega}A$. B is a valid transition matrix with nonnegative elements and each row summing to 1. It allows the discrete-time system to move back to the same state, which is impossible for the original Markov jump process. Then we run a discrete-time Markov chain with initial distribution π_0 and transition matrix B on W. The Markov chain has state v_0 at time 0 and states $V = (v_1, ..., v_{|W|})$ at times $(w_1, ..., w_{|W|})$, with the dynamics $v_0 \sim \pi_0$ and $P(v_{i+1} = s'|v_i = s) = B_{ss'}$. As shown in Figure 2.2 right, (v_0, V) have self-transitions. After discarding the self-transions, the resulting distribution of trajectories is identical to the original Markov jump process with rate matrix A for any $\Omega > \max|A_{ii}|$ and initial distribution π_0. As we know, the thinning theorem from (Lewis and Shedler, 1979) ensures that a Poisson process sample can be constructed by independently deleting events with probability from a Poisson process with higher rate. The Markov property of the MJP indicates that the independent thinning scheme does not apply any more. The Markov structure implies that if a point w_i should be discarded depends on the MJP state at the previous Poisson time w_{i-1}. Running a discrete-time Markov chain on the set of times W actually provides a mechanism to 'thin' the set W.

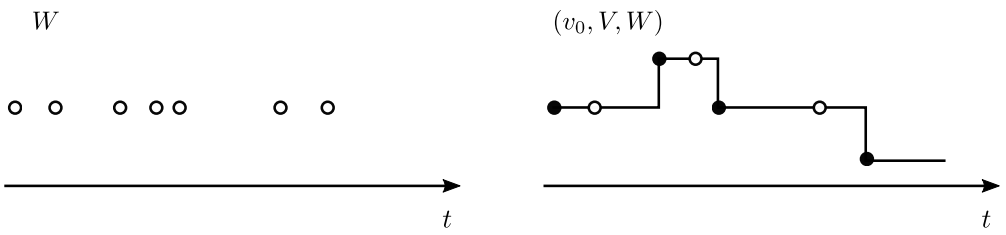

Figure 2.2. (left) Candidate transition times; (right) Uniformization: thin events from a subordinating Poisson process by running a discrete-time Markov chain on this set of times. The empty circles are the thinned events.

Theorem 2.3.1 (Uniformization theorem (Jensen, 1953)) *For any* $\Omega > \max |A_{ii}|$, *and the initial distribution* π_0, (v_0, V, W) *define the Markov jump process with rate matrix* A *and the initial distribution* π_0.

Proof We repeat the proof in Rao and Teh (2013), following the idea in Hobolth and Stone (2009). Let π_t be the marginal distribution of the Markov jump process at time t. We have

$$\begin{aligned}\pi_t &= \exp(At)\pi_0 = \exp(\Omega(B-I)t)\pi_0 \\ &= \exp(-\Omega t)\exp(\Omega t B)\pi_0 \\ &= \sum_{n=0}^{+\infty}\left[\exp(-\Omega t)\frac{(\Omega t)^n}{n!}\right]B^n\pi_0.\end{aligned}$$

The first term in the summation is the probability that a Poisson process with rate Ω has n events in the length t time interval. The second term is the marginal distribution over states after n step for a discrete-time Markov chain with transition matrix B and initial distribution π_0. Since the marginal distribution of the MJP at time t matches the marginal distribution induced by the uniformization procedure, they both define the same Markov jump process. ■

2.4 Structured rate matrices

The dynamics of MJPs are governed by the rate matrix A. Any off-diagonal element A_{ij} gives the rate of transitioning from state i to j, and the negative of the diagonal element, $-A_{ii}$, gives the total rate of the system leaving state i. While the rate matrix A can have $N(N-1)$ independent elements, in typical applications, especially with large state-spaces, it is determined by a much smaller set of parameters. We will write these as θ, with A a deterministic function of these parameters: $A \equiv A(\theta)$. The parameters θ are often more interpretable than the elements of A,and correspond directly to physical, biological or environmental parameters of interest. For example:

Immigration-death processes Here, $\theta = (\alpha, \beta)$, with α the arrival-rate and β the death-rate. The state represents the size of a population or queue. New individuals enter with rate α, so off-diagonal elements $A_{i,i+1}$ equal α. Each individual dies at a rate β, so that $A_{i,i-1} = i\beta$. All other transitions have rate 0.

Birth-death processes This variant of the earlier MJP moves from state i to $i+1$ with rate $i\alpha$, with growth-rate is proportional to population size. Again, the death-rate is β, so that $A_{i,i-1} = i\beta$. The other off-diagonal elements are 0, and again $\theta = (\alpha, \beta)$.

Codon substitution models These characterize transitions between codons at a DNA locus over evolutionary time. There are 61 codons, and in the simplest case, all transitions have the same rate (Jukes and Cantor, 1969): $A_{ij} = \alpha \; \forall i \neq j$. Thus the 61×61 matrix A is determined by a single α. Other models group transitions as 'synonymous' and 'nonsynonymous', based on whether old and new codons encode the same amino acid. Synonymous and nonsynonymous transitions have their own rates, so A is determined by 2 parameters α and β. More refined models (Goldman and Yang, 1994) introduce additional structure and parameters.

2.5 A Bayesian model

We first set up our Bayesian model of the data generation process. We model a latent piecewise-constant path $S(t)$ over $[0, t_{end})$ as an N-state MJP with rate matrix $A(\theta)$ and prior π_0 over $s_0 = S(0)$, the state at time 0. We use both $\{S(t), t \in [0, t_{end})\}$ and (s_0, S, T) (see section 2.2) to refer to the MJP path. We place a prior $P(\theta)$ over the unknown θ. For simplicity, we assume π_0 is known (or we set it to a uniform distribution over the N states). We have noisy measurements X of the latent process, with likelihood $P(X|\{S(t),\ t \in [0, t_{end})\})$. Again, for clarity we ignore any unknown parameters in the likelihood, else we can include them in θ. We assume

the observation process has the following structure: for fixed X, for any partition $\tilde{W} = \{\tilde{w}_1 = 0, \ldots, \tilde{w}_{|\tilde{W}|} = t_{end}\}$ of the interval $[0, t_{end})$ (where $|\cdot|$ denotes cardinality), there exist known functions ℓ_i such that the likelihood factors as:

$$P(X|\{S(t),\ t \in [0, t_{end})\}) = \prod_{i=1}^{|\tilde{W}|-1} \ell_i(\{S(t),\ t \in [\tilde{w}_i, \tilde{w}_{i+1})\}) \tag{2.1}$$

A common example is a finite set of independent observations $X = \{x_1, \ldots, x_{|X|}\}$ at times $T^X = \{t_1^X, \ldots, t_{|X|}^X\}$, each observation depending on the state of the MJP at that time:

$$P(X|\{S(t),\ t \in [0, t_{end})\}) = \prod_{i=1}^{|X|} P(x_i|S(t_i^X)). \tag{2.2}$$

Other examples include situations when the observations form an inhomogeneous Poisson process (Fearnhead and Sherlock, 2006), renewal process (Rao and Teh, 2011) or even another MJP (Nodelman et al., 2002; Rao and Teh, 2013), modulated by (s_0, S, T). The first example, called a Markov modulated Poisson process (MMPP) (Scott and Smyth, 2003), associates a positive rate λ_s with each state s, with $\ell_i(\{S(t),\ t \in [w_i, w_{i+1})\})$ equal to the likelihood of the Poisson events within $[w_i, w_{i+1})$ under an inhomogeneous Poisson process with piecewise-constant rate $\lambda_{S(t)},\ t \in [w_i, w_{i+1})$.

With $A(\cdot)$ and π_0 assumed known, the overall Bayesian model is then

$$\theta \sim P(\theta), \quad (s_0, S, T) \sim \mathrm{MJP}(\pi_0, A(\theta)), \quad X \sim P(X|s_0, S, T). \tag{2.3}$$

Given X, one is interested in the posterior distribution over the latent quantities, (θ, s_0, S, T).

3. THE STATE-OF-THE-ART MCMC METHODS FOR MARKOV JUMP PROCESSES

3.1 Introduction

If the MJP parameters θ in the Bayesian model in section 2.5 are known, we have the problem of trajectory inference. One wants to understand the conditional distribution over the latent MJP path given the noisy observations, $P(s_0, S, T|X, \theta)$ of the Bayesian model of equation (2.3). The problem of trajectory inference was addressed in Rao and Teh (2013) and extended to a broader class of jump processes in Rao and Teh (2012) (also see Fearnhead and Sherlock, 2006; Hobolth and Stone, 2009; El-Hay et al., 2008). Rao and Teh (2013, 2012) both involve MJP path representations with auxiliary *candidate* jump times that are later *thinned*. However, in practice, the parameters are typically unknown, and often, the conditional distribution over the parameters $P(\theta|X)$ is of primary interest. This is also intractable. One then has to characterize the complete posterior $P(\theta, s_0, S, T|X)$ of the Bayesian model of equation (2.3). Some approaches involving particle MCMC (Andrieu et al., 2010) or matrix exponentials (Fearnhead and Sherlock, 2006) have been proposed.

In this chapter, we introduce the Rao-Teh algorithm (Rao and Teh, 2013), which is the state-of-the-art approach for trajectory inference for MJPs. It is an efficient auxiliary variable Gibbs sampler, based on the idea of uniformization described in section 2.3, which is designed for simulating MJP trajectories when the parameters are known. Before diving into the Rao-Teh algorithm, we first introduce a classic sampling algorithm called forward-filtering backward-sampling (FFBS) algorithm for discrete-time Markov chains, which plays an important role in the Rao-Teh algorithm. Further, we introduce a Gibbs sampler based on the Rao-Teh algorithm to tackle the

problem of inference over both trajectory and parameters. It is a straightforward extension to the Rao-Teh algorithm.

3.2 The forward-filtering backward-sampling algorithm for discrete-time Markov chains

Developed originally for finite state hidden Markov models, the forward-filtering backward-sampling algorithm is a dynamic programming algorithm to efficiently simulate latent Markov chain given noisy observations. The algorithm makes a forward pass through time, recursively accounting for successive observations. Then, it samples a trajectory via a backward pass. The earliest references we know for the FFBS algorithm are Frühwirth-Schnatter (1994) and Carter and Kohn (1996). Let S_t, $t \in \{0, 1, ..., T\}$ be a discrete-time Markov chain with a discrete state space $\mathcal{S} = \{1, 2, 3, ..., N\}$. The transition probability of the Markov chain is $P(S_{t+1} = s'|S_t = s) = B_{s,s'}$, where B is the transition matrix, for any $t \in \{0, 1, ..., T-1\}$. π_0 is the initial distribution over states at time $t = 0$. Let X_t be a noisy observation on the state at time t, with the likelihood known as $\ell_t(s) = P(X_t|S_t = s)$. Given a set of observations $X = (X_0, X_1, ..., X_T)$, the corresponding joint distribution is

$$\begin{aligned} P(S_0, S_1, \cdots S_T, X) &= P(S_0, S_1, \cdots S_T) P(X|S_0, S_1, \cdots S_T) \\ &= \left[P(S_0) \prod_{i=0}^{T-1} P(S_{i+1}|S_i) \right] \left[\prod_{i=0}^{T} P(X_i|S_i) \right] \\ &= \pi_0(S_0) \prod_{i=0}^{T-1} B_{S_i,S_{i+1}} \prod_{i=0}^{T} P(X_i|S_i). \end{aligned}$$

The FFBS algorithm returns independent posterior samples of the state vector from the posterior distribution $P(S_0, S_1, \cdots S_T|X)$, given the transition matrix B and the initial distribution π_0.

Define $\mathtt{f}_t(s) = P(S_t = s, X_0, ..., X_{t-1})$ for any $t \in \{0, 1, ..., T\}$ and $s \in \mathcal{S}$. $\mathtt{f}_t(s)$ can be computed recursively from the Markov property.

$$\begin{aligned}\mathtt{f}_t(s') &= \sum_{s=1}^{N} \mathtt{f}_{t-1}(s) P(X_{t-1}|S_{t-1} = s) P(S_t = s'|S_{t-1} = s) \\ &= \sum_{s=0}^{N} \mathtt{f}_{t-1}(s) \ell_{t-1}(s) B_{s,s'}.\end{aligned}$$

At each step t, it takes $\mathcal{O}(N^2)$ calculations to compute $\mathtt{f}_t(s')$ for all $s \in \mathcal{S}$, and a forward pass through all T times takes $\mathcal{O}(TN^2)$. At the end of the forward algorithm, let $\mathtt{b}_T(s)$ be the following.

$$\begin{aligned}\mathtt{b}_T(s) = \ell_T(s)\mathtt{f}_T(s) &= P(X, S_T = s) \\ &\propto P(S_T = s|X).\end{aligned}$$

Then given S_{t+1}, define $\mathtt{b}_t(s)$ as follows recursively.

$$\mathtt{b}_t(s) = P(S_t = s|S_{t+1}, X)) \propto P(S_t = s, S_{t+1}, X)) \tag{3.1}$$

$$= \mathtt{f}_t(s) B^t_{sS_{t+1}} \ell_t(s) P(X^{t+1}, ..., X^T|S_{t+1}) \tag{3.2}$$

$$\propto \mathtt{f}_t(s) B^t_{sS_{t+1}} \ell_t(s). \tag{3.3}$$

First sample a realization of S_T from $\mathtt{b}_T$. Then, based on 3.3, FFBS sequentially samples S_{T-1}, S_{T-2}, ... , S_0. Also, as a byproduct, at the end of the forward pass, the marginal probability of the observations $P(X_0, X_1, ..., X_T)$ can be computed easily by

$$P(X_0, X_1, ..., X_T) = \sum_{s=1}^{N} \mathtt{f}_T(s) \ell_T(s). \tag{3.4}$$

Algorithm 2 includes the details of this algorithm.

Algorithm 2 The forward-filtering backward-sampling algorithm

Input: An initial distribution over states π_0, observations $X = (X_0, ..., X_T)$, with the likelihood $\ell_t(s) = p(X_t|S_t = s)$, transition matrix$B$.

Output: A realization of the Markov chain $(S_0, ..., S_T)$.

1: Run a forward to compute all the $\mathtt{f}_t(s)$ for all $t = 0 \to T$, $s \in \mathcal{S}$:

$$\mathtt{f}_0(s) = \pi_0(s);$$

$$\mathtt{f}_t(s') = \sum_{s=0}^{N} \mathtt{f}_{t-1}(s)\ell_{t-1}(s)B_{s,s'}.$$

2: Sample S_T:

$$\text{Sample } S_T \sim \mathtt{b}_T(s) = \ell_T(s)\mathtt{f}_T(s);$$

3: Backwardly sample S_t for $t = T - 1 \to 0$:

$$\text{Sample } S_t \sim \mathtt{b}_t(s) \propto \mathtt{f}_t(s)B_{sS_{t+1}}\ell_t(s).$$

4: Compute the marginal probability of the observations:

$$P(X_0, X_1, ..., X_T) = \sum_{s=1}^{N} \mathtt{f}_T(s)\ell_T(s).$$

3.3 Bringing FFBS from discrete-time to continuous-time

The Rao-Teh algorithm is based on the idea of *uniformization* (Jensen, 1953), described in Section 2.3. Uniformization involves a parameter $\Omega(\theta) \geq \max_i A_i(\theta)$; Rao and Teh (2013) suggest $\Omega(\theta) = 2 \max_i A_i(\theta)$. Define $B(\theta) = \left(I + \frac{1}{\Omega(\theta)}A(\theta)\right)$; this is a stochastic matrix with nonnegative elements, and rows adding up to 1. Unlike the sequential wait-and-jump Gillespie algorithm, uniformization first simulates a random grid of candidate transition-times W over $[0, t_{end})$, and then assigns these state values.

Introducing the thinned variables allowed Rao and Teh (2013) to develop an efficient MCMC sampler (algorithm 3). At a high-level, each MCMC iteration simulates

a new grid W conditioned on the path (s_0, S, T), and then a new path conditioned on W. Rao and Teh (2013) show that the resulting Markov chain targets the desired posterior distribution over trajectories, and is ergodic for any $\Omega(\theta)$ strictly greater than all the $A_i(\theta)$'s.

Given an MJP path $\{S(t), t \in [0, t_{end})\}$, the Rao-Teh algorithm proceeds by resampling the thinned events U from a Poisson process with piecewise-constant rate $\Omega(\theta) - A_{S(t)}(\theta), t \in [0, t_{end})$ (Figure 3.1 top right). When a state s has a high rate, the the Poisson rate for the thinned event corresponding to state s is small. Then, the new set of candidate transition times W includes the actual MJP transition times T and the thinned events U. Discard the state information V corresponding to W (Figure 3.1 bottom left). Conditioning on the candidate transition times W, the problem becomes a discrete-time hidden Markov model problem, with transition matrix $B(\theta) = \left(I + \frac{1}{\Omega(\theta)} A(\theta)\right)$. Thus, FFBS can be applied to resample the states corresponding to W (Figure 3.1 bottom right). After dropping the thinned events, we have a new MJP trajectory.

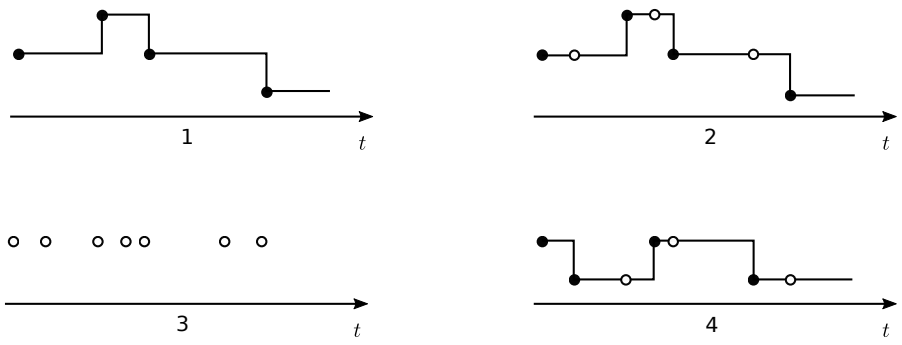

Figure 3.1. The Rao-Teh algorithm: Steps 1-2: Sample the thinned events (empty circles). Step 3: Discard state information to get a random grid. Step 4: Resample the trajectory by running the FFBS algorithm on the grid.

Algorithm 3 The (Rao and Teh, 2013) MCMC sampler for MJP trajectories

Input: Prior π_0, observations X, the previous path $S(t) = (s_0, S, T)$;
Parameter $\Omega(\theta) > \max_i A_i(\theta)$, where $A(\theta)$ is the MJP rate-matrix.

Output: New MJP trajectory $S'(t) = (s'_0, S', T')$.

1: **Simulate the thinned candidate times U given the MJP path** (S, T) from a piecewise-constant Poisson process with rate $\Omega(\theta) - A_{S(t)}(\theta)$:

$$U \sim \text{PoissProc}(\Omega(\theta) - A_{S(t)}(\theta)) \quad \text{(the rate at time } t \text{ is } \Omega(\theta) - A_s(\theta) \text{ if } S(t) = s).$$

2: **Discard the states** (s_0, S)**, and write** $W = T \cup U$.

3: **Simulate states** (v_0, V) **on** $0 \cup W$ **from a discrete-time HMM** with initial distribution over states π_0 and transition matrix $B(\theta) = \left(I + \frac{1}{\Omega(\theta)} A(\theta)\right)$. Following equation (2.1), between two consecutive times $(\tilde{w}_i, \tilde{w}_{i+1})$ in $\tilde{W} \stackrel{\text{def}}{=} 0 \cup W \cup t_{end}$, state s has likelihood $\ell_i(s) \equiv \ell_i(\{S(t) = s,\ t \in [\tilde{w}_i, \tilde{w}_{i+1})\})$. The simulation involves two steps:

Forward pass: Set $\mathtt{f}_0(\cdot) = \pi_0$. Sequentially update $\mathtt{f}_i(\cdot)$ at time $\tilde{w}_i \in \tilde{W}$ given $\mathtt{f}_{i-1}$:

for $i = 1 \to |\tilde{W}|$ **do:** $\quad \mathtt{f}_i(s') = \sum_{s \in \mathcal{S}} \ell_{i-1}(s) \cdot \mathtt{f}_{i-1}(s) \cdot B_{ss'}(\theta), \quad \forall s' \in \mathcal{S}.$

Backward pass: Set $v_{|W|} \sim \mathtt{b}_{|W|}(\cdot)$, where $\mathtt{b}_{|W|}(s) \propto \mathtt{f}_{|W|}(s) \cdot \ell_{|W|}(s) \quad \forall s \in \mathcal{S}$.

for $i = (|W|-1) \to 0$ **do:** $\quad v_i \sim \mathtt{b}_i(\cdot)$, where $\mathtt{b}_i(s) \propto \mathtt{f}_i(s) \cdot B_{s v_{i+1}}(\theta) \cdot \ell_i(s) \quad \forall s \in \mathcal{S}$.

4: **Discard self-transitions**: Set $s'_0 = v_0$. Let T' be the set of times in W when V changes state. Define S' as the corresponding set of state values. Return (s'_0, S', T').

3.4 Gibbs inference over MJP path and parameters

For fixed parameters θ, the efficiency of the Rao-Teh algorithm has been established, both empirically (Rao and Teh, 2013) and theoretically (Miasojedow and w. Niemiro, 2017). In the case when the parameters are unknown, one has to charac-

terize the complete posterior of the Bayesian model of equation (2.3). In this section, we introduce a Gibbs sampler to make inference over both trajectory and parameters. The inference problem is typically carried out by incorporating the previous algorithm into a Gibbs sampler that targets the joint $P(\theta, s_0, S, T|X)$ by conditionally simulating (s_0, S, T) given θ and then θ given (s_0, S, T). However, sampling parameters given path, and path given parameters alternatively can lead to poor Markov chain mixing. Algorithm 4 (see also Rao and Teh, 2013) outlines this:

Algorithm 4 Gibbs sampling for path and parameter inference for MJPs

Input: The current MJP path $S(t) = (s_0, S, T)$, the current MJP parameters θ.

Output: New MJP trajectory $S'(t) = (s_0', S', T')$ and parameters θ'.

1: Simulate a new trajectory from the conditional $P(s_0', S', T'|X, S(t), \theta)$ by algorithm 3.

2: Simulate a new parameter θ' from the conditional $P(\theta'|X, s_0', S', T')$ (see equation (3.5)).

The distribution $P(\theta'|X, s_0', S', T')$ depends on the amount of time τ_i spent in each state i, and the number of transitions c_{ij} between each pair of states i, j:

$$P(\theta'|X, s_0', S', T') \propto P(\theta') \prod_{i \in \mathcal{S}} \exp(-A_i(\theta')\tau_i) \prod_{j \in \mathcal{S}} \left(\frac{A_{ij}(\theta')}{A_i(\theta')} \right)^{c_{ij}}. \tag{3.5}$$

In some circumstances, this can be directly sampled from, otherwise, one has to use a Markov kernel like Metropolis-Hastings to update θ to θ'. In any event, this introduces no new technical challenges. However, the resulting Gibbs sampler can mix very poorly because of coupling between path and parameters. We illustrate this in figure 3.2 (inspired by Papaspiliopoulos et al., 2007)), which shows the posterior distribution of an MJP parameter (long-dashes) is less concentrated than the distribution conditioned on both observations as well as path (short-dashes). The coupling is strengthened as the trajectory grows longer (right panel), and the Gibbs sampler can mix very poorly with long observation periods, even if the observations themselves

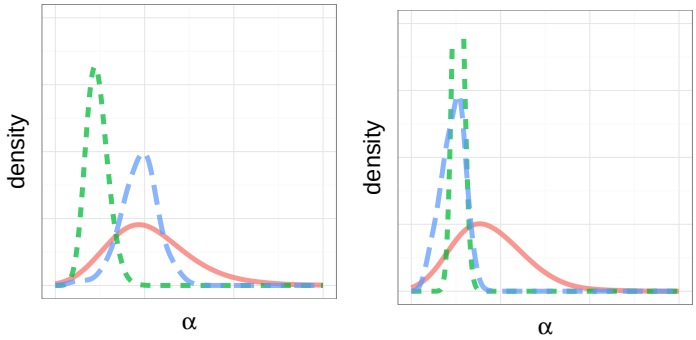

Figure 3.2. Prior density over an MJP parameter (solid curve), along with two conditionals: given observations only (long-dashes), and given observations and MJP path (short-dashes). As t_{end} increases from 10 (left) to 100 (right), the conditionals become more concentrated, implying stronger path-parameter coupling. The plots are from section 6.4 with 3 states.

are only mildly informative about the parameters. In the next chapter, we describe our first naïve attempt in order to get around this coupling issue.

4. NAïVE PARAMETER INFERENCE VIA METROPOLIS-HASTINGS

4.1 Introduction

In this chapter we outline an attempt around the path-parameter coupling we mentioned in section 3.4. We propose a naïve Metropolis-Hastings algorithm (algorithm 6) for Bayesian inference in Markov jump processes, when the parameters are unknown. Before we describe our algorithm, we first introduce the MH algorithm for parameter inference for discrete-time Markov chains, which is the idea we bring to the continuous-time setting.

4.2 Metropolis-Hastings algorithm for discrete-time Markov chains

For discrete-time HMMs, path-parameter coupling can be circumvented by marginalizing out the Markov trajectory, and directly sampling from the marginal posterior $P(\theta|X)$. In its simplest form, this involves a Metropolis-Hastings (MH) scheme that proposes a new parameter ϑ from some proposal distribution $q(\vartheta|\theta)$, accepting or rejecting according to the usual MH probability. The latter step requires calculating the marginal probabilities $P(X|\theta)$ and $P(X|\vartheta)$, integrating out the exponential number of possible latent trajectories. Fortunately, as shown in equation 3.4 and algorithm 2, the marginal probabilities over X given parameters can be computed while running the FFBS algorithm, without additional computational burden. Algorithm 5 shows the details of this algorithm.

Algorithm 5 Metropolis-Hastings parameter inference for a discrete-time Markov chain

Input: Observations X, proposal density $q(\vartheta|\theta)$,current parameters θ

Output: A new Markov chain parameter θ'.

1: Propose a new parameter ϑ from the proposal distribution $q(\vartheta|\theta)$.

2: Run the FFBS algorithm to obtain the marginal likelihood of the observations, $P(X|\vartheta)$.

3: accept ϑ with probability acc $= \min(1, \frac{P(X|\vartheta)P(\vartheta)q(\theta|\vartheta)}{P(X|\theta)P(\theta)q(\vartheta|\theta)})$.

The basic idea of marginalizing out information to accelerate MCMC convergence rates is formalized by the idea of the Bayesian fraction of missing information (Liu, 1994b). In this context, papers such as Papaspiliopoulos et al. (2007); Yu and Meng (2011) have studied MCMC algorithms for hierarchical latent variable models.

4.3 Naïve Metropolis-Hastings algorithm for Markov jump processes

The Rao-Teh algorithm, which recasts posterior simulation for continuous-time models as discrete-time simulation on a random grid, then provides a simple mechanism to incorporate the MH-scheme for discrete-time HMM we mentioned above into continuous-time settings: directly update θ, conditioning on the random grid W, but marginalizing out the states (v_0, V). In this section, we propose a naïve Metropolis-Hastings algorithm (algorithm 6) for Bayesian inference in Markov jump processes, when the parameters are unknown. We first use uniformization to sample the random discretization (Figure 4.1 step 1 to 3). Then update the parameters, in a Metropolis-Hastings scheme, conditioning on the random grid, with the state-values marginalized out, which aims to reduce the path-parameter coupling (Figure 4.1 step 4). Specifically, given θ and the Poisson grid W, rather than simulating new path values (the backward pass in algorithm 3), and then conditionally updating θ (the

second step in algorithm 4), we *first* propose a parameter ϑ from $q(\vartheta|\theta)$. This is accepted with probability

$$\texttt{acc} = \min\left(1, \frac{P(X|W,\vartheta)P(W|\vartheta)P(\vartheta)q(\theta|\vartheta)}{P(X|W,\theta)P(W|\theta)P(\theta)q(\vartheta|\theta)}\right),$$

thereby targeting the distribution $P(W,\theta|X)$. In the equation above, $P(X|W,\theta)$ is the probability of the observations X given W with (v_0, V) marginalized out. Uniformization says this is the marginal probability of X under a discrete-time HMM on W, with transition matrix $B(\theta)$. This can be computed using the forward pass of FFBS algorithm (steps 4 and 6 of algorithm 6). The term $P(W|\theta)$ is the probability of W under a rate-$\Omega(\theta)$ Poisson process. These, and the corresponding terms for ϑ allow the acceptance probability to be computed. Only *after* accepting or rejecting ϑ do we simulate new states (v_0', V'), using the new parameter θ' in a backward pass over W (Figure 4.1 step 5). The new trajectory and parameter are used to simulate a new grid W', and the process is repeated. Algorithm 6 includes all the details of this algorithm, with figure 4.1 sketching out the main idea.

The resulting MCMC algorithm updates θ with the MJP trajectory integrated out, and by instantiating less 'missing' information, can be expected to mix better. This can be quantified by the so-called Bayesian fraction of missing information (Liu, 1994b; Papaspiliopoulos et al., 2007). The Gibbs sampler of algorithm 4 can be viewed as operating on a centered parametrization (Papaspiliopoulos et al., 2007) or sufficient augmentation (Yu and Meng, 2011) of a hierarchical model involving θ, the Poisson events W, and the state values (v_0, V). The MH algorithm reverses the order in which the path and parameter are updated, and is closely related to noncentered parametrizations or ancillary augmentations. For a detailed review of the suitability of these two approaches, as well as ways to combine them together, we refer to Papaspiliopoulos et al. (2007); Yu and Meng (2011).

We note that even with the state values (v_0, V) marginalized out, θ is updated *conditioned on* W. The distribution of W depends on θ: W follows a rate-$\Omega(\theta)$ Poisson process. This dependence manifests in the $P(W|\theta)$ and $P(W|\vartheta)$ terms in

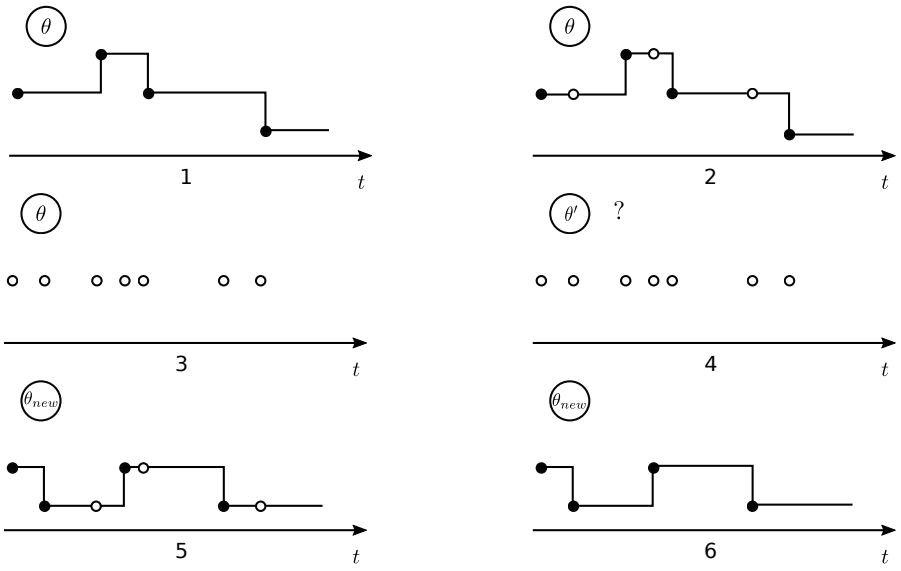

Figure 4.1. Naïve MH-algorithm: Step 1 to 3: sample thinned events and discard state information to get a random grid. Step 4: propose a new parameter θ', and accept or reject by making a forward pass on the grid. Steps 5 to 6: make a backward pass using the accepted parameter and discard self-transitions to produce a new trajectory.

equation (4.1). The fact that the MH-acceptance involves the probability of the observations X is inevitable, however the $P(W|\theta)$ term is an artifact of the computational algorithm of Rao-Teh. In our experiments, we show that this term significantly affects acceptance probabilities and mixing. For parameter θ, $|W|$ is Poisson distributed with mean and variance $\Omega(\theta)$. If the proposed ϑ is such that $\Omega(\vartheta)$ is half $\Omega(\theta)$, then the ratio $P(W|\vartheta)/P(W|\theta)$ will be small, and ϑ is unlikely to be accepted. This will slow down mixing. The next chapter describes our main algorithm that gets around this.

Algorithm 6 Naïve MH for parameter inference for MJPs

Input: Observations X, the MJP path $S(t) = (s_0, S, T)$, the parameters θ and π_0.

Output: A new MJP trajectory $S'(t) = (s'_0, S', T')$, new MJP parameters θ'.

1: Set $\Omega(\theta) > \max_s A_s(\theta)$ for some function $\Omega(\cdot)$, e.g. $\Omega(\theta) = 2\max_s A_s(\theta)$.

2: **Simulate the thinned times** U from a rate-$(\Omega(\theta) - A_{S(t)}(\theta))$ Poisson process:

$$U \sim \text{PoissProc}(\Omega(\theta) - A_{S(t)}(\theta)).$$

3: Set $W = T \cup U$ and discard (s_0, S). Define $\tilde{W} = 0 \cup W \cup t_{end}$.

4: **Forward pass:** Set $B(\theta) = I + \frac{1}{\Omega(\theta)} A(\theta)$ and $\mathtt{f}_0^\theta(\cdot) = \pi_0$.

for $i = 1 \to |\tilde{W}|$ **do:** $\mathtt{f}_i^\theta(s') = \sum_{s \in \mathcal{S}} \ell_{i-1}(s) \cdot \mathtt{f}_{i-1}^\theta(s) \cdot B_{ss'}(\theta), \quad \forall s' \in \mathcal{S}.$

5: **Propose** $\vartheta \sim q(\cdot|\theta)$**.** For all elements of $\tilde{W}$, calculate $\mathtt{f}_i^\vartheta(\cdot)$ similar to above.

6: **Accept/reject:** Set $P(X|W,\theta) = \sum_{s \in \mathcal{S}} \mathtt{f}_{|\tilde{W}|}^\theta(s)$, $P(W|\theta) = \Omega(\theta)^{|W|} \exp(-\Omega(\theta) t_{end})$, with similar expressions for ϑ. With probability $\mathtt{acc}$, set θ' to ϑ, else set it to θ, where:

$$\mathtt{acc} = 1 \wedge \frac{P(\vartheta|W,X)}{P(\theta|W,X)} \frac{q(\theta|\vartheta)}{q(\vartheta|\theta)} = 1 \wedge \frac{P(X|W,\vartheta)P(W|\vartheta)P(\vartheta)}{P(X|W,\theta)P(W|\theta)P(\theta)} \frac{q(\theta|\vartheta)}{q(\vartheta|\theta)}. \tag{4.1}$$

7: **Backward pass:** Set $v_{|W|} \sim \mathtt{b}_{|W|}^{\theta'}(\cdot)$, where $\mathtt{b}_{|W|}^{\theta'}(s) \propto \mathtt{f}_{|W|}^{\theta'}(s) \cdot \ell_{|W|}(s) \quad \forall s \in \mathcal{S}$.

for $i = (|W|-1) \to 0$ **do:** $v_i \sim \mathtt{b}_i^{\theta'}(\cdot)$, where $\mathtt{b}_i^{\theta'}(s) \propto \mathtt{f}_i^{\theta'}(s) \cdot B_{sv_{i+1}}(\theta') \cdot \ell_i(s) \; \forall s \in \mathcal{S}$.

8: Set $s'_0 = v_0$. Let T' be the set of times in W when V changes state. Define S' as the corresponding set of state values. Return (s'_0, S', T', θ').

5. SYMMETRIZED METROPOLIS-HASTINGS ALGORITHM FOR PARAMETER INFERENCE

5.1 Introduction

In this chapter, we describe our first main contribution for the Bayesian inference for the MJPs. We propose a symmetrized Metropolis-Hastings (algorithm 7) to get around the dependency between the random grid and the MJP parameters. Our main idea is to symmetrize the probability of W under the old and proposed parameters, so that $P(W|\theta)$ disappears from the acceptance ratio.

5.2 Symmetrized Metropolis-Hastings algorithm

As before, the MCMC iteration begins with (s_0, S, T, θ). Instead of simulating the thinned events U like earlier algorithms, we *first* generate a new parameter ϑ from some distribution $q(\vartheta|\theta)$ (Figure 7 step 2). Treat this as an auxiliary variable, so that the augmented space now is $(s_0, S, T, \theta, \vartheta)$. Define a function $\Omega(\theta, \vartheta) > \max_s A_s(\theta)$ that is symmetric in its arguments (the number of arguments will distinguish $\Omega(\cdot, \cdot)$ from $\Omega(\cdot)$ of the earlier sections). Two examples are $\Omega(\theta, \vartheta) = \kappa \max_s A_s(\theta) + \kappa \max_s A_s(\vartheta)$, for $\kappa \geq 1$, and $\Omega(\theta, \vartheta) = \kappa \max(\max_s A_s(\theta), \max_s A_s(\vartheta))$, for $\kappa > 1$.

We will treat the path (s_0, S, T) as simulated by uniformization, but now with the dominating Poisson rate equal to $\Omega(\theta, \vartheta)$ instead of $\Omega(\theta)$ as before (Figure 7 step 3). The transition matrix $B(\theta, \vartheta)$ of the embedded Markov chain is $B(\theta, \vartheta) = I + \frac{1}{\Omega(\theta,\vartheta)} A(\theta)$, so that the resulting trajectory (s_0, S, T) will still be a realization from a MJP with rate-matrix $A(\theta)$.

Following the Rao-Teh algorithm, the conditional distribution of the thinned events U given $(s_0, S, T, \theta, \vartheta)$ is a piecewise-constant Poisson with rate $\Omega(\theta, \vartheta) -$

$A_{S(t)}(\theta)$. This reconstructs the set $W = U \cup T$, and as we saw (see also Rao and Teh, 2013), $P(W|\theta, \vartheta)$ is a homogeneous Poisson process with rate $\Omega(\theta, \vartheta)$. Having imputed W, discard the state values, so that the MCMC state space is (W, θ, ϑ). Now, propose swapping θ with ϑ (Figure 7 step 4). From the symmetry of $\Omega(\cdot, \cdot)$, the Poisson grid W has the same probability both before and after this proposal, and unlike the previous scheme, the ratio $P(W|\vartheta)/P(W|\theta)$ equals 1. This simplifies computation, and as suggested in the previous section, can significantly improve mixing. An acceptance probability of $\min\left(1, \frac{P(X|W,\vartheta,\theta)P(\vartheta)q(\theta|\vartheta)}{P(X|W,\theta,\vartheta)P(\theta)q(\vartheta|\theta)}\right)$ targets the conditional $P(W, \theta, \vartheta|X) \propto P(\theta)q(\vartheta|\theta)P(W, X|\theta, \vartheta)$. The terms $P(X|\vartheta)$ and $P(X|\theta)$ can be calculated from the forward pass of FFBS, and after accepting or rejecting the proposal, a new trajectory is sampled by completing the backward pass (Figure 7 step 5). Finally, the thinned events and auxiliary parameter are discarded (Figure 7 step 6). Algorithm 7 and figure 5.1 outline the details of these steps.

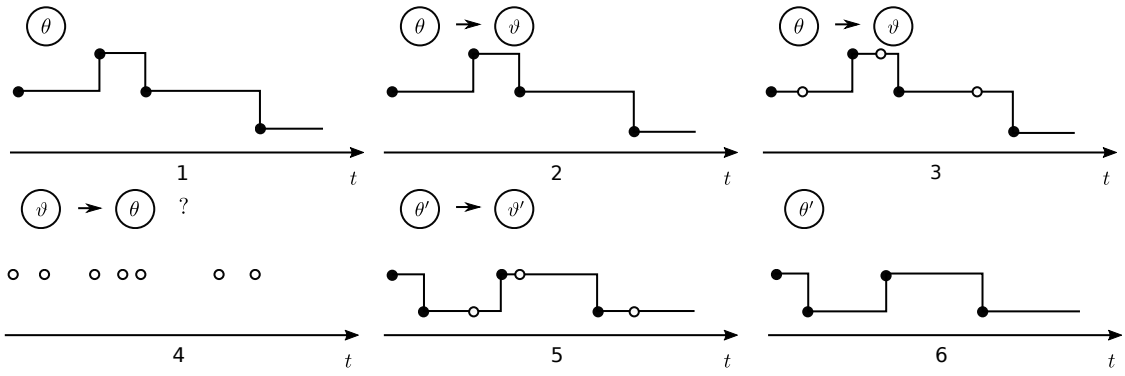

Figure 5.1. Symmetrized MH algorithm: Steps 1-3: Starting with a trajectory and parameter θ, simulate an auxiliary parameter ϑ, and then the thinned events U from a rate $\Omega(\theta, \vartheta) - A_{S(t)}$ Poisson process. Step 4: Discard state values, and propose swapping θ and ϑ. Step 5: Run a forward pass to accept or reject this proposal, calling the new parameters θ', ϑ'). Use these to simulate a new trajectory. Step 6: Discard ϑ' and the thinned events.

Proposition 5.2.1 *The sampler described in Algorithm 7 has the posterior distribution $P(\theta, S(t)|X)$ as its stationary distribution.*

Algorithm 7 Symmetrized MH for parameter inference for MJPs

Input: The observations X, the MJP path $S(t) = (s_0, S, T)$, parameters θ and π_0.

Output: A new MJP trajectory $S'(t) = (s'_0, S', T')$, new MJP parameters θ'.

1: **Sample** $\vartheta \sim q(\cdot|\theta)$, and set $\Omega \doteq \Omega(\theta, \vartheta)$ for some symmetric $\Omega(\theta, \vartheta) > \max_s A_s(\theta)$.

2: **Simulate the thinned times** U from a rate-$(\Omega - A_{S(t)}(\theta))$ Poisson process:

$$U \sim \text{PoissProc}(\Omega - A_{S(t)}(\theta)).$$

3: Set $W = T \cup U$ and discard (s_0, S). Define $\tilde{W} = 0 \cup W \cup t_{end}$.

4: **Forward pass:** Set $B(\theta, \vartheta) = I + \frac{A(\theta)}{\Omega(\theta,\vartheta)}$ and $\mathtt{f}_0^{\theta,\vartheta}(\cdot) = \pi_0$.

for $i = 1 \to |\tilde{W}|$ **do:** $\mathtt{f}_i^{\theta,\vartheta}(s') = \sum_{s \in \mathcal{S}} \ell_{i-1}(s) \cdot \mathtt{f}_{i-1}^{\theta,\vartheta}(s) \cdot B_{ss'}(\theta, \vartheta), \quad \forall s' \in \mathcal{S}.$

Similarly, for $B(\vartheta, \theta) = I + \frac{A(\vartheta)}{\Omega(\vartheta,\theta)}$, calculate $\mathtt{f}_i^{\vartheta,\theta}(\cdot)$ for all elements of $\tilde{W}$.

5: **Swap**: Set $P(X|W, \theta, \vartheta) = \sum_{s \in \mathcal{S}} \mathtt{f}_{|W|}^{\theta,\vartheta}(s)$, and $P(X|W, \vartheta, \theta) = \sum_{s \in \mathcal{S}} \mathtt{f}_{|W|}^{\vartheta,\theta}(s)$. Swap θ and ϑ with probability $1 \wedge \frac{P(X|W,\vartheta,\theta)P(\vartheta)q(\theta|\vartheta)}{P(X|W,\theta,\vartheta)P(\theta)q(\vartheta|\theta)}$. Write the new parameters as (θ', ϑ').

6: **Backward pass:** Set $v_{|W|} \sim \mathtt{b}_{|W|}^{\theta',\vartheta'}(\cdot)$, where $\mathtt{b}_{|W|}^{\theta',\vartheta'}(s) \propto \mathtt{f}_{|W|}^{\theta',\vartheta'}(s) \cdot \ell_{|W|}(s) \quad \forall s \in \mathcal{S}.$

for $i = (|W|-1) \to 0$ **do:** $v_i \sim \mathtt{b}_i^{\theta',\vartheta'}(\cdot)$, where $\mathtt{b}_i^{\theta',\vartheta'}(s) \propto \mathtt{f}_i^{\theta',\vartheta'}(s) \cdot B_{sv_{i+1}}(\theta', \vartheta') \cdot \ell_i(s)\ \forall s \in \mathcal{S}.$

7: Set $s'_0 = v_0$. Let T' be the set of times in W when V changes state. Define S' as the corresponding set of state values. Return (s'_0, S', T', θ').

Proof Consider a realization (s_0, S, T, θ) from the posterior distribution $P(\theta, s_0, S, T|X)$. An iteration of the algorithm first simulates ϑ from $q(\vartheta|\theta)$. By construction, the marginal distribution over all but the last variable in the set $(\theta, s_0, S, T, \vartheta)$ is still the posterior.

The algorithm next simulates U from a Poisson process with rate $\Omega(\theta, \vartheta) - A_{S(t)}(\theta)$. Write $W = T \cup U$. The random grid W consists of the actual and thinned candidate transition times, and is distributed according to a rate-$\Omega(\theta, \vartheta)$ Poisson process (Proposition 2 in (Rao and Teh, 2013)). Thus, the triplet (W, θ, ϑ) has probability proportional to $P(\theta)q(\vartheta|\theta)\text{PoissProc}(W|\Omega(\theta, \vartheta))P(X|W, \theta, \vartheta)$. Next, the algorithm pro-

poses swapping θ and ϑ (a deterministic proposal), and accepts with MH-acceptance probability

$$\texttt{acc} = 1 \wedge \frac{P(\vartheta)q(\theta|\vartheta)P(X|W,\vartheta,\theta)}{P(\theta)q(\vartheta|\theta)P(X|W,\theta,\vartheta)} = 1 \wedge \frac{P(\vartheta)q(\theta|\vartheta)\text{PoissProc}(W|\Omega(\vartheta,\theta))P(X|W,\vartheta,\theta)}{P(\theta)q(\vartheta|\theta)\text{PoissProc}(W|\Omega(\theta,\vartheta))P(X|W,\theta,\vartheta)},$$

where we exploit the symmetry of $\Omega(\cdot,\cdot)$. Write the new parameters as (θ',ϑ').

The Markov kernel has stationary distribution over (θ',ϑ') proportional to $P(\theta')q(\vartheta'|\theta)$ $\text{PoissProc}(W|\Omega(\theta',\vartheta'))P(X|W,\theta',\vartheta')$, and the triplet (θ',ϑ',W) has the same distribution as (θ,ϑ,W). The algorithm uses $B(\theta',\vartheta')$ to make a backward pass through W, simulating state values on W from the conditional of a Markov chain with transition matrix $B(\theta',\vartheta')$ given observations X. Dropping the self-transition times results in $(\theta', s_0', S', T', \vartheta')$. From uniformization (see also Lemma 1 in Rao and Teh (2013)), the trajectory (s_0', S', T') is distributed according to the conditional of a rate-$A(\theta')$ MJP given observations X. Finally, dropping ϑ' results in (θ', s_0', S', T') from the posterior given X. ■

Now, the probability of accepting a proposal ϑ will depend only on the prior probabilities of θ and ϑ, as well as how well they both explain the data given W. This is in contrast to the previous algorithm, where one must also factor in how well each parameter explains the current value of the grid W. This results in an MCMC sampler that mixes significantly more rapidly. Since we also need do account for the probabilities $P(W|\theta)$, we also have a simpler MCMC scheme. This forms one of the main contributions of this thesis.

As mentioned earlier, uniformization forms such a representation for MJPs: first sample θ, then sample the latent W, and use this to sample MJP state values. The naïve algorithm 6 is a direct application of ideas presented in Rao and Teh (2013). An interesting direction is to see how these frameworks can shed light on, and improve our symmetrized MH algorithm 7. Viewed in this light, our contribution is a rewriting of uniformization that includes the auxiliary parameter ϑ. Our swap operator forms a particular Markov kernel that exploits this reparametrization for fast mixing.

5.3 Discussion

Our symmetrized MH algorithm 7 modifies the algorithm from Rao and Teh (2013) to include parameter inference. That algorithm requires a uniformization rate $\Omega(\theta) > \max_s A_s(\theta)$, and empirical results from that paper suggest $\Omega(\theta) = 2\max_s A_s(\theta)$. The uniformization rate $\Omega(\theta, \vartheta)$ in our algorithm includes a proposed new parameter ϑ, must be symmetric in both arguments and must be greater than both $\max_s A_s(\theta)$ and $\max_s A_s(\vartheta)$. A natural and simple setting is $\Omega(\theta, \vartheta) = \max_s A_s(\theta) + \max_s A_s(\vartheta)$. When θ is known, our algorithm has ϑ equal to θ (i.e. the proposed ϑ equals θ), and our uniformization rate reduces to $2\max A_i$. This provides a principled motivation for the particular choice of Ω in Rao and Teh (2013).

Of course, we can consider other choices for the uniformization rate, such as $\Omega(\theta, \vartheta) = \kappa(\max A_i(\theta) + \max A_i(\vartheta))$ for $\kappa > 1$. These result in more thinned events, and so more computation, with the benefit of faster MCMC mixing. We study the effect of κ in our experiments, but find the smallest setting of $\kappa = 1$ performs best. It is also possible to have non-additive settings for $\Omega(\theta, \vartheta)$, for example, $\Omega(\theta, \vartheta) = \kappa \max(\max_i A_i(\theta), \max A_i(\vartheta))$ for some choice of $\kappa > 1$. We investigate this as well.

A key idea in our symmetrized MH algorithm, as well as Rao and Teh (2013), is to impute the random grid of candidate transition times W every MCMC iteration. Conditioned on W, the MJP trajectory follows an HMM with transition matrix B. By running the FFBS algorithm over W, we can marginalize out the states associated with W, and calculate the marginal $P(X|W, \theta)$. There have been some MCMC approaches to posterior inference. Our proposed MCMC algorithm in chapter 5 (algorithm 7) modifies the algorithm from Rao and Teh (2013) to include parameter inference.

Another approach to parameter inference that integrates out state values follows Fearnhead and Sherlock (2006). This algorithm makes a sequential forward pass through all *observations* X (rather than W). Unlike with W fixed, one cannot a priori bound the number of transitions between two successive observations, so that Fearn-

head and Sherlock (2006) have to use matrix exponentials of A (rather than just B) to calculate transition probabilities.

The resulting algorithm is cubic, rather than quadratic in the number of states, and the number of expensive matrix exponentiations needed scales with the number of observations, rather than the number of transitions. Further, matrix exponentiation results in a dense matrix, so that Fearnhead and Sherlock (2006) cannot exploit sparsity in the transition matrix. In our framework, we will use $B = I + \frac{1}{\Omega}A$, which inherits sparsity present in A. Thus if A is tri-diagonal, our algorithm is *linear* in the number of states.

A second approach to marginalizing out state information is particle MCMC (Andrieu et al., 2010). This algorithm, described in section A, uses particle filtering to get an unbiased estimate of $P(X|\theta)$. Plugging this estimate into the MH acceptance probability results in an MCMC sampler that targets the correct posterior, however the resulting scheme does not exploit the Markovian structure of the MJP the way FFBS can. In particular, observations that are informative of the MJP state can result in marginal probability estimates that have large variance, resulting in slow mixing. By contrast, given W, FFBS can compute the marginal probability $P(X|W,\theta)$ *exactly*.

Two interesting directions are to see how such symmetrization ideas apply to other problems considered in those works, and how ideas from those works can shed more light on, and improve our algorithm.

Our approach of first simulating ϑ, and then simulating W from a Poisson process whose rate is symmetric in θ and ϑ is related to Neal (2004). In that work, to simulate from an 'energy' model $P(x,y) \propto \exp(-E(x,y))$, the author proposes a new parameter x^*, and then updates y via intermediate transitions to be symmetric in x and x^*, before proposing to swap x and x^*. Our approach exploits the specific structure of the Poisson and Markov jump processes to do this directly, avoiding the need for any tempered transitions.

Our algorithm 7 is also related to work on MCMC for doubly-intractable distributions. Algorithms like (Møller et al., 2006; Murray et al., 2006; Andrieu and Roberts,

2009) all attempt to evaluate an intractable likelihood under a proposed parameter ϑ by introducing auxiliary variables, typically sampled independently under the proposed parameters. For MJPs, this would involve proposing ϑ, generating a new grid W^*, and then using $P(X|W,\theta)$ and $P(X|W^*,\vartheta)$ in the MH acceptance step. This is more involved (with two sets of grids), and introduces additional variance that reduces acceptance rates. if the new parameter ϑ is incompatible with the old grid U or vice versa. While Murray et al. (2006) suggests annealing schemes to try to address this issue, we exploit the uniformization structure to provide a cleaner solution: generate a single set of auxiliary variables that depends symmetrically on both the new and old parameters.

6. EMPIRICAL SIMULATION RESULTS

6.1 Introduction

In this chapter, we evaluate Python implementations of a number of algorithms, focusing our contribution, the symmetrized MH algorithm (algorithm 7), and as well as the naïve MH algorithm (algorithm 6). We evaluate different variants of these algorithms, corresponding to different uniformizing Poisson rates. For naïve MH, we set $\Omega(\theta) = \kappa \max_s A_s(\theta)$ with κ equal to $1.5, 2$ and 3 (here κ must be greater than 1), while for symmetrized MH, where the uniformizing rate depends on both the current and proposed parameters, we consider $\Omega(\theta, \vartheta) = \kappa(\max A(\theta) + \max A(\vartheta))$ ($\kappa = 1$ and 1.5), and $\Omega(\theta, \vartheta) = 1.5 \max(\max A(\theta), \max A(\vartheta))$. We evaluate two other baselines: Gibbs sampling (algorithm 4), and particle MCMC (Andrieu et al., 2010, see also Appendix). Gibbs sampling involves a uniformization step to update the MJP trajectory (step 1 in algorithm 4), for which we use $\Omega(\theta, \vartheta) = \kappa \max_s A_s(\theta)$ for $\kappa = 1.5, 2, 3$. Unless specified, our results were obtained from 100 independent MCMC runs, each of 10000 iterations. We found particle MCMC to be more computationally intensive, and limited each run to 3000 iterations, the number of particles being $5, 10$ and 20.

For each run of each MCMC algorithm, we calculated the effective sample size (ESS) of the posterior samples of the MJP parameters using the R package `rcoda` (Plummer et al., 2006). This estimates the number of independent samples returned by the MCMC algorithm, and dividing this by the runtime of a simulation gives the ESS per unit time (ESS/sec). We used this to compare different samplers and different parameter settings.

6.2 A simple synthetic MJP

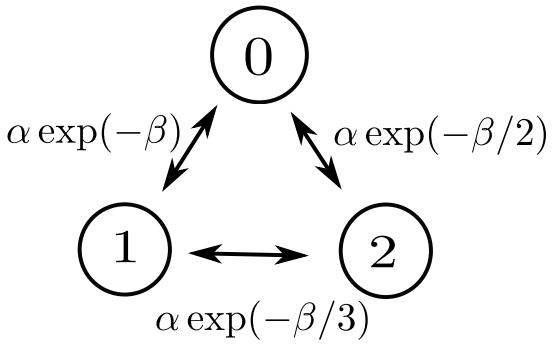

Figure 6.1. A 3-state MJP with exponentially decaying rates

Consider an MJP with a uniform distribution over states at time 0, and with transitions between states i and j having rate $\alpha \exp(-\beta/(i+j))$, for two parameters $(\alpha, \beta) \stackrel{\text{def}}{=} \theta$. We consider three settings: 3 states (figure 6.1), 5 states, and 10 states. We place Gamma(α_0, α_1), and Gamma(β_0, β_1) priors on the parameters α and β, with $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ having values $(3, 2, 5, 2)$ respectively. For each run, we draw random parameters from the prior to construct a transition matrix A, and simulate an MJP trajectory. We simulate observations uniformly at integer values on the time interval $[0, 20]$. Each observation is Gaussian distributed with mean equal to the state at that time, and variance equal to 1. For the MH proposal, we used a lognormal distribution centered at the current parameter value, with variance σ^2 whose effect we study.

Results: Figure 6.2 shows the MCMC estimates of the posterior distribution over $\alpha, P(\alpha|X)$ from the Gibbs sampler as well as our symmetrized MH sampler. Visually these agree, and we quantify this by running a Kolmogorov-Smirnov two-sample test using 1000 samples from each algorithm: this returns a p-value of 0.5085, clearly failing to reject the null hypothesis that both samples come from the same distribution. The figure also shows the average acceptance probabilities for the two MH samplers: we see that for the same proposal distribution, symmetrization significantly improves acceptance probability. This shows the benefit of eliminating the $P(W|\theta)$ terms from

the acceptance probability (we will investigate this further). Figure 6.3 shows trace-plots and autocorrelation plots for α from the symmetrized MH and Gibbs samplers. Clearly, our sampler mixes much more efficiently than Gibbs, with naïve MH slightly worse than both.

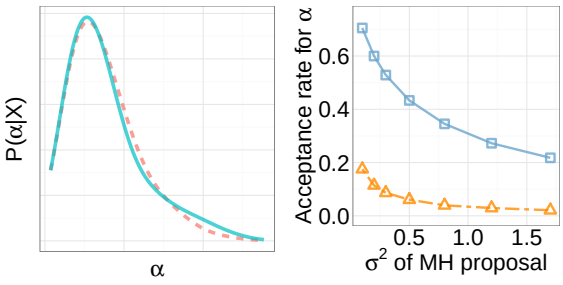

Figure 6.2. (Left) posterior $P(\alpha|X)$ from Gibbs (dashed line) and symmetrized MH (solid line) for the synthetic model. (Right) acceptance probabilities of α for symmetrized (squares) and naïve (triangles) MH.

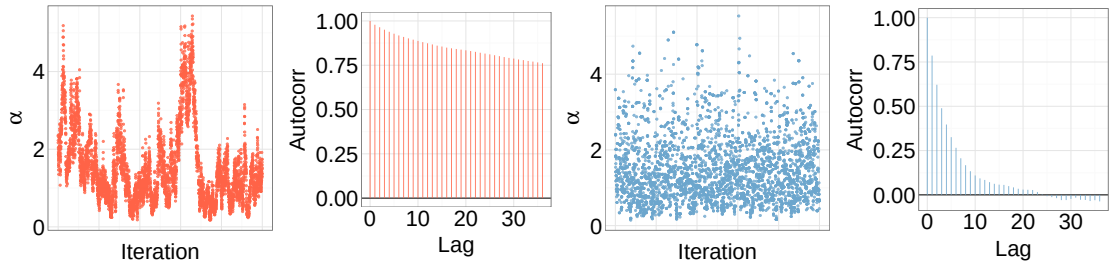

Figure 6.3. Trace and autocorrelation plots for Gibbs (left two panels) and symmetrized MH (right two panels). All plots are for the synthetic odel with 10 states.

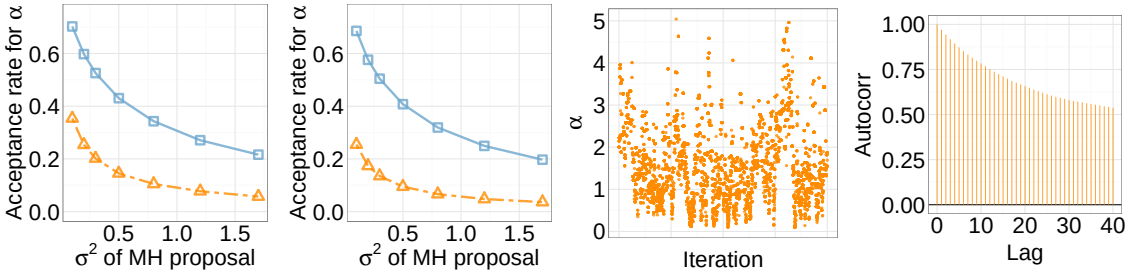

Figure 6.4. Acceptance Rate for α in the synthetic model (Left two), the first being dimension 3, and the second,dimension 5. Blue square and yellow triangle curves are the symmetrized MH, and naïve MH algorithm. The multiplicative factor is 2. Trace and autocorrelation plots for naïve MH (right two panels) for the synthetic model with 3 states.

To quantify this, figure 6.5 plots the ESS/sec in the top row, and the raw ESS in the bottom row for α and β. The left two columns consider α and β for MJPs with 3 states, and the right two, with 10 states. We include results for 5 states later, the conclusions are the same. For each plot, we vary the scale-parameter σ^2 of the log-normal proposal $q(\vartheta|\theta)$, and look at its effects on ESS/s and ESS. Note that the conditional over parameters given trajectory is not conjugate, so that the Gibbs sampler is really a Metropolis-within-Gibbs (MWG) sampler with an associated proposal distribution parameterized by σ^2.

We see that our symmetrized MH algorithm, shown with blue squares, is significantly more efficient than the baselines over a wide range of σ^2 values, including the natural choice of 1. Among the baselines, Gibbs (red circles) does better than naïve MH (yellow triangles), showing that the dependency of the Poisson grid on the MJP parameters (as indicated in figure 6.2) does indeed slow down mixing. This, coupled with the fact that MWG tends to have higher MH acceptance than naïve MH results in Gibbs having superior performance. Our symmetrized MH avoids this problem at no additional computational cost. Particle MCMC (black diamonds) has the worst performance.

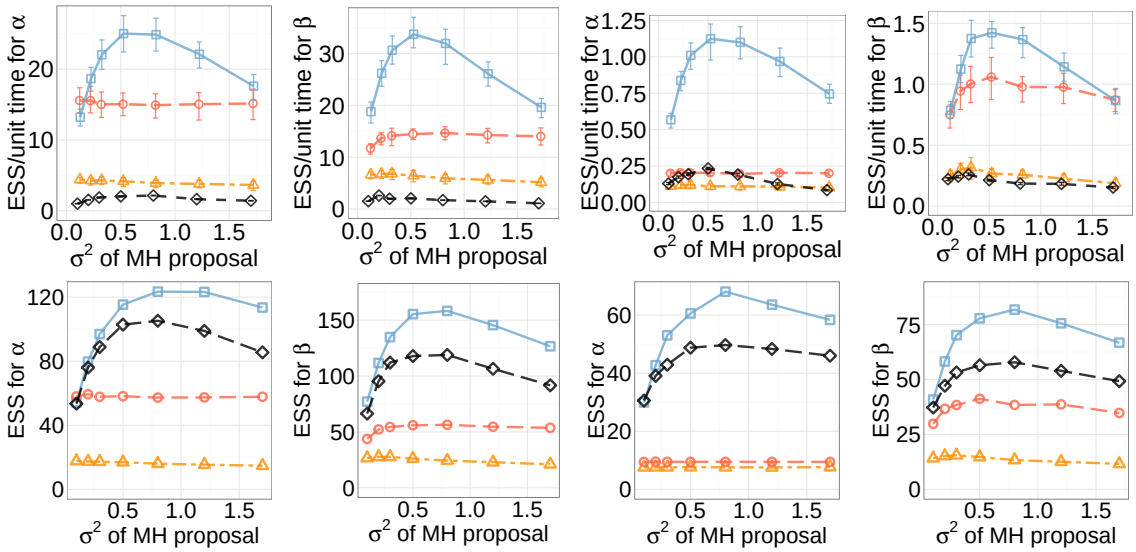

Figure 6.5. ESS/sec (top row) and raw ESS per 1000 samples (bottom row) of different algorithms on the synthetic model. The left two panels are α and β for 3 states, the right two, for 10 states. Blue squares, yellow triangles, red circles and black diamonds are the symmetrized MH, naïve MH, Gibbs and particle MCMC algorithm.

Among the three setting of our algorithm, the simple additive setting (squares) does best, slightly better than the max-of-max setting (circles). The additive setting with a multiplicative factor of 1.5 (triangles) does worse than both the additive choice with $\kappa = 1$ and the max-of-max choice but still better than the other algorithms. The results in figure 6.5 for 10 states shows that ESS is slightly lower, and thus mixing is slightly poorer for all samplers. This, coupled with greater computational cost per iteration results in a drop in ESS/s across all algorithms, compared with 3 states. We still observe the same pattern of relative performance, with our sampler with $\Omega(\theta, \vartheta) = \max_s A_s(\theta) + \max_s A(\vartheta)$ the best. Figure 6.7 shows the results for different settings and different algorithms.

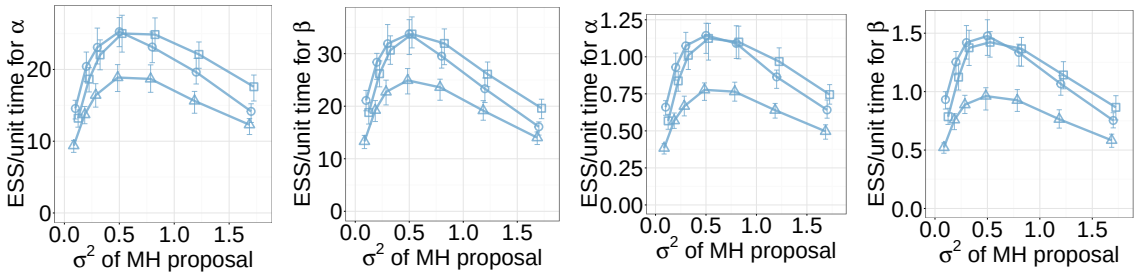

Figure 6.6. ESS/sec of symmetrized MH for different choices of $\Omega(\theta, \vartheta)$ for the synthetic model. The left two panels are α and β for 3 states, and the right two for 10 states. Squares, circles and trianges correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

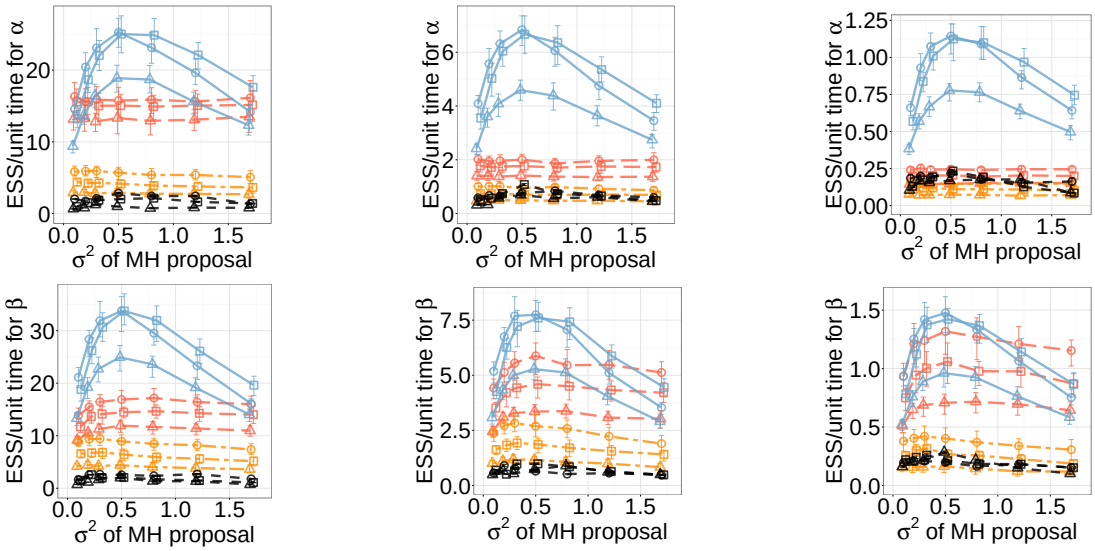

Figure 6.7. ESS/sec for the synthetic model, the top three are for α for 3 states, 5 states, and 10 states. The bottom three are for β for 3 states, 5 states, and 10 states. Blue, yellow, red and black are the symmetrized MH, naïve MH, Gibbs and particle MCMC algorithm. Squares, circles and trianges correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$. And for PMCMC, they correspond to 10 particles, 5 particles and 15 particles.

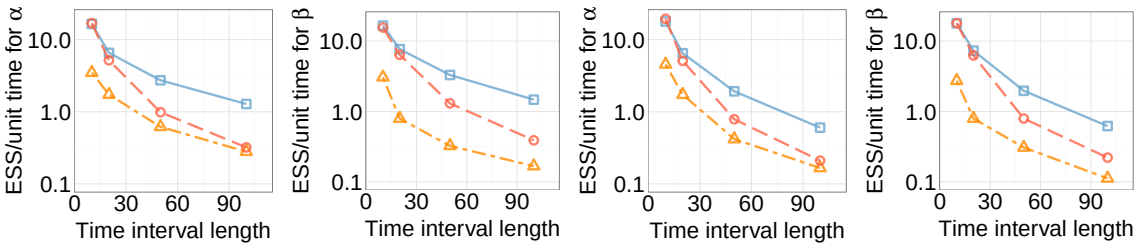

Figure 6.8. Time interval vs ESS/sec for the synthetic MJP. The left two plots are for α and β, with the number of observations fixed; in the right two, this grows linearly with the interval length. Blue squares, yellow triangles and red circles curves are the symmetrized MH, naïve MH and Gibbs algorithm.

In figure 6.8, we plot ESS per unit time as the observation interval t_{end} increases. We consider the 3-state MJP, and as before there are 19 observations uniformly located over a time interval $(0, t_{end})$. We consider four settings, with t_{end} equal to $10, 20, 50, 100$. For each, we compare our symmetrized MH sampler (with κ set to 1) with the naïve MH and Gibbs samplers (with κ set to 2). While the performance of the Gibbs sampler is comparable with our symmetrized algorithm for the smallest value of t_{end}, its performance is considerably worse for longer time-intervals. This is the limitation of Gibbs sampling that motivated this work: when updating θ conditioned on the MJP trajectory, longer time intervals result in stronger coupling between MJP path and parameters (figuree 3.2), and thus poorer mixing. The performance of the naïve sampler demonstrates that it is not sufficient just to integrate out the state values of the trajectory, we also have to get around the coupling between the Poisson grid and the parameters. Our symmetrized MH-algorithm allows this.

To the right of figure 6.8, we plot results from a similar experiment. Now, instead of keeping the number of measurements fixed as we increase the observation interval, we keep the observation *rate* fixed at one observation every unit interval of time, so that longer observation intervals have larger number of observations. The results

are similar to the previous case: Gibbs sampling performs well for small observation intervals, with performance degrading sharply for larger intervals.

6.3 The Jukes and Cantor (JC69) model

The Jukes and Cantor (JC69) model (Jukes and Cantor, 1969) is a popular model of DNA nucleotide substitution. We write its state space as $\{0, 1, 2, 3\}$, representing the four nucleotides $\{A, T, C, G\}$. The model has a single parameter α, representing the rate at which the system transitions between any pair of states. Thus, the rate matrix A is given by $A_i = -A_{i,i} = 3\alpha, A_{i,j} = \alpha, i \neq j$. We place a Gamma$(3, 2)$ prior on the parameter α. Figure 6.10(a), (b) and (c) compare different samplers: we see that the symmetrized MH samplers comprehensively outperforms all others. Part of the reason why the difference is so dramatic here is because now a *single* paramter $\alpha \stackrel{\text{def}}{=} \theta$ defines the transition matrix, implying a stronger coupling between MJP path and parameter. We point out that for Gibbs sampling, the conditional distribution over θ is conjugate to the Gamma prior. We can thus simulate directly from this distribution without any MH proposal (hence its performance remains fixed along the x-axis). Despite this, its performance is worse than our symmetrized algorithm. Particle MCMC performs worse than all the algorithms, and we do not include it in our plots. Figure 6.10(d) compares different settings of $\Omega(\theta, \vartheta)$ for our sampler: again, the simple additive setting $\Omega(\theta, \vartheta) = \max_s A_s(\theta) + \max_s A_s(\vartheta)$ does best.

Figure 6.11 plots MCMC diagnostics for the Gibbs and symmetrized MH sampler, again the latter outperforms the former. Both agree on the posterior $P(\alpha|X)$ (figure 6.12(a)), with a two sample Kolmogorov-Smirnov test giving a p-value of 0.97. Figure 6.12(b) plots the average MH acceptance probabilities for the naïve and symmetrized MH samplers for different settings of the proposal distribution, again we see that the former has lower acceptance rates because of the $P(W|\theta)$ grids.

Figure 6.12 (c) and (d) plot the ESS per unit time for the different samplers as t_{end} increases. The left plot keeps the number of observations fixed, while the right

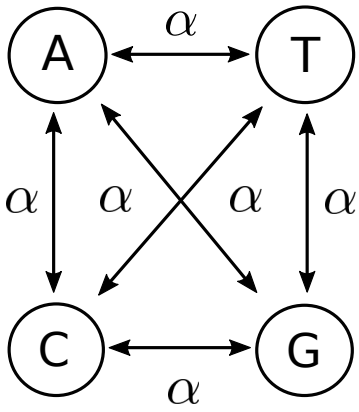

Figure 6.9. Jukes-Cantor (JC69) model.

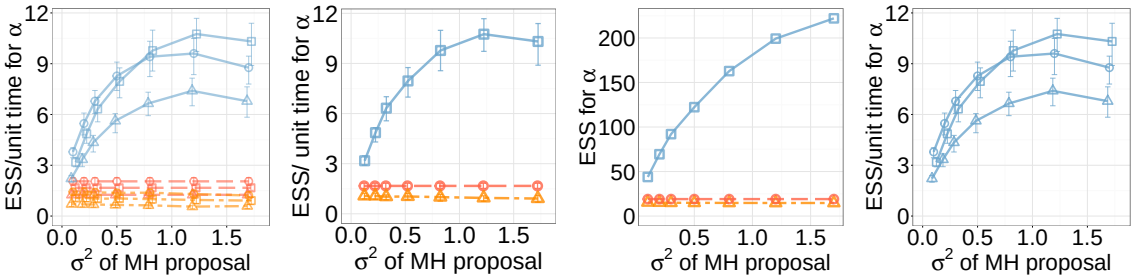

Figure 6.10. ESS/sec for the JC immigration model, blue, yellow and red curves are the symmetrized MH, naïve MH, and Gibbs algorithm. The next two panels from left to right are ESS/sec and raw ESS per 1000 samples for this. Blue squares, yellow triangles and red circles are the symmetrized MH, naïve MH and Gibbs algorithm. The rightmost panel looks at different settings of the symmetrized MH algorithm, with squares, circles and triangles corresponding to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

keeps the observation rate fixed. Once again we see that our proposed algorithm 1) performs best over all interval lengths, and 2) suffers a performance degradation with interval length that is much milder than the other algorithms.

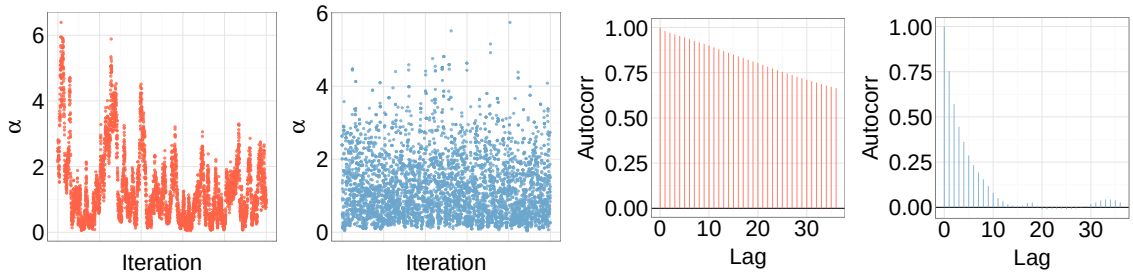

Figure 6.11. Trace (left two) and autocorrelation (right two) plots of α for the JC69 model. Red is for Gibbs and blue is for the symmetrized MH algorithm.

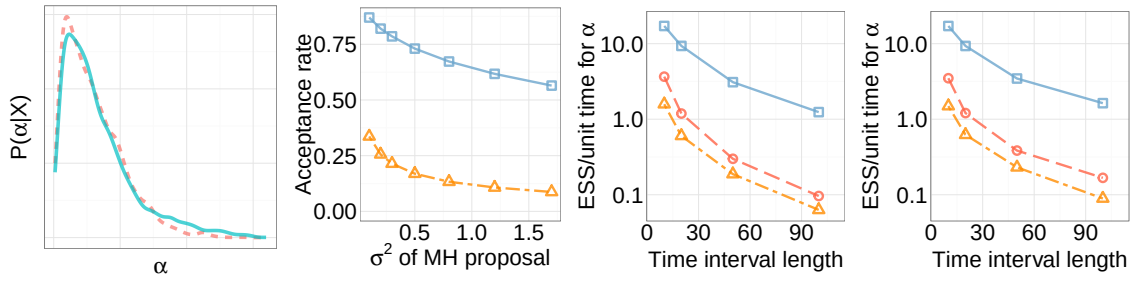

Figure 6.12. (a) Posterior $P(\alpha|X)$ in the JC69 model for Gibbs (dashed) and symmetrized MH (continuous). (b) MH acceptance rates for naïve and symmetrized MH. (c) and (d): ESS/sec against t_{end} for $\kappa = 2$ with: (c) number of observations fixed, and (d) observation rate fixed. Squares, triangles and circles are symmetrized MH, naïve MH and Gibbs.

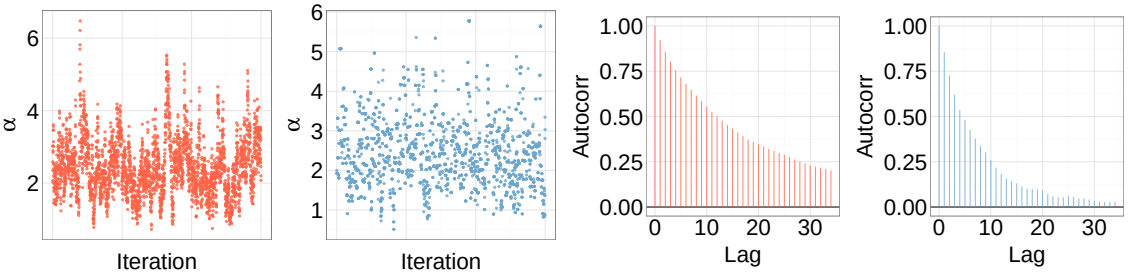

Figure 6.13. Trace and autocorrelation plots for Gibbs (left two panels) and symmetrized MH (right two panels). All plots are for the time-inhomogeneous immigration model with 10 states.

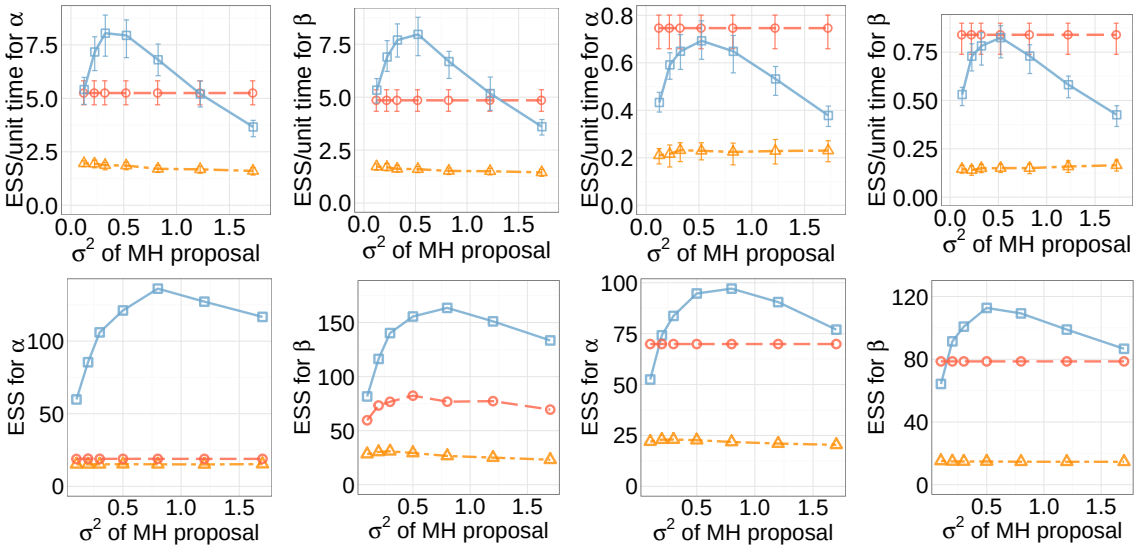

Figure 6.14. ESS/sec (top row) and raw ESS per 1000 samples (bottom row) for the immigration model. The left two columns are α and β for 3 states, and the right two, for 10 states. Squares, triangles and circles are symmetrized MH, naïve MH, and Gibbs algorithm.

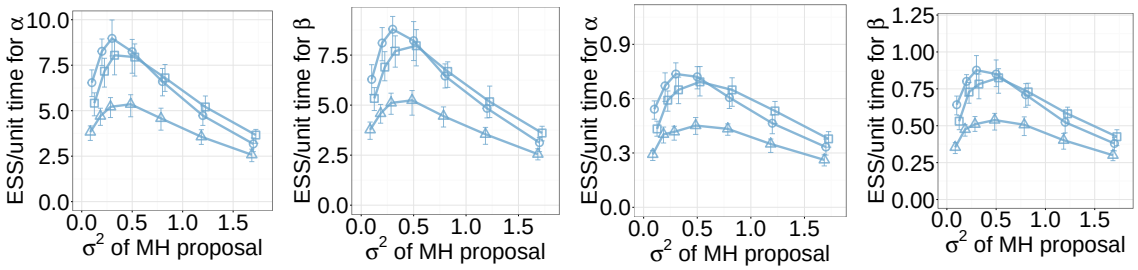

Figure 6.15. ESS/sec for symmetrized MH for the immigration model for different settings of $\Omega(\theta, \vartheta)$. The left two columns are for α and β with states, and the right two, with 10. Squares, circles and triangles correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

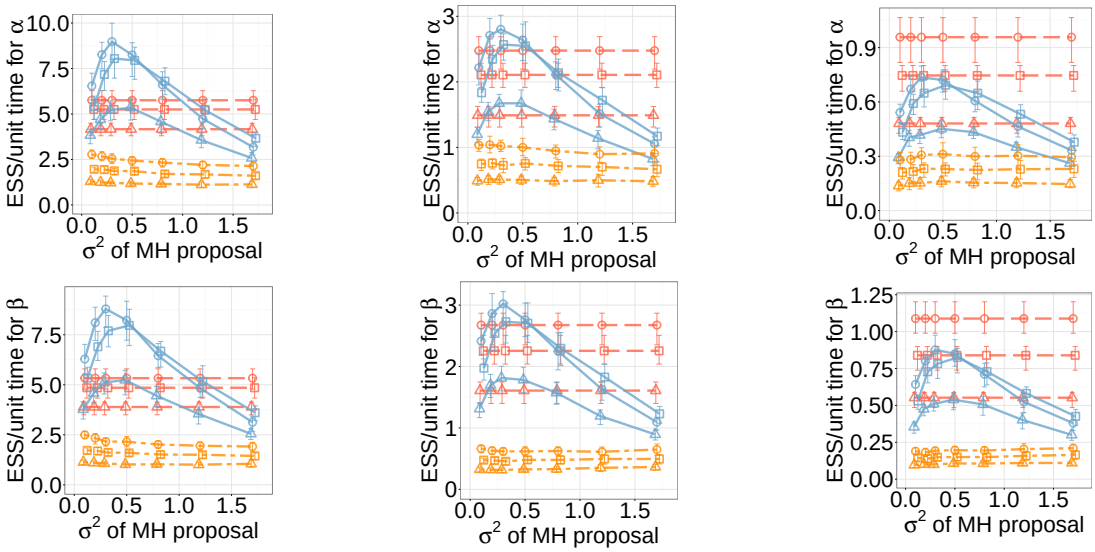

Figure 6.16. ESS/sec for the immigration model, the top three are for α for 3 states, 5 states, and 10 states. The bottom three are for β for 3 states, 5 states, and 10 states. Blue, yellow, and red are the symmetrized MH, naïve MH, Gibbs algorithm. Squares, circles and trianges correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

6.4 An immigration model with finite capacity

Next, we consider an M/M/N/N queue (Gross et al., 2011). The state space of this stochastic process is $\{0, 1, 2, 3, \cdots, N-1\}$ giving the number of customers/jobs/individuals in a system/population. Arrivals follow a rate-α Poisson process, moving the process from state i to $i+1$ for $i < N$. The system has a capacity of N, so any arrivals when the current state is N are discarded. Service times or deaths are exponentially distributed, with a rate that is now state-dependent: the system moves from i to $i-1$ with rate $i\beta$.

We follow the same setup as the first experiment: for $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ equal to $(3, 2, 5, 2)$, we place Gamma(α_0, α_1), and Gamma(β_0, β_1) priors on α, β. These prior distributions are used to sample transition matrices A, which, along with a uniform

distribution over initial states, are used to generate MJP trajectories. We observe these at integer-valued times according to a Gaussian likelihood. We consider three settings: $3, 5$ and 10 states.

Figure 6.14 plots the ESS per unit time (top row) as well as raw ESS values (bottom row) for the parameters α and β, again as we change the variance of the proposal kernel. The left two columns show these for α and β for the MJP state-space having size 3, while the right two columns show these for size 10. Our symmetrized MH algorithm does best for dimensions 3 and 5 , although now Gibbs sampling performs best for dimensionality 10 (although there is no significant different between the best proposal variance for our sampler and the Gibbs sampler). The Gibbs sampler performs so well partly because the conditionals over α and β are conjugate, following simple Gamma distributions. Also, unlike the earlier problem, the rate matrix is tridiagonal, and governed by two parameters, so that path-parameter coupling is now milder.

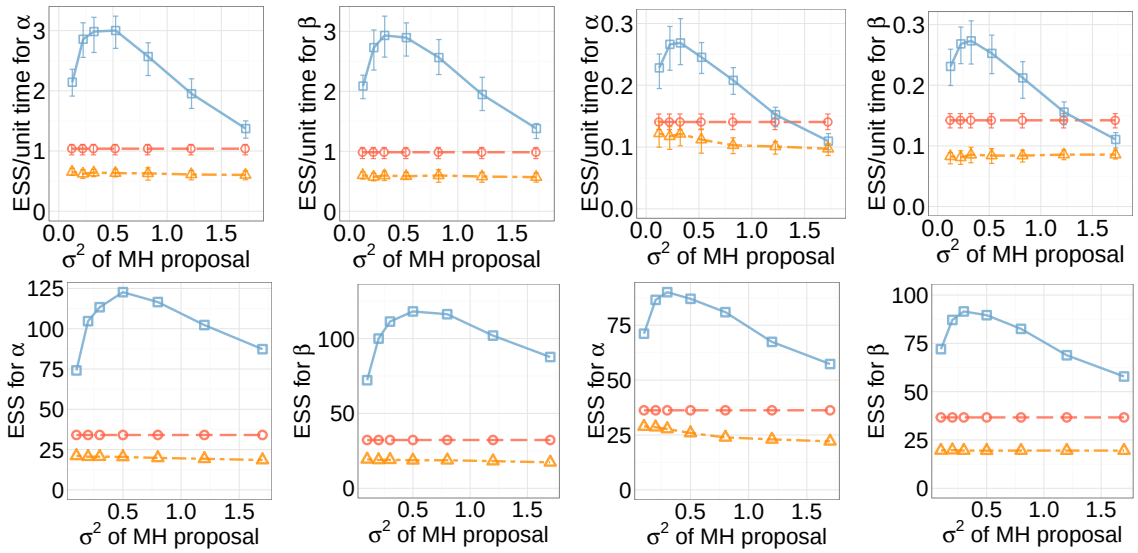

Figure 6.17. ESS/sec (top row) and raw ESS per 1000 samples (bottom row) for the time-inhomogeneous immigration model. The left columns are α and β for 3 states, and the right two for 10. Blue squares, yellow triangles and red circles are the symmetrized MH, naïve MH, and Gibbs algorithm.

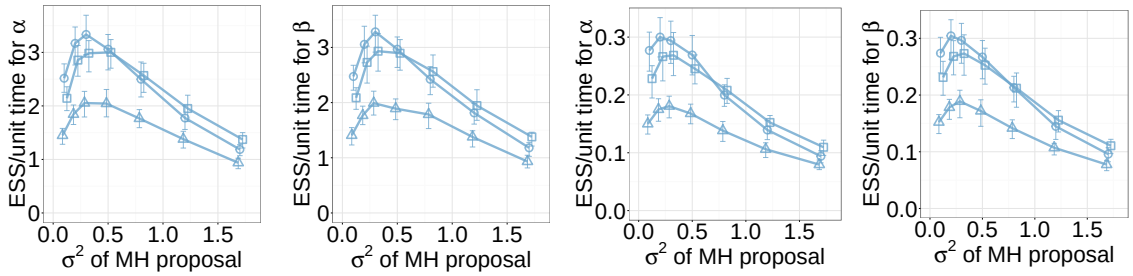

Figure 6.18. ESS/sec for symmetrized MH for the time-inhomogeneous immigration model for different settings of $\Omega(\theta, \vartheta)$. The left two columns are α and β for 3 states, and the right two for 10. Squares, circles and trianges correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

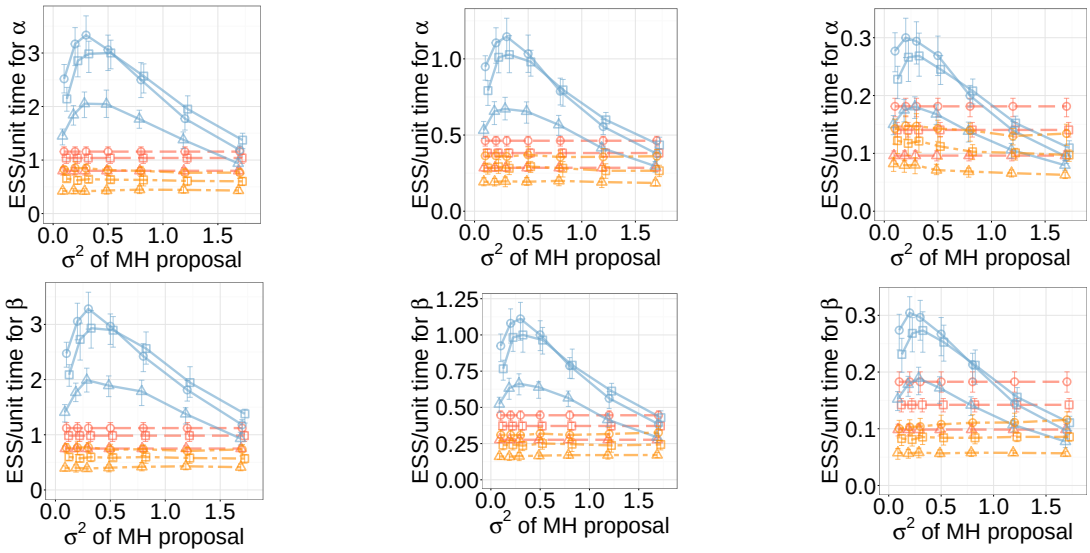

Figure 6.19. ESS/sec for the time-inhomogeneous immigration model, the top three are for α for 3 states, 5 states, and 10 states. The bottom three are for β for 3 states, 5 states, and 10 states. Blue, yellow, and red are the symmetrized MH, naïve MH, Gibbs algorithm. Squares, circles and trianges correspond to $\Omega(\theta, \vartheta)$ set to $(\max_s A_s(\theta) + \max_s A_s(\vartheta))$, $\max(\max_s A_s(\theta), \max_s A_s(\vartheta))$ and $1.5(\max_s A_s(\theta) + \max_s A_s(\vartheta))$.

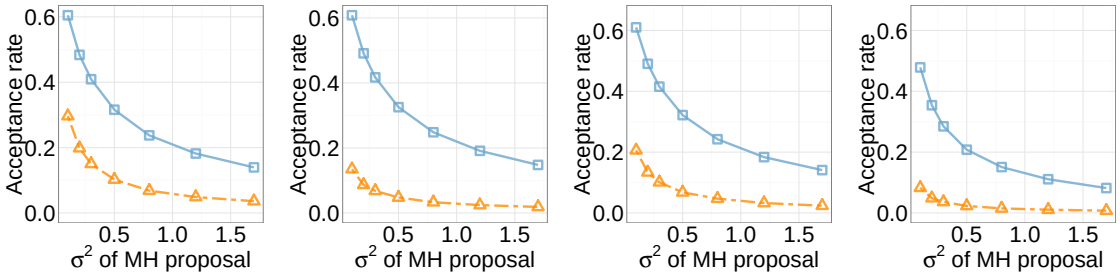

Figure 6.20. Acceptance Rate for α in the immigration model (left two) and time-inhomogeneous immigration model (right two) , the left two being dimension 3, and the right,dimension 10 and the right two being dimension 3, and the right,dimension 10. Blue square and yellow triangle curves represent symmetrized MH, and naïve MH algorithm. The multiplicative factor is 2.

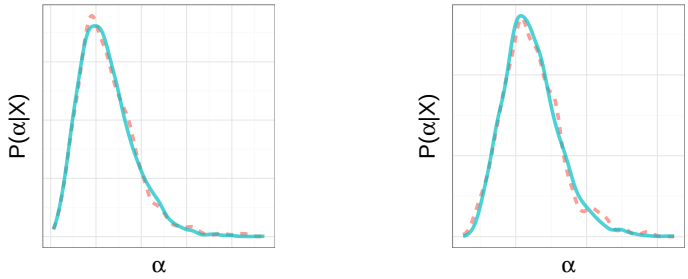

Figure 6.21. Posterior $P(\alpha|X)$ from Gibbs (dashed line) and symmetrized MH (solid line) for the immigration model(Left), and time-inhomogeneous immigration model(right)

A time-inhomogeneous immigration model: We extend the previous model to incorporate a known time-inhomogeneity. The arrival and death rates are now no longer constant, and are instead given by $A_{i,i+1}(t) = \alpha w(t)$ $(i = 0, 1, \cdots, N-1)$ respectively. While it is not difficult to work with sophisticated choices of $w(t)$, we limit ourselves to a simple piecewise-constant $w(t) = \lfloor \frac{t}{5} \rfloor$. Even such a simple change in the original model can dramatically affect the performance of the Gibbs sampler.

The top row of figure 6.17 plots the ESS per unit time for the parameters α (left) and β (right) for this model with capacity 3. Now, the symmetrized MH algorithm

is significantly more efficient, comfortably outperforming all samplers (including the Gibbs sampler) over a wide range of settings. Figure 6.17 shows performance for dimension 3 and dimension 10, once again the symmetrized MH-algorithm performans best over a range of settings of the proposal variance. We note that increasing the dimensionality of the state space results in a more concentrated posterior, shifting the optimal setting of the proposal variance to smaller values.

6.5 Chi site data for Escherichia coli

We consider a dataset recording positions of a particular DNA motif on the *E. coli* genome. These motifs consist of eight base pairs GCTGGTGG, and are called Chi sites (Fearnhead and Sherlock, 2006). The rates of occurence of Chi sites provide information about genome segmentation, allowing the identification of regions with high mutation or recombination rates. Following Fearnhead and Sherlock (2006), we use this data to infer a two-state piecewise-constant segmentation of the DNA strand. We focus on Chi sites along the inner (lagging) strand of the *E. coli* genome. We place a MJP prior over this segmentation, and indexing position along the strand with t, we write this as $S(t), t \in [0, 2319.838]$. To each state $s \in \{1, 2\}$, we assign a rate λ_s, which together with $S(t)$, defines a piecewise-constant rate function $\lambda_{S(t)}$. We model the Chi-site positions as drawn from a Poisson process with rate $\lambda_{S(t)}$, resulting in a Markov-modulated Poisson process (Scott and Smyth, 2003) (see also section 2.5). MJP transitions from state 1 to state 2 have rate α while transitions from state 2 to state 1 have rate β. We place Gamma$(2, 2)$, Gamma$(2, 3)$, Gamma$(3, 2)$, and Gamma$(1, 2)$ priors for α, β, λ_1, λ_2 respectively.

We use this setup to evaluate our symmetrized MH sampler along with Gibbs sampling (other algorithms perform much worse, and we do not include them). For our MH proposal distribution, we first run 2000 iterations of Gibbs sampling to estimate the posterior covariance of the vector $\theta = (\alpha, \beta, \lambda_1, \lambda_2)$, call this Σ_θ. Our MH

proposal distribution is then $q(\nu|\theta) = N(\nu|\theta, \sigma^2\Sigma_\theta)$ for different settings of σ^2 (the typical choice is $\sigma^2 = 1$), where we set $\Omega(\theta, \vartheta) = \max_s A_s(\theta) + \max_s A_s(\vartheta)$.

Figure 6.22 shows trace and autocorrelation plots for the parameter α produced by the Gibbs sampler (left) and our proposed sampler with κ set to 1. We see that this is a fairly hard MCMC sampling problem, however our sampler clearly outperforms Gibbs, which mixes very poorly. Both posterior distribution agreed with each other though, with a two sample-Kolmogorov Smirnov test returning a p-value of 0.1641.

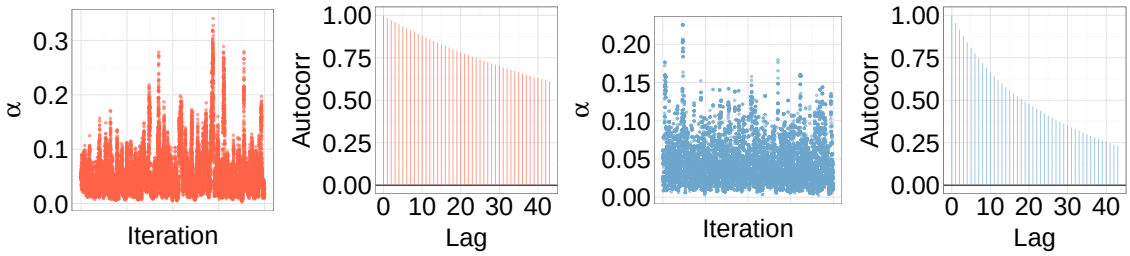

Figure 6.22. Trace and autocorrelation plots of posterior samples for α for the E. Coli data. The left two plots are the Gibbs sampler and the right two are the symmetrized MH.

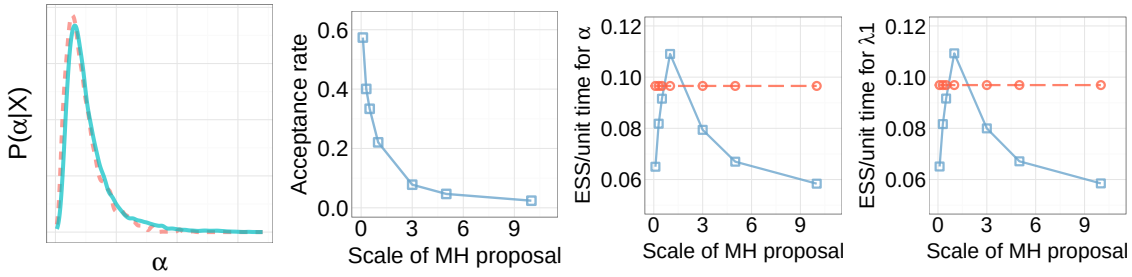

Figure 6.23. Posterior $P(\alpha|X)$ (a) from Gibbs (dashed line) and symmetrized MH (solid line) for the E. Coli data. Acceptance Rate(b) of α generated by the symmetrized MH algorithm for the E. Coli data. ESS/sec for (α, λ_1) for the E. Coli data(c, d). The circles (in blue) are our proposed sampler as we vary the variance of the proposal distribution. The straight line is Gibbs.

Figure 6.23 shows the ESS/s for different settings of σ^2, for parameters (α, λ_1). Both parameters have very similar results, and as suggested by the earlier figure, we see that for the typical setting of $\sigma^2 = 1$, our sampler ourperforms the Gibbs sampler. In this problem though, Gibbs sampling does outperform our method for large or small σ^2. This is because a) large or small σ^2 mean the proposal variance is too large or too small, and b) the Gibbs conditionals over the parameters are conjugate for this model. We expect the improvements our method offers to be more robust to the proposal distribution for more complex models without such conditional conjugacy.

7. GEOMETRIC ERGODICITY OF THE SYMMETRIZED METROPOLIS-HASTINGS ALGORITHM

7.1 Introduction

In this chapter, we show that if the ideal MCMC sampler is geometrically ergodic, then so is our sampler in Algorithm 7. We start with a review of geometric ergodicity. Informally, an MCMC algorithm is geometrically ergodic when the total variation distance between the distribution over states and the stationary distribution decreases geometrically with the number of iterations. Meyn and Tweedie (2009) provides more details, as well as sufficient conditions that we exploit in Theorem 7.2.1. Geometric ergodicity is an important property of an MCMC chain with stationary distribution $\mu(\cdot)$, guaranteeing that the central limit theorem (CLT) holds for ergodic averages calculated with MCMC samples $(\theta_1, \cdots, \theta_n)$. For a function f, the ergodic average is $\frac{1}{n}\sum_{i=1}^{n} f(\theta_i)$, for $n \geq 1$. Denote $P^n(\theta, A)$ for the n-step transition probability of a Markov chain$(\theta_0, \theta_1, \cdots, \theta_n, \cdots)$, which takes values in the space $\mathcal{X}$.

$$P^n(\theta, A) = P[\theta_n \in A | \theta_0 = \theta], \text{for } A \subseteq \mathcal{X}.$$

The distance between two different probability measures can be measured using total variation distance, defined as follows:

Definition 7.1.1 (total variation) *The total variation between two probability measures $p_1(\cdot)$ and $p_2(\cdot)$ is defined as follows.*

$$\|p_1(\cdot) - p_2(\cdot)\|_{TV} = sup_{A \subseteq \mathcal{X}} |p_1(A) - p_2(A)|.$$

We introduce $\phi-$irreducible property of a Markov chain and then give the definition of geometric ergodicity.

Definition 7.1.2 ($\phi-$irreducible) *A Markov chain is $\phi-$irreducible if there exists a non-zero σ finite measure ϕ on $\mathcal{X}$, such that for any $A \subseteq \mathcal{X}$ with $\phi(A) > 0$ and for all $\theta \in \mathcal{X}$, there exists $n > 0$, which depends on θ and A , such that $P^n(\theta, A) > 0$.*

Definition 7.1.3 (geometric ergodicity) *A Markov chain with stationary distribution $\mu(\cdot)$, is geometrically ergodic if*

$$\|P^n(\theta, \cdot) - \mu(\cdot)\|_{TV} \leq M(\theta)\rho^n, \text{for } n = 1, 2, \cdots$$

for some $\rho < 1$, where $M(\theta) < +\infty$, for π-a.e. $\theta \in \mathcal{X}$.

Small set condition and drift condition are defined below, which can lead to geometric ergodicity.

Definition 7.1.4 (small set) *A subset $B \subseteq \mathcal{X}$ is a small (n-small set) set if there exists a integer $n > 0$, and $\epsilon > 0$, and a probability measure $\nu(\cdot)$ on $\mathcal{X}$ such that $P^n(\theta, A) \geq \epsilon\nu(A)$, for all $\theta \in B$, and all measurable set $A \subseteq \mathcal{X}$.*

Definition 7.1.5 (drift condition) *A Markov chain with transition probability $P(\theta, d\theta')$ for $\theta, \theta' \in \mathcal{X}$, satisfies a drift condition if there are constants $0 < \lambda < 1$ and $b < +\infty$ and a set B, and a function $V : \mathcal{X} \to [1, +\infty]$ (called Lyapunov-Foster function), such that*

$$\int_{\mathcal{X}} P(\theta, d\theta')V(\theta) \leq \lambda V(\theta) + b\mathbb{1}_B(\theta), \text{for all } \theta \in \mathcal{X}.$$

The n-small set condition implies that for $\theta \in B$, the Markov chain takes n steps to forget its current location with probability ϵ. The drift condition ensures that for θ outside the set B, the the Markov chain drifts towards B. It is easy to see that if the space $\mathcal{X}$ is compact, then the drift condition is satisfied with the choice of B being the whole space $\mathcal{X}$ as well as b being $\sup_{\theta \in \mathcal{X}} V(\theta)$.

For the ideal sampler, we use $\Omega(\theta)$ as the so-called Lyapunov-Foster function to define the drift condition and $B_M = \theta : \Omega(\theta) \leq M$ is defined as the small set. For our symmetrized auxiliary variable MCMC sampler, $\lambda_1|W| + \Omega(\theta)$ is used as the

Lyapunov-Foster function, for some $\lambda_1 > 0$ while $B_{h,M} = (W, \theta, \vartheta) : |W| \leq h, \theta \in B_M$ is the 2-small set.

The theorem guaranteeing geometric ergodicity is the following. (See Meyn and Tweedie (2009) for detailed proofs.)

Theorem 7.1.1 (geometric ergodic theorem) *Consider a $\phi-$irreducible, aperiodic Markov chain. If there exists a small set B, and for the set B, the Markov chain satisfies the drift condition with a function $V(\theta)$ finite at some $\theta_0 \in \mathcal{X}$, then the Markov chain is geometric ergodic.*

7.2 Geometric ergodicity of the symmetrized MH algorithm

We derive conditions under which our symmetrized MH algorithm inherits mixing properties of an 'ideal' sampler that can compute the marginal likelihood $P(X|\theta)$. This algorithm proposes a new parameter ϑ from a distribution $q(\vartheta|\theta)$, and accepts with probability $\alpha_I(\theta, \vartheta; X) = 1 \wedge \frac{P(X,\vartheta)q(\theta|\vartheta)}{P(X,\theta)q(\vartheta|\theta)}$. The resulting Markov chain has transition probability $P_I(\theta'|\theta) = q(\theta'|\theta)\alpha_I(\theta, \theta'; X) + \left[1 - \int \mathrm{d}\vartheta q(\vartheta|\theta)\alpha_I(\theta, \vartheta; X)\right] \delta_\theta(\theta')$, the first term corresponding to acceptance, and the second, rejection (Meyn and Tweedie, 2009).

Our main result is Theorem 7.2.1, which shows that if the ideal MCMC sampler is geometrically ergodic, then so is our sampler in Algorithm 7. Before diving into the proofs, we first state our assumptions,

Assumption 7.2.1 *The uniformization rate is set as $\Omega(\theta, \vartheta) = \Omega(\theta) + \Omega(\vartheta)$, where $\Omega(\theta) = k_1 \max_s A_s(\theta) + k_0$, for some $k_1 > 1, k_0 > 0$.*

Although it is possible to specify broader conditions under which our result holds, for clarity we focus on this case. We can drop k_0 if $\inf_\theta \max_s A_s(\theta) > 0$

Assumption 7.2.2 *There exists a positive constant θ_0 such that for any θ_x, θ_y satisfying $\|\theta_x\| \geq \|\theta_y\| > \theta_0$, we have $\Omega(\theta_x) \geq \Omega(\theta_y)$.*

This assumption avoids book-keeping by making $\Omega(\theta)$ increase monotonically with θ.

Definition 7.2.1 *Let π_θ be the stationary distribution of the MJP with rate-matrix $A(\theta)$, and define $D_\theta = diag(\pi_\theta)$. Define $\tilde{A}(\theta) = D_\theta^{-1} A(\theta) D_\theta$, and the* reversibilization *of $A(\theta)$ as $R_A(\theta) = (A(\theta) + \tilde{A}(\theta))/2$.*

This definition is from Fill (1991), who shows that $R_A(\theta)$ is reversible with real eigenvalues, the smallest being 0. The larger its second smallest eigenvalue, the faster the MJP converges to its stationary distribution π_θ. Note that if $A(\theta)$ is reversible, then $R_A(\theta) = A(\theta)$.

Assumption 7.2.3 *Write $\lambda_2^{R_A}(\theta)$ for the second smallest eigenvalue of $R_A(\theta)$. There exist $\mu > 0, \theta_1 > 0$ such that for all θ satisfying $\|\theta\| > \theta_1$, we have $\lambda_2^{R_A}(\theta) \geq \mu \max_s A_s(\theta)$ (or equivalently from Assumption 7.2.1, $\lambda_2^{R_A}(\theta) \geq \mu\Omega(\theta)$), and $\min_s \pi_\theta(s) > 0$.*

This assumption is the strongest we need, requiring that $\lambda_2^{R_A}(\theta)$ (which sets the MJP mixing rate) grows at least as fast as $\max_s A_s(\theta)$. This is satisfied when, for example, all elements of $A(\theta)$ grow with θ at similar rates, controlling the relative stability of the least and most stable states. While not trivial, this is a reasonable assumption: the MCMC chain over MJP paths will mix well if we can control the mixing of the MJP itself. To better understand this, recall $B(\theta, \theta') = I + \frac{A(\theta)}{\Omega(\theta,\theta')}$ is the transition matrix of the embedded Markov chain from uniformization, this has the same stationary distribution π_θ as $A(\theta)$. Define the reversibilization $R_B(\theta, \theta')$ of $B(\theta, \theta')$ just as we did $R_A(\theta)$ from $A(\theta)$.

Lemma 7.2.1 *Consider $\|\theta\| > \max(\theta_0, \theta_1)$ and θ' such that $\frac{1}{K_0} \leq \frac{\Omega(\theta')}{\Omega(\theta)} \leq K_0$, where K_0 satisfies $(1 + \frac{1}{K_0})k_1 \geq 2$. For all such (θ, θ'), the Markov chain with transition matrix $B(\theta, \theta')$ converges geometrically to stationarity at a rate uniformly bounded away from 0.*

Proof A little algebra gives $R_B(\theta, \theta') = I + R_A(\theta)/\Omega(\theta, \theta')$. It follows that both R_A and R_B share the same eigenvectors, with eigenvalues satisfying $\lambda_{R_B}(\theta, \theta') = 1 - \frac{\lambda_{R_A}(\theta)}{\Omega(\theta,\theta')}$. The second largest eigenvalue $\lambda_2^{R_B}(\theta, \theta')$ of R_B and second smallest eigenvalue

$\lambda_2^{R_A}(\theta,\theta')$ of R_A then satisfy $\lambda_2^{R_B}(\theta,\theta') = 1-\frac{\lambda_2^{R_A}(\theta)}{\Omega(\theta,\theta')}$. From assumptions 7.2.1 and 7.2.3, and the lemma's assumptions, $1-\lambda_2^{R_B}(\theta,\theta') = \frac{\lambda_2^{R_A}(\theta)}{\Omega(\theta,\theta')} \geq \frac{\lambda_2^{R_A}(\theta)}{(K_0+1)\Omega(\theta)} \geq \frac{\mu}{K_0+1}$. Also, since $(1+\frac{1}{K_0})k_1 \geq 2$,

$$\Omega(\theta,\theta') = \Omega(\theta)+\Omega(\theta') \geq (1+\frac{1}{K_0})\Omega(\theta) > (1+\frac{1}{K_0})k_1 \max_s A_s(\theta) \geq 2\max_s A_s(\theta).$$

So for any state s, the diagonal element $B_s(\theta,\theta') = 1-\frac{A_s(\theta)}{\Omega(\theta,\theta')} > \frac{1}{2}$. From Fill (1991), this diagonal property and the bound on $1-\lambda_2^{R_B}(\theta,\theta')$ give the result. ■

Our overall proof strategy is to show that for $\|\theta\|$ and W large enough, the conditions of Lemma 7.2.1 hold with high probability. Lemma 7.2.1 then will imply that the distribution over latent states for the continuous-time MJP and its discrete-time counterpart embedded in W to be brought arbitrarily close to π_θ (and thus to each other), allowing our sampler to inherit mixing properties of the ideal sampler. We will exploit the boundedness of the set of remaining θ and W, to establish a 'small-set condition' where the MCMC algorithm forgets its state with some probability. These two conditions will be sufficient for geometric ergodicity. The next assumption states these small-set conditions for the ideal sampler.

Assumption 7.2.4 *For the ideal sampler with transition probability $p_I(\theta'|\theta)$:*
i) for each M, for the set $B_M = \{\theta : \Omega(\theta) \leq M\}$, there exists a probability measure ϕ and a constant $\kappa_1 > 0$ s.t. $\alpha_I(\theta,\theta';X)q(\theta'|\theta) \geq \kappa_1\phi(\theta')$ for $\theta \in B_M$. Thus B_M is a 1*-small set.*
ii) for M large enough, $\exists\rho < 1$ s. t. $\int \Omega(\nu)p_I(\nu|\theta)d\nu \leq (1-\rho)\Omega(\theta)+L_I$, $\forall\theta \notin B_M$.

These two conditions are standard small-set and drift conditions necessary for the ideal sampler to satisfy geometric ergodicity. The first implies that for θ in B_M, the ideal sampler 'forgets' its current location with probability κ_1. The second condition ensures that for θ outside this set, the ideal sampler drifts towards B_M. These two conditions together imply geometric mixing with rate equal or faster than κ_1 (Meyn and Tweedie, 2009). Observe that we have used $\Omega(\theta)$ as the so-called Lyapunov-Foster function to define the drift condition for the ideal sampler. This is the most

natural choice, though our proof can be tailored to different choices. Similarly, we could easily allow B_M to be an n-small set for any $n \geq 1$ (so the ideal sampler needs n steps before it can forget its current value in B_M); we restrict ourselves to the 1-small case for clarity.

Assumption 7.2.5 *$\exists\ u > \ell > 0$ s.t. $\prod P(X|s_o, \theta) \in [\ell, u]$ for any state s_o and θ.*

This assumption follows Miasojedow and w. Niemiro (2017), and holds if θ does not include parameters of the observation process (or if so, the likelihood is finite and nonzero for all settings of θ). We can relax this assumption, though this will introduce technicalities unrelated to our focus, which is on complications in parameter inference arising from the continuous-time dynamics, rather than the observation process.

Assumption 7.2.6 *Given the proposal density $q(\nu|\theta)$, $\exists \eta_0 > 0, \theta_2 > 0$ such that for θ satisfying $\|\theta\| > \theta_2$, $\int_\Theta \Omega(\nu)^2 q(\nu|\theta) d\nu \leq \eta_0 \Omega(\theta)^2$.*

This mild requirement can be satisfied by choosing a proposal distribution q that does not attempt to explore large θ's too aggressively. The next corollary follows from a simple application of the Cauchy-Schwarz inequality.

Corollary 7.2.1 *Given the proposal density $q(\nu|\theta)$, $\exists \eta_1 > 0, \theta_2 > 0$ such that for θ satisfying $\|\theta\| > \theta_2$, $\int_\Theta \Omega(\nu) q(\nu|\theta) d\nu \leq \eta_1 \Omega(\theta)$.*

Proof From assumption 7.2.6, we have $\int_\Theta \Omega(\nu)^2 q(\nu|\theta) d\nu \leq \eta_0 \Omega(\theta)^2$ for θ satisfying $\|\theta\| > \theta_2$. For such θ, by the Cauchy-Schwarz inequality, we have

$$\left[\int_\Theta \Omega(\nu) q(\nu|\theta) d\nu\right]^2 \leq \int_\Theta \Omega(\nu)^2 q(\nu|\theta) d\nu \cdot \int_\Theta q(\nu|\theta) d\nu \leq \eta_0 \Omega(\theta)^2.$$

So for θ satisfying $\|\theta\| > \theta_2$, we have $\int_\Theta \Omega(\nu) q(\nu|\theta) d\nu \leq \sqrt{\eta_0} \Omega(\theta)$. ■

We need two further assumptions on the proposal distribution $q(\theta'|\theta)$.

Assumption 7.2.7 *For any $\epsilon > 0$, there exist finite M_ϵ, $\theta_{3,\epsilon}$ such that for θ satisfying $\|\theta\| > \theta_{3,\epsilon}$, the condition $q(\{\theta' : \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)} \leq M_\epsilon\}|\theta) > 1 - \epsilon$ holds.*

This holds, when e.g. $p(\theta)$ is a gamma distribution, and $q(\theta'|\theta)$ is Gaussian.

Assumption 7.2.8 *For any $\epsilon > 0$ and $K > 1$, there exists $\theta^K_{4,\epsilon}$ such that for θ satisfying $\|\theta\| > \theta^K_{4,\epsilon}$, the condition $q(\{\theta' : \frac{\Omega(\theta')}{\Omega(\theta)} \in \left[\frac{1}{K}, K\right]\}|\theta) > 1 - \epsilon$ holds.*

This holds when e.g. $q(\theta'|\theta)$ is a centered on θ and has finite variance.

Theorem 7.2.1 *Under the above assumptions, our symmetrized auxiliary variable MCMC sampler in algorithm 7 is geometrically ergodic.*

Proof This theorem follows from two lemmas we will prove. Lemma 7.2.2 shows there exist small sets $\{(W, \theta, \vartheta) : \lambda_1|W| + \Omega(\theta) < M\}$ for $\lambda_1, M > 0$, within which our sampler forgets its current state with some positive probability. Lemma 7.2.4 shows that for appropriate (λ_1, M), our sampler drifts towards this set whenever outside. Together, these two results imply geometric ergodicity (Meyn and Tweedie, 2009, Theorems 15.0.1 and Lemma 15.2.8). If $\sup_\theta \Omega(\theta) < \infty$, we just need the small set $\{(W, \theta, \vartheta : |W| < M\}$ for some M. ■

For easier comparison with the ideal sampler, we begin an MCMC iteration from step 5 in Algorithm 7. Thus, our sampler operates on (θ, ϑ, W), with θ the current parameter, ϑ the auxiliary variable, and W the Poisson grid. An MCMC iteration updates this by (a) sampling states V with a backward pass, (b) discarding ϑ and self-transition times, (c) sampling ν from $q(\nu|\theta)$, (d) sampling U' given (θ, ν, S, T), setting $W' = T \cup U'$, and discarding S, (e) proposing to swap (θ, ν) and then (f) accepting or rejecting with a forward pass. On acceptance, $\theta' = \nu$ and $\vartheta' = \theta$, and on rejection, $\theta' = \theta$ and $\vartheta' = \nu$, so that the MCMC state at the end of the iteration is $(\theta', \vartheta', W')$. We write $(\theta'', \vartheta'', W'')$ for the MCMC state after two iterations. Recall that step (a) actually assigns states V to W. T are the elements of W where V changes value, and S are the corresponding elements of V. The remaining elements U are the elements of W corresponding to self-transitions.

Before we start our proof, we recall some notation used in our proof. The figure 1.1 shows a realization $S(t)$ of an MJP with rate matrix $A(\theta)$ and initial distribution π_0

over an interval $[0, t_{end}]$. The empty circles are the thinned events. The crosses are observations X. π_0 is the initial distribution over states, and π_θ is the staionary distribution of the MJP. $p(\theta)$ is the prior over θ, and $q(\nu|\theta)$ is the proposal distribution.

- The uniformized representation of $S(t)$ is the pair (V, W), with the Poisson grid $W = [w_1, w_2, w_3, w_4, w_5, w_6, w_7]$ and the states $V = [v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7]$ assigned with through a Markov chain with initial distribution π_0 and transition matrix $B(\theta, \theta')$. In the figure, the circles (filled and empty) correspond to W.

- The more standard representation of $S(t)$ is the pair (S, T). Here T are the elements of W which are true jump times (when V changes value), and S are the corresponding elements of V. U are the remaining elements of W corresponding to self-transitions. Here, $T = [w_2, w_4, w_7]$ and $U = [w_1, w_3, w_5, w_6]$.

- The filled circles represent W_X, which are the elements of W containing observations. V_X are the states corresponding to W_X. In this example, $W_X = [w_2, w_5, w_7] \cup \{0\}$ and $V_X = [v_2, v_5, v_7] \cup \{v_0\}$.

- We write $|W^{\downarrow}|$ for the minimum number of elements of W between successive pairs of observations (including start time 0). In this example, $|W^{\downarrow}| = \min(3, 3, 2) = 2$.

- $P(X|W, \theta, \theta')$ is the marginal distribution of X on W under a Markov chain with transition matrix $B(\theta, \theta')$ (after integrating out the state information V). Recall that the LHS does not depend on θ' because of uniformization.

- $P(X|\theta)$ is the marginal probability of the observations under the rate-$A(\theta)$ MJP. $P(X|\theta) = \int_W P(X|W, \theta, \theta')P(W|\theta, \theta')dW$.

- $P_B(V_X|W, \theta, \theta')$ is the probability distribution over states V_X for the Markov chain with transition matrix $B(\theta, \theta')$ on the grid W, with the remaining elements of V integrated out.

- $P_{st}(V_X|\theta)$ is the probability of V_X when elements of V_X are sampled i.i.d. from π_θ).
- $P_{st}(X|\theta)$ is the marginal probability of X when V_X is drawn from $P_{st}(V_X|\theta)$.

We first bound self-transition probabilities of the embedded Markov chain from 0:

Proposition 7.2.1 *The posterior probability that the embedded Markov chain makes a self-transition,* $P(V_i = V_{i+1}|W, X, \theta, \vartheta) \geq \delta_1 > 0$*, for any* θ, ϑ, W*.*

Proof We use k_0 from assumption 7.2.1 to bound *a priori* self-transition probabilities:

$$P(V_{i+1} = s|V_i = s, W, \theta, \vartheta) = B_{ss}(\theta, \vartheta) = 1 - \frac{A_s(\theta)}{\Omega(\theta, \vartheta)} \geq 1 - \frac{A_s(\theta)}{\Omega(\theta)} \geq 1 - \frac{1}{k_0}.$$

We then have

$$\begin{aligned}
P(V_i = V_{i+1}|W, X, \theta, \vartheta) &= \sum_v P(V_i = V_{i+1} = v|W, X, \theta, \vartheta) \\
&= \sum_v \frac{P(V_i = V_{i+1} = v, X|W, \theta, \vartheta)}{P(X|W, \theta, \vartheta)} \\
&= \sum_v \frac{P(X|V_i = V_{i+1} = v, W, \theta, \vartheta)P(V_i = V_{i+1} = v|W, \theta, \vartheta)}{P(X|W, \theta, \vartheta)} \\
&\geq \frac{\ell}{u} \sum_v P(V_i = V_{i+1} = v|W, \theta, \vartheta) \\
&= \frac{\ell}{u} \sum_v P(V_{i+1} = v|V_i = v, W, \theta, \vartheta)P(V_i = v|\theta, \vartheta) \\
&\geq \frac{\ell}{u}(1 - \frac{1}{k_0}) \doteq \delta_1 > 0.
\end{aligned}$$

■

The proof exploits the bounded likelihood from assumption 7.2.5. A simple by-product of the proof is the following corollary:

Corollary 7.2.2 $P(V_{i+1} = s|V_s = s, W, X, \theta, \vartheta) \geq \delta_1 > 0$*, for any* θ, ϑ, W, s*.*

Lemma 7.2.2 *For all $M, h > 0$, the set $B_{h,M} = \{(W, \theta, \vartheta) : |W| \leq h, \theta \in B_M\}$ is a 2-small set under our proposed sampler. Thus, for all (W, θ, ϑ) in $B_{h,M}$, the two-step transition probability satisfies $P(W'', \theta'', \vartheta''|W, \theta, \vartheta) \geq \rho_1 \phi_1(W'', \theta'', \vartheta'')$ for a constant ρ_1 and a probability measure ϕ_1 independent of the initial state.*

Proof Recall the definition of B_M, and of an n-small set from Assumption 7.2.4. The 1-step transition probability of our MCMC algorithm consists of two terms, corresponding to the proposed parameter being accepted and rejected. Discarding the latter, we have

$$P(W', \theta', \vartheta'|W, \theta, \vartheta, X) \geq q(\theta'|\theta)\delta_\theta(\vartheta')\alpha(\theta, \theta', W'; X) \sum_{S,T} \left[P(S, T|W, \theta, \vartheta, X)P(W'|S, T, \theta, \theta')\right].$$

This follows from steps (c) to (e) in the reordered algorithm. The summation is over all (S, T) values produced by the backward pass (which are then discarded after sampling W'). We have used the fact that given (S, T), $P(W'|S, T, \theta, \theta', X)$ is independent of X.

We bound the summation over (S, T) by considering only terms with S constant. When this constant is state s^*, we write this as $(S = [s^*], T = \emptyset)$. This corresponds to $|W|$ self-transitions after starting state $S_0 = s^*$. Then the first term in the square brackets becomes

$$\begin{aligned} P(S = [s^*], T = \emptyset|W, \theta, \vartheta, X) &= P(S_0 = s^*|X, W, \theta, \vartheta) \prod_{i=0}^{|W|-1} P(V_{i+1} = s^*|V_i = s^*, X, W, \theta, \vartheta) \\ &\geq P(S_0 = s^*|X, W, \theta, \vartheta)\delta_1^{|W|} \qquad \text{(from Corollary 7.2.2).} \end{aligned}$$

With $S(t)$ fixed at s^*, W' is distributed as a Poisson process with rate $\Omega(\theta') + \Omega(\theta) - A_{s^*}(\theta)$. Write PoissProc$(W'|R(t))$ for the probability of W' under a rate-$R(t)$ Poisson process on $[0, t_{end}]$, so that $P(W'|S = [s^*], T = \emptyset, \theta', \theta) =$ PoissProc$(W'|\Omega(\theta') + \Omega(\theta) -$

$A_{s^*}(\theta)$). Then, from the Poisson superposition theorem, writing $2^{W'}$ for the power set of W, we have

$$\begin{aligned}
P(W'|S=[s^*], T=\emptyset, \theta', \theta) &= \sum_{Z\in 2^{W'}} \text{PoissProc}\left(Z|\Omega(\theta')\right) \text{PoissProc}\left(W'\backslash Z|\Omega(\theta) - A_{s^*}(\theta)\right) \\
&\geq \text{PoissProc}(W'|\Omega(\theta'))\text{PoissProc}(\emptyset|\Omega(\theta) - A_{s^*}(\theta)) \\
&\geq \text{PoissProc}(W'|\Omega(\theta'))\text{PoissProc}(\emptyset|\Omega(\theta)) \\
&\geq \text{PoissProc}(W'|\Omega(\theta'))\exp(-Mt_{end}), \quad \text{since for } \theta \in B_M,\ \Omega(\theta) \leq M.
\end{aligned}$$

Thus we have

$$\begin{aligned}
\sum_{S,T} P(S,T,W'|W,\theta,\vartheta,X) &\geq \sum_{s^*} P(S=[s^*], T=\emptyset|W,\theta,\vartheta,X)P(W'|S=[s^*], T=\emptyset,\theta',\theta) \\
&\geq \delta_1^{|W|}\exp(-Mt_{end})\text{PoissProc}(W'|\Omega(\theta')). \qquad (7.1)
\end{aligned}$$

Finally, we consider the third term in the square brackets. Using assumption 7.2.5,

$$\begin{aligned}
\alpha(\theta,\theta',W';X) &= 1 \wedge \frac{P(X|W',\theta',\theta)/P(X|\theta')}{P(X|W',\theta,\theta')/P(X|\theta)} \cdot \frac{P(X|\theta')q(\theta|\theta')p(\theta')}{P(X|\theta)q(\theta'|\theta)p(\theta)} \\
&\geq 1 \wedge \frac{\ell^2}{u^2} \cdot \frac{P(X|\theta')q(\theta|\theta')p(\theta')}{P(X|\theta)q(\theta'|\theta)p(\theta)} \geq \alpha_I(\theta,\theta';X)\frac{\ell^2}{u^2}. \qquad (7.2)
\end{aligned}$$

Inside $B_{h,M}$, $|W| \leq h$, and by assumption 7.2.4, $q(\theta'|\theta)\alpha_I(\theta,\theta';X) \geq \kappa_1\phi(\theta')$. Then the three inequalities above let us simplify the equation at the start of the proof:

$$\begin{aligned}
P(W',\theta',\vartheta'|W,\theta,\vartheta) &\geq \frac{\ell^2}{u^2}\delta_1^h \exp(-Mt_{end})\delta_\theta(\vartheta')\kappa_1\text{PoissProc}(W'|\Omega(\theta')\phi(\theta') \\
&\stackrel{\text{def}}{=} \rho_1\delta_\theta(\vartheta')\text{PoissProc}(W'|\Omega(\theta')\phi(\theta').
\end{aligned}$$

Write $F_{Poiss(a)}$ for the CDF of a rate-a Poisson. The two-step transition satisfies

$$\begin{aligned}
P(W'', \theta'', \vartheta''|W, \theta, \vartheta) &\geq \int_{B_{h,M}} P(W'', \theta'', \vartheta''|W', \theta', \vartheta')P(W', \theta', \vartheta'|W, \theta, \vartheta)\mathrm{d}W'\mathrm{d}\theta'\mathrm{d}\vartheta' \\
&\geq \int_{B_{h,M}} \rho_1 \delta_{\theta'}(\vartheta'')\mathrm{PoissProc}(W''|\Omega(\theta''))\phi(\theta'') \\
&\qquad \rho_1 \delta_{\theta}(\vartheta')\mathrm{PoissProc}(W'|\Omega(\theta'))\phi(\theta')\mathrm{d}W'\mathrm{d}\theta'\mathrm{d}\vartheta' \\
&\geq \rho_1^2 \phi(\theta'')\mathrm{PoissProc}(W''|\Omega(\theta'')) \int_{B_{h,M}} \delta_{\theta'}(\vartheta'')F_{Poiss(\Omega(\theta'))}(h)\phi(\theta')\mathrm{d}\theta' \\
&\geq \rho_1^2 \mathrm{PoissProc}(W''|\Omega(\theta''))\phi(\theta'')\phi(\vartheta'')F_{Poiss(\Omega(\vartheta''))}(h)\delta_{B_{h,M}}(\vartheta'') \\
&\geq \rho_1^2 \mathrm{PoissProc}(W''|\Omega(\theta''))\phi(\theta'')\phi(\vartheta'')\delta_{B_{h,M}}(\vartheta'')\exp(-\Omega(\vartheta'')) \qquad (7.3)
\end{aligned}$$

The last line uses $F_{Poiss(a)}(h) \geq F_{Poiss(a)}(0) = \exp(-a)\ \forall a$, and gives our result, with $\phi_1(W'', \theta'', \vartheta'') \propto \mathrm{PoissProc}(W''|\Omega(\theta''))\phi(\theta'')\phi(\vartheta'')\delta_{B_{h,M}}(\vartheta'')\exp(-\Omega(\vartheta''))$. ■

We have established the small set condition: for a point inside $B_{h,M}$ our sampler forgets its state with nonzero probability, sampling a new state from $\phi_1(\cdot)$. We next establish a drift condition, showing that outside $B_{h,M}$, the algorithm drifts back towards it (Lemma 7.2.4). We first establish a result needed when $\max_s |A_s(\theta)|$ is unbounded as θ increases. This states that the acceptance probabilities of our sampler and the ideal sampler can be brought arbitrarily close outside a small set, so long as $\Omega(\theta)$ and $\Omega(\theta')$ are sufficiently close.

Lemma 7.2.3 *Suppose* $\frac{1}{K_0} \leq \frac{\Omega(\theta)}{\Omega(\theta')} \leq K_0$, *for* K_0 *satisfying* $(1 + \frac{1}{K_0})k_1 \geq 2$ *(*k_1 *is from Assumption 7.2.1). Write* $|W^{\downarrow}|$ *for the minimum number of elements of grid* W *between any successive pairs of observations. For any* $\epsilon > 0$, *there exist* $w_\epsilon^{K_0}, \theta_{5,\epsilon}^{K_0} > 0$ *such that* $|P(X|W, \theta, \theta') - P(X|\theta)| < \epsilon$ *for any* (W, θ) *with* $|W^{\downarrow}| > w_\epsilon^{K_0}$ *and* $\|\theta\| > \theta_{5,\epsilon}^{K_0}$.

Proof From lemma 7.2.1, for all θ, θ' satisfying the lemma's assumptions, the Markov chain with transition matrix $B(\theta, \theta')$ converges geometrically to stationarity distribution π_θ at a rate uniformly bounded away from 0. By setting $|W^{\downarrow}|$ large enough, for all such (θ, θ') and for any initial state, the Markov chain would have mixed beween

each pair of observations, with distribution over states returning arbitrarily close to π_θ.

Write W_X for the indices of the grid W containing observations, and V_X for the Markov chain state at these times. Let $P_B(V_X|W,\theta,\theta')$ be the probability distribution over V_X under the Markov chain with transition matrix B given W and $P_{st}(V_X|\theta)$ be the probability of V_X sampled independently under the stationary distribution. Let $P(X|W,\theta,\theta')$ be the marginal probability of the observations X under that Markov chain $B(\theta,\theta')$ given W. Dropping W and θ' from notation, $P(X|\theta)$ is the probability of the observations under the rate-$A(\theta)$ MJP.

From the first paragraph, for $|W^\downarrow| > w_0$ for large enough w_0, $P_B(V_X|W,\theta,\theta')$ and $P_{st}(V_X|W,\theta)$ can be brought ϵ' close. Then for any W with $|W^\downarrow| > w_0$, we have

$$\begin{aligned}|P(X|W,\theta,\theta')-P_{st}(X|\theta)| &= |\sum_{V_X} P(X|V_X,\theta)[P_B(V_X|W,\theta,\theta') - P_{st}(V_X|\theta)]| \\ &\leq \sum_{V_X} P(X|V_X,\theta)|P_B(V_X|W,\theta,\theta') - P_{st}(V_X|\theta)| \leq \epsilon'',\end{aligned}$$

using $P(X|V_X,\theta) \leq u$ (Assumption 7.2.5), and $\sum_{V_X} |P_B(V_X|W,\theta,\theta')-P_{st}(V_X|\theta)| < \epsilon$. For large θ, we prove a similar result in the continuous case by uniformization. For any θ',

$$P(X|\theta) = \int \mathrm{d}W\, P(X|W,\theta,\theta')\mathrm{PoissProc}(W|\Omega(\theta)+\Omega(\theta')).$$

We split this integral into two parts, one over the set $\{|W^\downarrow| > w_0\}$, and the second over its complement. On the former, for w_0 large enough, $|P(X|W,\theta,\theta') - P_{st}(X|\theta)| \leq \epsilon''$. For θ large enough, $\{|W^\downarrow| > w_0\}$ occurs with arbitrarily high probability for any θ'. Since the likelihood is bounded, the integral over the second set can be made arbitrarily small (say, ϵ'' again). Finally, from the triangle inequality,

$$\begin{aligned}|P(X|\theta) - P(X|W,\theta,\theta')| &\leq |P(X|\theta) - P_{st}(X|\theta)| + |P_{st}(X|\theta) - P(X|W,\theta,\theta')| \\ &\leq (\epsilon'' + \epsilon'') + \epsilon'' \stackrel{\text{def}}{=} \epsilon\end{aligned}$$

■

The previous lemma bounds the difference in probability of observations under the discrete-time and continuous-time processes for θ and $|W|$ large enough. The next result uses this to bound with high probability the different in acceptance probabilities of the ideal sampler, and our proposed sampler with a grid W.

Proposition 7.2.2 *Let (W, θ, ϑ) be the current state of the sampler. Then, for any ϵ, there exists $\theta_\epsilon > 0$ as well as a set $E_\epsilon \subseteq \{(W', \theta') : |\alpha_I(\theta, \theta'; X) - \alpha(\theta, \theta'; W', X)| \leq \epsilon\}$, such that for θ satisfying $\|\theta\| > \theta_\epsilon$ and any ϑ, we have $P(E_\epsilon | W, \theta, \vartheta) > 1 - \epsilon$.*

Proof Fix $\epsilon > 0$ and $K > 1$ satisfying $(1 + \frac{1}{K})k_1 \geq 2$.

- From assumption 7.2.7, there exist M_ϵ and $\theta_{1,\epsilon}$, such that $P(\frac{q(\theta|\theta')p(\theta')}{q(\theta'|\theta)p(\theta)} \leq M_\epsilon) > 1 - \epsilon/2$ for θ satisfying $\|\theta\| > \theta_{1,\epsilon}$. Define $E_1^\epsilon = \{\theta' s.t. \ \frac{q(\theta|\theta')p(\theta')}{q(\theta'|\theta)p(\theta)} \leq M_\epsilon\}$.

- Define $E_2^K = \{\theta' s.t. \ \frac{\Omega(\theta')}{\Omega(\theta)} \in [1/K, K]\}$. Following assumption 7.2.8, define $\theta_{2,\epsilon}^K$ such that $P(E_2^K | \theta) > 1 - \epsilon/2$ for all θ satisfying $\|\theta\| > \theta_{2,\epsilon}^K$.

- On the set E_2^K, $\Omega(\theta') \leq K\Omega(\theta)$ (and also $\Omega(\theta) \leq K\Omega(\theta')$). Lemma 7.2.3 ensures that there exist $\theta_{3,\epsilon}^K > 0, w_\epsilon^K > 0$, such that for $|W^\downarrow| > w_\epsilon^K$, $\|\theta\| > \theta_{3,\epsilon}^K$ and $\|\theta'\| > \theta_{3,\epsilon}^K$, we have $|P(X|W, \theta', \theta) - P(X|\theta')| < \epsilon$, and $|P(X|W, \theta, \theta') - P(X|\theta)| < \epsilon$. Define $E_{3,\epsilon}^K = \{\theta' s.t. \|\theta'\| > \theta_{3,\epsilon}^K\}$.

- Define $E_{4,\epsilon}^K = \{W s.t. \ |W^\downarrow| > w_\epsilon^K\}$. Set $\theta_{4,\epsilon}^K$, so that for $\|\theta\| > \theta_{4,\epsilon}^K$, $P(E_{4,\epsilon}^K | E_2^K, E_1^\epsilon) > 1 - \epsilon$.. This holds since W comes from a Poisson processes, whose rate can be made arbitrarily large by increasing $\Omega(\theta)$.

- From assumption 7.2.2, there exists θ_0, such that $\Omega(\theta)$ increases as $\|\theta\|$ increases, for θ satisfying $\|\theta\| > \theta_0$. Set $\theta_\epsilon = \max(\theta_0, \theta_{1,\epsilon}, \theta_{2,\epsilon}^K, \theta_{3,\epsilon}^K, \theta_{4,\epsilon}^K)$.

Now consider the difference

$$\begin{aligned} |\alpha(\theta, \theta'; W, X) - \alpha_I(\theta, \theta'; X)| &= | \ 1 \wedge \frac{P(X|W, \theta', \theta) q(\theta|\theta') p(\theta')}{P(X|W, \theta, \theta') q(\theta'|\theta) p(\theta)} - 1 \wedge \frac{P(X|\theta') q(\theta|\theta') p(\theta')}{P(X|\theta) q(\theta'|\theta) p(\theta)} \ | \\ &\leq | \ \frac{P(X|W, \theta', \theta)}{P(X|W, \theta, \theta')} - \frac{P(X|\theta')}{P(X|\theta)} \ | \ \frac{q(\theta|\theta') p(\theta')}{q(\theta'|\theta) p(\theta)}. \end{aligned}$$

On E_1^ϵ, $\frac{q(\theta|\theta')p(\theta')}{q(\theta'|\theta)p(\theta)} \leq M_\epsilon$. Since $P(X|W,\theta,\theta')$ and $P(X|\theta)$ are lower-bounded by ℓ, for any $\epsilon > 0$ we can find a K such that on $E_2^K \cap E_{3,\epsilon}^K$,

$$|\frac{P(X|W,\theta',\theta)}{P(X|W,\theta,\theta')} - \frac{P(X|\theta')}{P(X|\theta)}| < \epsilon/M_\epsilon.$$

This means that on $E_1^\epsilon \cap E_2^K \cap E_{3,\epsilon}^K$, $|\alpha(\theta,\theta',W,X) - \alpha_I(\theta,\theta',X)| < \epsilon$.

For $\theta > \max(\theta_{1,\epsilon}, \theta_{2,\epsilon}^K)$ we have $P(E_2^K E_1^\epsilon) \geq P(E_2^K) + P(E_1^\epsilon) - 1 \geq 1 - \epsilon$.

When E_2^K holds, $\Omega(\theta') \geq \Omega(\theta)/K$. For θ large enough, we can ensure $\|\theta'\| > \theta_{3,\epsilon}^K$. So

$$P(E_1^\epsilon E_2^K E_{3,\epsilon}^K E_{4,\epsilon}^K) > (1-\epsilon)^2.$$

Finally, set $E_\epsilon \doteq E_1^\epsilon \cap E_2^K \cap E_{3,\epsilon}^K \cap E_{4,\epsilon}^K$, giving us our result.

■

Lemma 7.2.4 *(drift condition) There exist $\delta_2 \in (0,1), \lambda_1 > 0$ and $L > 0$ such that* $\mathbb{E}\left[\lambda_1|W'| + \Omega(\theta')|W,\theta,\vartheta,X\right] \leq (1-\delta_2)\left(\lambda_1|W| + \Omega(\theta)\right) + L.$

Proof Since $W' = T \cup U'$, we consider $\mathbb{E}[|T||W,\theta,\vartheta,X]$ and $\mathbb{E}[|U'||W,\theta,\vartheta,X]$ separately. An upper bound of $\mathbb{E}[|T||W,\theta,\vartheta,X]$ can be derived directly from proposition 7.2.1:

$$\mathbb{E}[|T||W,\theta,\vartheta,X] = \mathbb{E}[\sum_{i=0}^{|W|-1} \mathbb{I}_{\{V_{i+1}\neq V_i\}}|W,\theta,\vartheta,X] \leq \sum_{i=0}^{|W|-1}(1-\delta_1) = |W|(1-\delta_1).$$

By corollary 7.2.1, there exist η_1, θ_2 such that for $\|\theta\| > \theta_2$, $\int \Omega(\nu)q(\nu|\theta)d\nu \leq \eta_1\Omega(\theta)$. Then,

$$\begin{aligned}\mathbb{E}[|U'||W,\theta,\vartheta,X] &= \mathbb{E}_{S,T,\nu}\mathbb{E}[|U'||S,T,W,\theta,\vartheta,\nu,X] = \mathbb{E}_{S,T,\nu}\mathbb{E}[|U'||S,T,W,\theta,\nu] \\ &\leq \mathbb{E}_{S,T,\nu}\left[t_{end}\Omega(\theta,\nu)\right] = t_{end}\int \Omega(\theta,\nu)q(\nu|\theta)\mathrm{d}\nu \\ &= t_{end}\left[\left(\Omega(\theta) + \int_\Theta \Omega(\nu)q(\nu|\theta)\mathrm{d}\nu\right)\right] \leq t_{end}(\eta_1+1)\Omega(\theta).\end{aligned}$$

To bound $\mathbb{E}\left[\Omega(\theta')|W,\theta,\vartheta,X\right]$, consider the transition probability over (W',θ'):

$$\begin{aligned}P(\mathrm{d}W',\mathrm{d}\theta'|W,\theta,\vartheta) = \mathrm{d}\theta'\mathrm{d}W'\Bigg[&q(\theta'|\theta)\sum_{S,T}P(S,T|W,\theta,\vartheta,X)P(W'|S,T,\theta,\theta')\alpha(\theta,\theta';W',X) \\ &+ \int q(\nu|\theta)\sum_{S,T}P(S,T|W,\theta,\vartheta,X)P(W'|S,T,\theta,\nu)(1-\alpha(\theta,\nu;W',X))\mathrm{d}\nu\delta_\theta(\theta')\Bigg].\end{aligned}$$

With $P(W'|W,\theta,\vartheta,\theta',X) = \sum_{S,T} P(S,T|W,\theta,\vartheta,X)P(W'|S,T,\theta,\theta')$, integrate out W':

$$P(\mathrm{d}\theta'|W,\theta,\vartheta) = \mathrm{d}\theta' \int \mathrm{d}W' \Big[q(\theta'|\theta)P(W'|W,\theta,\vartheta,\theta',X)\alpha(\theta,\theta';W',X) + \int q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)(1-\alpha(\theta,\nu;W',X))d\nu\delta_\theta(\theta') \Big]$$

Let $\int \Omega(\theta')P(\mathrm{d}\theta'|W,\theta,\vartheta) = I_1(W,\theta,\vartheta) + \Omega(\theta)I_2(W,\theta,\vartheta)$, with

$$I_1(W,\theta,\vartheta) = \int \mathrm{d}\theta'\Omega(\theta')q(\theta'|\theta) \int \mathrm{d}W' P(W'|W,\theta,\vartheta,\theta',X)\alpha(\theta,\theta';W',X),$$
$$I_2(W,\theta,\vartheta) = \int \mathrm{d}\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)(1-\alpha(\theta,\nu;W',X)).$$

Consider the second term I_2. From Proposition 7.2.2, for any positive ϵ, there exists $\theta_\epsilon > 0$ such that the set E_ϵ (where $|\alpha(\theta,\nu;X,W') - \alpha_I(\theta,\nu;X)| \leq \epsilon$) has probability greater than $1-\epsilon$. Write I_{2,E_ϵ} for the integral restricted to this set, and I_{2,E_ϵ^c} for that over the complement, so that $I_2 = I_{2,E_\epsilon} + I_{2,E_\epsilon^c}$. Then for $\theta > \theta_\epsilon$,

$$\begin{aligned}
I_{2,E_\epsilon}(W,\theta,\vartheta) &= \int_{E_\epsilon} d\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)(1-\alpha(\theta,\nu;W',X)) \\
&\leq \int_{E_\epsilon} d\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)[1-(\alpha_I(\theta,\nu;X)-\epsilon)] \\
&\leq \int d\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)[1-(\alpha_I(\theta,\nu;X)-\epsilon)] \\
&\leq (1+\epsilon) - \int q(\nu|\theta)\alpha_I(\theta,\nu;X)d\nu, \quad \text{and} \\
I_{2,E_\epsilon^c}(W,\theta,\vartheta) &= \int_{E_\epsilon^c} d\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X)(1-\alpha(\theta,\nu;W',X)) \\
&\leq \int_{E_\epsilon^c} d\nu dW' q(\nu|\theta)P(W'|W,\theta,\vartheta,\nu,X) \leq \epsilon.
\end{aligned}$$

We similarly divide the integral I_1 into two parts, I_{1,E_ϵ} (over E_ϵ) and I_{1,E_ϵ^c} (over its complement E_ϵ^c). For $\|\theta\|$ large enough, we can bound the acceptance probability by $\alpha_I(\theta,\theta';X)+\epsilon$ on the set E_ϵ, and by corollary 7.2.1, we get

$$I_{1,E_\epsilon} \leq \int_{E_\epsilon} \Omega(\theta')q(\theta'|\theta)(\alpha_I(\theta,\theta';X)+\epsilon)d\theta' \leq \int \Omega(\theta')q(\theta'|\theta)\alpha_I(\theta,\theta';X)d\theta' + \eta_1\epsilon\Omega(\theta).$$

For I_{1,E_ϵ^c}, from assumption 7.2.6, we have $\int_\Theta \Omega(\nu)^2 q(\nu|\theta)d\nu \leq \eta_0 \Omega(\theta)^2$ for $\|\theta\| > \theta_2$. So, by Cauchy-Schwarz inequality and bounding the acceptance probability by one, we have

$$\begin{aligned}\left(I_{1,E_\epsilon^c}\right)^2 &\leq \int_{E_\epsilon^c} q(\theta'|\theta)P(W'|W,\theta,\vartheta,\theta',X)d\theta' dW' \int_{E_\epsilon^c} \Omega(\theta')^2 q(\theta'|\theta)P(W'|W,\theta,\vartheta,\theta',X)d\theta' dW' \\ &\leq \epsilon \int \Omega(\theta')^2 q(\theta'|\theta)d\theta' \leq \epsilon\eta_0\Omega(\theta)^2,\end{aligned}$$

giving $I_{1,E_\epsilon^c} \leq \sqrt{\epsilon\eta_0}\Omega(\theta)$. Putting these four results together, for θ satisfying $\|\theta\| > \max(\theta_2, \theta_\epsilon, M)$ (where M is from Assumption 7.2.4 on the ideal sampler), we have

$$\begin{aligned}\int \Omega(\theta')P(d\theta'|W,\theta,\vartheta) &\leq \int \Omega(\theta')q(\theta'|\theta)\alpha_I(\theta,\theta'|X)d\theta' + \Omega(\theta)\int q(\nu|\theta)(1-\alpha_I(\theta,\nu|X))d\nu + \\ &\quad \sqrt{\eta_0\epsilon}\Omega(\theta) + \eta_1\epsilon\Omega(\theta) + 2\epsilon\Omega(\theta) \\ &\leq (1-\rho)\Omega(\theta) + (\sqrt{\eta_0\epsilon} + \eta_1\epsilon + 2\epsilon)\Omega(\theta) + L_I, \quad \text{giving}\end{aligned}$$

$$\begin{aligned}\mathbb{E}[\lambda_1|W'| + \Omega(\theta')|W,\theta,\vartheta,X] &\leq \lambda_1(1-\delta_1)|W| + \lambda_1 t_{end}(1+\eta_1)\Omega(\theta) + \\ &\quad (1-\rho)\Omega(\theta) + (\sqrt{\eta_0}\sqrt{\epsilon} + \eta_1\epsilon + 2\epsilon)\Omega(\theta) + L_I \\ &= (1-\delta_1)\lambda_1|W| + [1 - (\rho - \lambda_1 t_{end}(1+\eta_1) - (2+\eta_1)\epsilon - \sqrt{\eta_0\epsilon})]\Omega(\theta) + L_I \\ &\stackrel{\text{def}}{=} (1-\delta_1)\lambda_1|W| + (1-\delta_2)\Omega(\theta) + L_I\end{aligned}$$

For (λ_1, ϵ) small enough, $\delta_2 \in (0,1)$, and $\delta = \min(\delta_1, \delta_2)$ gives the drift condition. ■

8. VARIATIONAL BAYESIAN INFERENCE

8.1 Introduction

So far, we have focused on Monte Carlo sampling methods for posterior inference. In this chapter, referring to Blei et al. (2017), we review variational Bayesian (VB) inference. Variational Bayesian inference is an alternative to MCMC, that has grown popular in statistics and machine learning. Often, MCMC can suffer from slow convergence to the posterior distribution as well as the high auto-correlation of the consecutive samples. Variaional inference is an alternative approach approximating the intractable posterior distribution with some simple probability distributions. The convergence of variational inference is usually fast. The disadvantage is that, unlike MCMC, they are biased. Nevertheless, this offers an additional computational tool for practitioners.

Consider a joint density of latent variables $z = z_{1:m}$ and observations $x = x_{1:n}$,

$$p(z, x) = p(z)p(x|z).$$

By constructing a Markov chain whose stationary distribution is the posterior density $p(z|x)$, MCMC methods can draw samples from the posterior distribution. Unlike MCMC, the main idea of variational Bayesian inference is to approximate $p(z|x)$ with an element $q(z)$ of a simple family of distribution $\mathcal{L}$. VB finds the approximate distribution which minimizes the Kullback-Leibler (KL) divergence, from the family of approximate distributions $\mathcal{L}$.

$$q^*(z) = \underset{q(z)\in\mathcal{L}}{\operatorname{argmin}} \operatorname{KL}(q(z)||p(z|x)).$$

Recall the KL divergence between two probability distributions $p_1(x)$ and $p_2(x)$ is

$$kl(p_1||p_2) = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx.$$

MCMC algorithms approximate the posterior distribution by sampling while variational inference algorithms approximate the posterior distribution by solving an optimization problem.

8.2 The evidence lower bound

As stated in the previous section, our goal is to find the approximate distribution in $\mathcal{L}$, closest to the posterior distribution, in the KL divergence sense. The expression for KL divergence for variational inference can be expanded to give

$$\begin{aligned}
\mathrm{KL}(q(z)||p(z|x)) &= \int q(z) \log \frac{q(z)}{p(z|x)} dz \\
&= -\left(\int q(z) \log \frac{p(x,z)}{q(z)} dz - \int q(z) \log p(x) dz \right) \\
&= -\int q(z) \log \frac{p(x,z)}{q(z)} dz + \log p(x) \\
&= -(\mathbb{E}_q \log p(z,x) - \mathbb{E}_q \log p(z)) + \log p(x)
\end{aligned}$$

In many cases, $p(x)$ is not easy to compute. Call $\mathbb{E}_q \log p(z,x) - \mathbb{E}_q \log p(z)$ as the evidence lower bound (ELBO) and rearrange the above equation.

$$\begin{aligned}
\mathrm{ELBO}(q) &= (\mathbb{E}_q \log p(z,x) - \mathbb{E}_q \log p(z)) \\
&= \log p(x) - \mathrm{KL}(q(z)||p(z|x)).
\end{aligned}$$

Since ELBO is equivalent to the negative KL divergence up to a constant, maximizing the ELBO is equivalent to minimizing the KL divergence. Notice that KL divergence $\mathrm{KL}(q(z)||p(z|x)) \geq 0$. Thus we have

$$\mathrm{ELBO}(q) \leq \log p(x).$$

8.3 Mean field variational inference

In this section, following Bishop (2006), we introduce mean-field variational Bayesian inference. For any $q(z) \in \mathcal{L}$, restrict $\mathcal{L}$ to the set of q's satisfying

$$q(z_{1:m}) = \prod_{i=1}^{m} q_i(z_i).$$

This means that under a variational approximation, each variable z_i is independent. In order to maximize the ELBO, coordinate ascent algorithm can be used to find the optimal q_i^* for $i = 1, \cdots, m$. This proceeds by optimizing each component q_i of q, one at a time. Let z_{-j} be $(z_1, \cdots, z_{j-1}, z_{j+1}, \cdots z_m)$, where the notation $-j$ denotes all indices except j. We consider jth variable z_j while fixing the other $q_k(z_k)$, $k \neq j$. Denote the optimal $q_j(z_j)$ as $q_j^*(z_j)$. We have

$$q_j^*(z_j) \propto \exp\left[\mathbb{E}_{-j} \log p(z_j, z_{-j}, x)\right].$$

To derive this, we need to write the ELBO ($\mathbb{E}_q \log p(z, x) - \mathbb{E}_q \log p(z)$) as a function of q_j while fixing all other q_k, $k \neq j$.

$$\begin{aligned} \text{ELBO} &= \mathbb{E}_j \left[\mathbb{E}_{-j} \log p(z_j, z_{-j}, x)\right] - \mathbb{E}_j \log q_j(z_j) + C; \\ &= -\operatorname{KL}(q_j(z_j) || \mathbb{E}_{-j} \log p(z_j, z_{-j}, x)) + C, \end{aligned}$$

where C is a constant. It is easy to see that the q_j^* maximize the ELBO.

8.4 Collapsed variational Bayesian inference

In this section, referring to Teh et al. (2006), we introduce the collapsed variational Bayesian (CVB) inference. Collapsed variational Bayesian algorithm leverages the idea of variational Bayesian inference and the collapsed Gibbs sampler (Liu, 1994a). It works in a collapsed space, where the parameters are marginalized out. Now, consider a model with latent variable variables $z = z_{1:m}$ and observations $x = x_{1:n}$, as well as the parameters θ. With the prior distribution of θ being $p(\theta)$, the joint probability density is $p(z, x, \theta) = p(x, z|\theta)p(\theta)$.

The goal of the collapsed Gibbs sampler is to draw latent variable samples from the conditional probability density $p(z|x)$, given the observations x. It proceeds by integrating out the parameter θ and only consider the marginal distribution

$$p(x, z) = \int p(x, z|\theta)p(\theta)d\theta. \tag{8.1}$$

Then it updates each component z_j of z given all the other components fixed, as well as the observations, from the conditional distribution $p(z_j|z_{-j}, x)$. It skips sampling θ from the conditional distribution, which is usually time consuming. Not having to simulate θ avoids coupling between θ and z and can speed up mixing.

Inspired by the collapsed Gibbs sampler, CVB models the dependence between the parameters and the latent variables in an exact fashion, instead of assuming independence. This means CVB does not make any assumptions on the structure of the the conditional distribution $q(\theta|z, x)$. CVB still assumes the mean-field structure, $q(z|x) \approx \prod_{t=1}^{m} q(z_t|x)$. Formally, CVB assumes the following approximation structure.

$$p(z, \theta|x) \approx q(\theta|z, x)q(z|x),$$
$$q(z|x) = \prod_{t=1}^{m} q(z_t|x).$$

By maximizing the ELBO, we can get the following updating rules.

$$q^*(\theta|z, x) = p(\theta|z, x), \text{which is the true posterior;}$$
$$q^*(z_j|x) \propto \exp(\mathbb{E}_{q(z_{-j})}[\log p(z_j|x, z_{-j})]).$$

Teh et al. (2006) shows that it is equivalent to directly working from the marginal distribution (equation 8.1) and then assuming the mean-field structure

$$p(z|x) \approx q(z|x) = \prod_{t=1}^{m} q(z_t|x).$$

9. COLLAPSED VARIATIONAL INFERENCE FOR MARKOV JUMP PROCESSES

9.1 Introduction

As described in earlier chapters, inference for Markov jump processes typically proceeds via Markov chain Monte Carlo, and can suffer from various computational challenges. In this chapter, we bring ideas from variational Bayes towards posterior inference for Markov jump processes, proposing a novel and efficient collapsed variational algorithm based on the idea of *uniformization*, which marginalizes out the MJP parameters, thereby addressing the issue of slow mixing. Our algorithm adaptively finds regions of low and high transition activity, optimizing the discretization of time, rather than integrating these out. We apply our ideas to synthetic data as well as a dataset of check-in recordings, where we show that these can bring significant computational benefits. This work is published in our paper Pan et al. (2017).

9.2 An alternate prior on the parameters of an MJP

We use uniformization to formulate a novel prior distribution over the parameters of an MJP; this will facilitate our later variational Bayes algorithm. Consider A_i, the ith row of the rate matrix A. This is specified by the diagonal element A_{ii}, and the vector $B_i := \frac{1}{|A_{ii}|}(A_{i1}, \cdots, A_{i,i-1}, 0, A_{i,i+1}, \cdots, A_{iN})$. Recall that the latter is a probability vector, giving the probability of the next state after i. In Fearnhead and Sherlock (2006), the authors place a Gamma prior on $|A_{ii}|$, and what is effectively, a Dirichlet$(\alpha, \cdots, 0, \cdots, \alpha)$ prior on B_i (although they treat B_i as an $N-1$-component vector by ignoring the 0 at position i).

We place a Dirichlet$(a, \cdots, a_0, \cdots, a)$ prior on B_i for all i. Such B_i's allow self-transitions, and form the rows of the transition matrix B from uniformization. Note that under uniformization, the row A_i is uniquely specified by the pair (Ω, B_i) via the relationship $A_i = \Omega(B_i - 1_i)$, where 1_i is the indicator for i. We complete our specification by placing a Gamma prior over Ω.

Note that since the rows of A sum to 0, and the rows of B sum to 1, both matrices are completely determined by $N(N-1)$ elements. On the other hand, our specification has $N(N-1)+1$ random variables, the additional term arising because of the prior over Ω. Given A, Ω plays no role in the generative process defined by Gillespie's algorithm, although it is an important parameter in MCMC inference algorithms based on uniformization. In our situation, B represents transition probabilities *conditioned on there being a transition*, and now Ω does carry information about A, namely the distribution over event times. Later, we will look at the implied marginal distribution over A. First however, we consider the generalized uniformization scheme of Rao and Teh (2012). Here we have N additional parameters, Ω_1 to Ω_N. Again, under our model, we place Gamma priors over these Ω_i's, and Dirichlet priors on the rows of the transition matrix B.

Note that in Rao and Teh (2013, 2012), Ω is set to $2 \max_i |A_{ii}|$. From the identity $B = I + \frac{1}{\Omega} A$, it follows that under any prior over A, with probability 1, the smallest diagonal element of B is $1/2$. Our specification avoids such a constrained prior over B, instead introducing an additional random variable Ω. Indeed, our approach is equivalent to a prior over (Ω, A), with $\Omega = k \max_i A_{ii}$ for some *random* k. We emphasize that the choice of this prior over k does not effect the generative model, only the induced inference algorithms such as Rao and Teh (2013) or our proposed algorithm.

To better understand the implied marginal distribution over A, consider the representation of Rao and Teh (2012), with independent Gamma priors over the Ω_i's. We have the following result:

Proposition 9.2.1 *Place independent Dirichlet priors on the rows of B as above, and independent Gamma$((N-1)a + a_0, b)$ priors on the Ω_i. Then, the associated matrix A has off-diagonal elements that are marginally Gamma(a, b)-distributed, and negative-diagonal elements that are marginally Gamma$((N-1)a, b)$-distributed, with the rows of A adding to 0 almost surely.*

The proposition is a simple consequence of the Gamma-Dirichlet calculus: first observe that the collection of variables $\Omega_i B_{ij}$ is a vector of independent Gamma(a, b) variables. Noting that $A_{ij} = \Omega_i B_{ij}$, we have that the off-diagonal elements of A are independent Gamma(a, b)s, for $i \neq j$. Our proof is complete when we notice that the rows of A sum to 0, and that the sum of independent Gamma variables is still Gamma-distributed, with scale parameter equal to the sum of the scales. It is also easy to see that given A, the Ω_i is set by $\Omega_i = |A_{ii}| + \omega_i$, where $\omega_i \sim \text{Gamma}(a_0, b)$.

In this work, we will simply matters by scaling all rows by a single, shared Ω. This will result in a vector of A_{ij}'s each marginally distributed as a Gamma variable, but now positively correlated due to the common Ω. We will see that this simplification does not affect the accuracy of our method. In fact, as our variational algorithm will maintain just a point estimate for Ω, so that its effect on the correlation between the A_{ii}'s is negligible.

9.3 Collapsed variational inference for MJPs

Given noisy observations X of an MJP, we are interested in the posterior $p(S(t), A|X)$. Following the earlier section, we choose an augmented representation, where we replace A with the pair (B, Ω). Similarly, we represent the MJP trajectory $S(t)$ with the pair (v_0, V, W), where W is the set of candidate transition times, and V (with $|W| = |V|$), is the set of states at these times. For our variational algorithm, we will integrate out the Markov transition matrix B, working instead with the marginal distribution $p(W, V, \Omega)$. Such a collapsed representation avoids issues that plague MCMC and VB approaches, where coupling between trajectory and transition ma-

trix slows down mixing/convergence. The distribution $p(W, V, \Omega)$ is still intractable however, and as is typical in variational algorithms, we will make a factorial approximation $p(W, V, \Omega) \approx q(W, V)q(\Omega)$. Writing $q(W, V) = q(V|W)q(W)$, we shall also restrict $q(W)$ to a delta-function: $q(W) = \delta_{\hat{W}}(W)$ for some $\hat{W}$. In this way, finding the 'best' approximating $q(W)$ within this class amounts to finding a 'best' discretization of time. This approach of optimizing over a time-discretization is in contrast to MCMC schemes that integrate out the time discretization, and has a two advantages: *Simplified computation*: Searching over time-discretization can be significantly more efficient than integrating it out. This is especially true when a trajectory involves bursts of transitions interspersed with long periods of inactivity, where schemes like Rao and Teh (2013) can be quite inefficient.

Better interpretability: A number of applications use MJPs as tools to segment a time interval into inhomogeneous segments. A full distribution over such an object can be hard to deal with.

Following work on variational inference for discrete-time Markov chains (Wang and Blunsom, 2013), we will approximate $q(V|W)$ factorially as $q(V|W) = \prod_{t=1}^{|W|} q(v_t)$. Finally, since we fix $q(W)$ to a delta function, we will also restrict $q(\Omega)$ to a delta function, only representing uncertainty in the MJP parameters via the marginalized transition matrix B.

We emphasize that even though we optimize over time discretizations, we still maintain posterior uncertainty of the MJP state. Thus our variational approximation represents a distribution over piecewise-constant trajectories as a single discretization of time, with a probability vector over states for each time segment (Figure 9.3). Such an approximation does not involve too much loss of information, while being more convenient than a full distribution over trajectories, or a set of sample paths. While optimizing over trajectories, our algorithm attempts to find segments where the distribution over states is reasonably constant, if not it will refine a segment into two smaller ones. Our overall variational inference algorithm then involves minimizing

the Kullback-Liebler distance between this posterior approximation and the true posterior. We do this in a coordinate-wise manner:

1) Updating $q(V|W) = \prod_{t=1}^{|W|} q(v_t)$: Given a discretization W, and an Ω, uniformization tells us that inference over V is just inference for a discrete-time hidden Markov model. We adapt the approach of Wang and Blunsom (2013) to update $q(V)$. Assume the observations X follow an exponential family likelihood with parameter C_s for state s: $p(x_t^l|S_t = s) = \exp(\phi(x_t^l)^T C_s)h(x_t^l)/Z(C_s)$, where Z is the normalization constant, and x_t^l is the l-th observation observed in between $[W_t, W_{t+1})$. Then for a sequence of $|W|$ observations, we have

$$\begin{aligned} p(X, V|B, C) &\propto \prod_{t=0}^{|W|} B_{v_t, v_{t+1}} \prod_{l=1}^{n_t} \exp(\phi(x_t^l)^T C_{v_t}) h(x_t^l)/Z(c_{v_t}) \\ &= \left[\prod_{i=1}^{S}\prod_{j=1}^{S} B_{ij}^{\#_{ij}}\right] \prod_{i=1}^{S} \exp(\sum_{t=0}^{|T|} \tilde{\phi}_t^T C_i \mathbb{I}_{\{v_t=i\}}) (\prod_{t=0}^{|T|}\prod_{l=1}^{n_t} \frac{h(x_t^l)}{Z(C_{v_t})}) \\ &= \left[\prod_{i=1}^{S}\prod_{j=1}^{S} B_{ij}^{\#_{ij}}\right] \prod_{i=1}^{S} \exp(\bar{\phi}_i^T C_i) (\prod_{t=0}^{|T|}\prod_{l=1}^{n_t} \frac{h(x_t^l)}{Z(C_{v_t})}) \end{aligned}$$

Here n_t is the number of observations in $[W_t, W_{t+1})$ and $\#_{ij}$ is the number of transitions from state i to j, and $\tilde{\phi}_t = \sum_{l=1}^{n_t} \phi(x_t^l)$ and $\bar{\phi}_i = \sum_{t,s.t.\ u_t=i} \tilde{\phi}_t$.

Placing Dirichlet(α) priors on the rows of B, and an appropriate conjugate prior on C, which depends on β, we have

$$p(X, V, B, C) \propto= \left[\prod_{i=1}^{S} \Gamma(S\alpha) \prod_{j=1}^{S} \frac{B_{ij}^{\#_{ij}+\alpha-1}}{\Gamma(\alpha)}\right] \prod_{i=1}^{S} \exp(C_i^T(\bar{\phi}_i + \beta)) (\prod_{t=0}^{|W|}\prod_{l=1}^{n_t} \frac{h(x_t^l)}{Z(C_{v_t})}).$$

Integrating out B and C, and writing $\#_i$ for the number of visits to state i, we have:

$$p(X, V) \propto= \left[\prod_{i=1}^{S} \frac{\Gamma(S\alpha)}{\Gamma(\#_i + \alpha)} \prod_{j=1}^{S} \frac{\Gamma(\#_{ij} + \alpha)}{\Gamma(\alpha)}\right] \prod_{i=1}^{S} \bar{Z}_i(\bar{\phi}_i + \beta).$$

$$\text{Then, } p(v_t = k|\cdot) \propto \frac{(\#_{v_{t-1},k}^{\neg t} + \alpha)^{\delta_k^t} (\#_{k,v_{t-1}}^{\neg t} + \delta_k^{t-1,t+1} + \alpha)^{\delta_k^t}}{(\#_k^{\neg t} + \alpha)^{\delta_k^t}} \cdot \bar{Z}_k(\bar{\phi}_k^{\neg t} + \bar{\phi}_k(X_t) + \beta)$$

Standard calculations for variational inference give the solution to

$$q(v_t) = \text{argmin } \text{KL}(q(V, W, \Omega) \| p(V, W, \Omega|X))$$

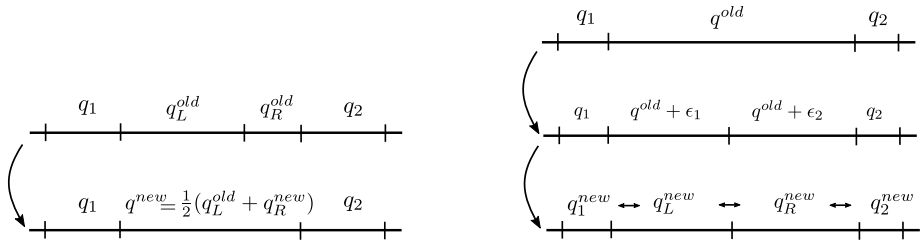

Figure 9.1. (left) Merging to time segments. (right) splitting a time segment. Horizontal arrows are VB messages.

as $q(v_t) = E_{q^{\neg t}}[\log p(v_t|\cdot)]$, We then have the update rule:

$$q(v_t = k) \propto \frac{\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{v_{t-1},k} + \alpha]\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{k,v_{t-1}} + \delta_k^{t-1,t+1} + \alpha]}{\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{v_{t-1},k} + S\alpha]\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{k,v_{t-1}} + \delta_k^{t-1,t+1} + \alpha]} \cdot \frac{\mathbb{E}_{q^{\neg t}}\bar{Z}_k(\bar{\phi}_k^{\neg t} + \bar{\phi}_k(X_t) + \beta)}{\mathbb{E}_{q^{\neg t}}\bar{Z}_k(\bar{\phi}_k^{\neg t} + \beta)}$$

For the special case of multinomial observations, we refer to Wang and Blunsom (2013).

2) Updating $q(W)$: We perform a greedy search over the space of time-discretizations by making local stochastic updates to the current W. Every iteration, we first scan the current W to find a beneficial *merge* (Figure 9.3, left): go through the transition times in sequential or random order, merge with the next time interval, compute the variational lower bound under this discretization, and accept if it results in an improvement. This eliminates unnecessary transitions times, reducing fragmentation of the segmentation, and the complexity of the learnt model. Calculating the variational bound for the new time requires merging the probability vectors associated with the two time segments into a new one. One approach is to initialize this vector to some arbitrary quantity, run step 1 till the q's converge, and use the updated variational bound to accept or reject this proposal. Rather than taking this time-consuming approach, we found it adequate to set the new q to a convex combination to the old q's, each weighted by the length of their corresponding interval length. In our experiments, we found that this performed comparably at a much lower computational cost.

If no merge is found, we then try to find a beneficial split. Again, go through the time segments in some order, now splitting each interval into two. After each split, compare the likelihood before and after the split, and accept (and return) if the improvement exceeds a threshold. Again, such a split requires computing probability vectors for the newly created segments. Now, we assign each segment the same vector as the original segment (plus some noise to break symmetry). We then run one pass of step 1, updating the q's on either side of the new segment, and then updating the q's in the two segments. We consider two interval splitting schemes, bisection and random-splitting.

Overall, our approach is related to split-merge approaches for variational inference in nonparametric Bayesian models (Hughes et al., 2015); these too maintain and optimize point estimates of complex, combinatorial objects, instead maintaining uncertainty over quantities like cluster assignment. In our real-world check-in applications, we consider a situation where there is not just one MJP trajectory, but a number of trajectories corresponding to different users. In this situation, we take a stochastic variational Bayes approach, picking a random user and following the steps outlined earlier.

Updating $q(\Omega)$: With a Gamma(a_1, a_2) prior over Ω, the posterior over Ω is also Gamma, and we could set Ω to the MAP. We found this greedy approach unstable sometimes, instead using a partial update, with the new Ω equal to the mean of the old value and the MAP value. Writing s for the total number of transition times in all m trajectories, this gives us $\Omega_{new} = (\Omega_{curr} + (a_1 + s)/(a_2 + m))/2$.

Algorithm 8 Collapsed variational inference for Markov jump processes

Input: Noisy observations X and prior parameters α, β, a_1, a_2.
Current variational approximation $\prod_{t=1}^{|W_{curr}|} q_{curr}(v_t)\delta_{W_{curr}}(W)\delta_{\Omega_{curr}}(\Omega)$.

Output: New variational approximation $\prod_{t=1}^{|W_{new}|} q_{new}(v_t)\delta_{W_{new}}(W)\delta_{\Omega_{new}}(\Omega)$.

1: **Update $q(V|W)$ given the current grid W_{curr}**:

$$q(v_t = k) \propto \frac{\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{v_{t-1},k} + \alpha]\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{k,v_{t-1}} + \delta_k^{t-1,t+1} + \alpha]}{\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{v_{t-1},k} + S\alpha]\mathbb{E}_{q^{\neg t}}[\#^{\neg t}_{k,v_{t-1}} + \delta_k^{t-1,t+1} + \alpha]} \cdot \frac{\mathbb{E}_{q^{\neg t}}\bar{Z}_k(\bar{\phi}_k^{\neg t} + \bar{\phi}_k(X_t) + \beta)}{\mathbb{E}_{q^{\neg t}}\bar{Z}_k(\bar{\phi}_k^{\neg t} + \beta)}$$

2: **Update $q(W)$**: Perform a greedy search over the space of time-discretizations by making local stochastic updates to the current grid W_{curr} and update $q(V|W)$ accordingly, resulting in $q_{new}(W) = \delta_{W_{new}}(W)$ and $q_{new}(V|W) = \prod_{t=1}^{|W_{new}|} q_{new}(v_t)$.

3: **Update $q(\Omega)$**: Writing s for the total number of transition times in all m trajectories, set $\Omega_{new} = (\Omega_{old} + (a_1 + s)/(a_2 + m))/2$. $q_{new}(\Omega) = \delta_{\Omega_{new}}(\Omega)$.

9.4 Experiments

We present qualitative and quantitative experiments using synthetic and real datasets to demonstrate the accuracy and efficiency of our variational Bayes (VB) algorithm. We mostly focus on comparisons with the MCMC algorithms from Rao and Teh (2013) and Rao and Teh (2012).

Datasets. We use a dataset of check-in sequences from 8967 FourSquare users in the year 2011, originally collected by Gao et al. (2012) for studying location-based social networks. Each check-in has a time stamp and a location (latitude and longitude), with users having 191 check-in records on average. We only consider check-ins inside a rectangle containing the United States and parts of Mexico and Canada (see Figure 9.4, left), and randomly select 200 such sequences for our experiments. We partition the space into a 40×40 grid, and define the observation distribution of each MJP state as a categorical distribution over the grid cells. See Pan et al. (2016) for more details on this application.

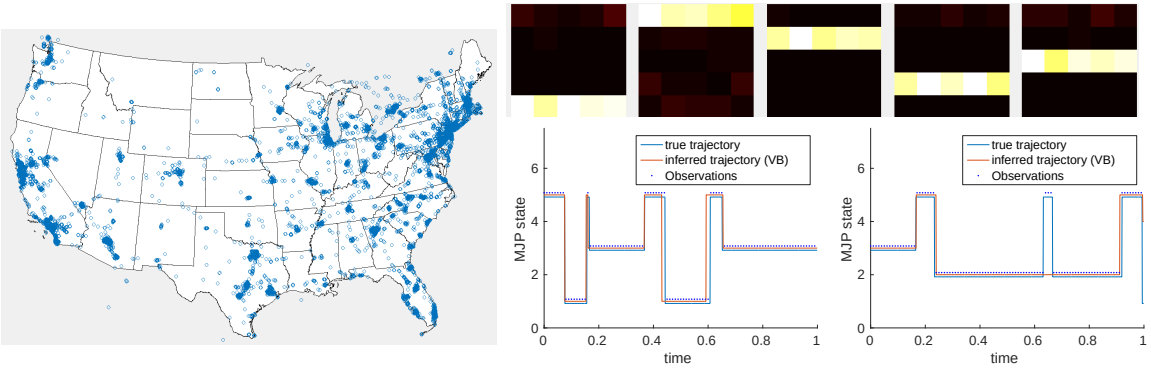

Figure 9.2. (left) check-ins of 500 users. (right-top) heatmap of emission matrices; (right-bottom) true and inferred trajectories: the y-values are perturbed for clarity.

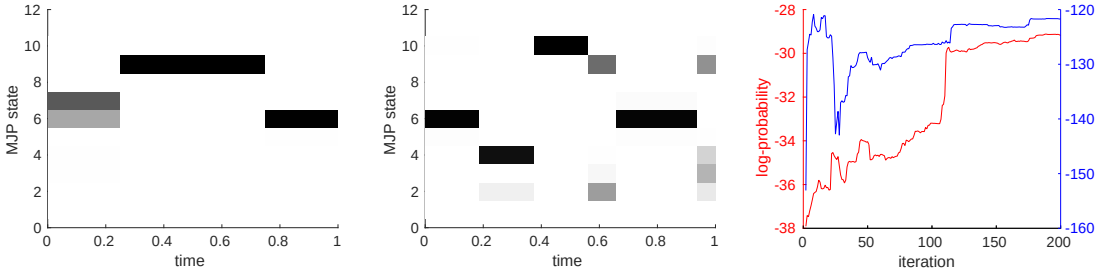

Figure 9.3. (left,middle) posterior distribution over states of two trajectories in second synthetic dataset; (right) evolution of $\log p(W \mid \Omega, X)$ in the VB algorithm for two sample sequences

We also use two synthetic datasets in our experiments, with observations in a 5×5 grid. For the first dataset, we fix $\Omega = 20$ and construct a transition matrix B for 5 states with $B(i,i) = 0.8$, $B(i,5) = 0.19$, $B(5,5) = 0$, and $B(5,i) = 0.25$ for $i \in [1,4]$. By construction, these sequences can contain many short time intervals at state 5, and longer time intervals at other states. We set the observation distribution of state i to have 0.2 probability on grid cells in the i-th row and 0 probability otherwise. For the second synthetic dataset, we use 10 states and draw both the transition probabilities

of B and the observation probabilities from Dirichlet(1) distribution. Given (Ω, B), we sample 50 sequences, each containing 100 evenly spaced observations.

Hyperparameters: For VB on synthetic datasets we place a Gamma(20, 2) prior on Ω, and Dirichlet(2) priors on the transition probabilities and the observation probabilities, while on the check-in data, a Gamma(6, 1), a Dirichlet(0.1) and a Dirichlet(0.01) are placed. For MCMC on synthetic datasets, we place a Gamma(2, 0.2) and a Dirichlet(0.1) for the rate matrix, while on the check-in data, a Gamma(1, 1) and a Dirichlet(0.1) are placed.

Visualization: We run VB on the first synthetic dataset for 200 iterations, after which we use the posterior expected counts of observations in each state to infer the output emission probabilities (see Figure 9.4(top-right)). We then relabel the states under the posterior to best match the true state (our likelihood is invariant to state labels); Figure 9.4(bottom-right) shows the true and MAP MJP trajectories of two sample sequences in the synthetic dataset. Our VB algorithm recovers the trajectories well, although it is possible to miss some short "bumps". MCMC also performs well in this case, although as we will show, it is significantly more expensive.

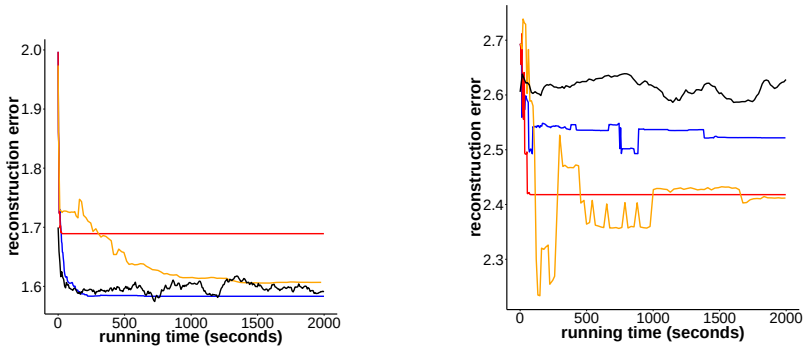

Figure 9.4. reconstruction error of MCMC and VB (using random and even splitting) for the (left) first and (right) the second synthetic dataset. The random split scheme is in blue , even split scheme is in red , and VB random split scheme with true omega in orange. MCMC is in black.

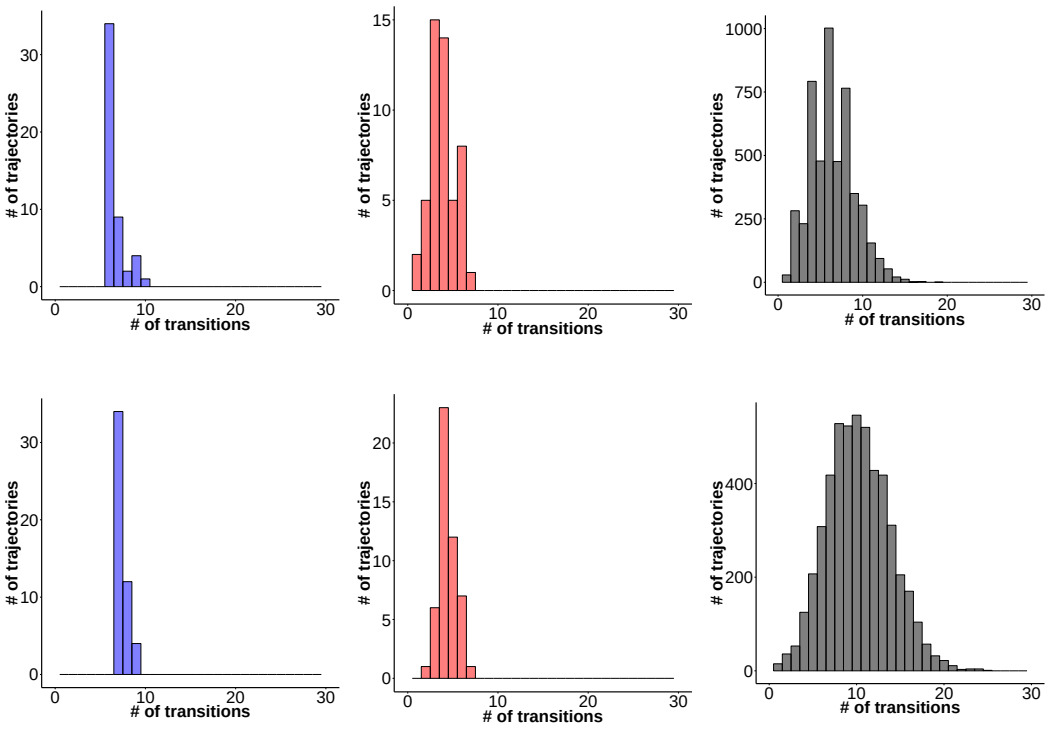

Figure 9.5. Synthetic dataset 1(top) and 2(bottom): Histogram of number of transitions using VB with (left) random splitting; (middle) even spliting; (right) using MCMC.

The inferred posteriors of trajectories have more uncertainty for the second synthetic dataset. Figure 9.3 (left and middle) visualizes the posterior distributions of two hidden trajectories with darker regions for higher probabilities. The ability to maintain posterior uncertainty about the trajectory information is important in real world applications, and is something that k-means-style approximate inference algorithms (Huggins et al., 2015) ignore.

Inferred trajectories for real-world data. We run the VB algorithm on the check-in data using 50 states for 200 iterations. Modeling such data with MJPs will recover MJP states corresponding to cities or areas of dense population/high check-in activity. We investigate several aspects about the MJP trajectories inferred by the

algorithm. Figure 9.3(right) shows the evolution of $\log p(W \mid \Omega, X)$ (up to constant factor) of two sample trajectories. This value is used to determine whether a merge or split is beneficial in our VB algorithm. It has an increasing trend for most sequences in the dataset, but can sometimes decrease as the trajectory discretization evolves. This is expected, since our stochastic algorithm maintains a pseudo-bound. Figure 9.5 shows similar results for the synthetic datasets.

Normally, we expect a user to switch areas of check-in activity only a few times in a year. Indeed, Figure 9.6 (left) shows the histogram of the number of transition times across all trajectories, and the majority of trajectories have 3 or less transitions. We also plot the actual transition times of 10 random trajectories (right). In contrast, MCMC tends to produce more transitions, many of which are redundant. This is a side effect of uniformization in MCMC sampling, which requires a homogeneously dense Poisson distributed trajectory discretization at every iteration.

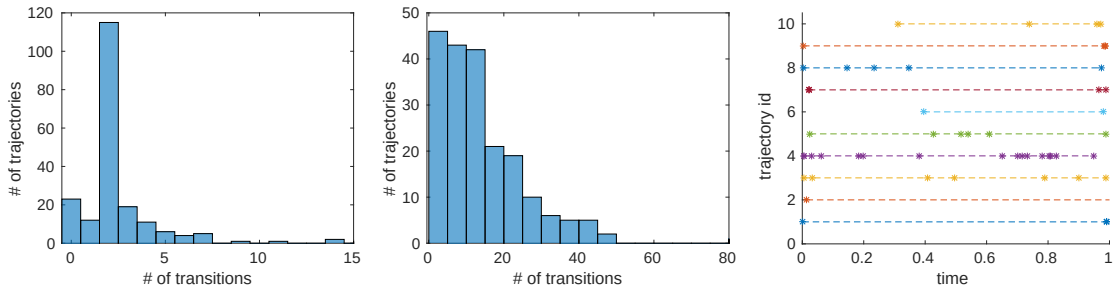

Figure 9.6. histogram of number of transitions using (left) VB and (middle) MCMC; (right) transition times of 10 users using VB

Running time vs. reconstruction error. We measure the quality of the inferred posterior distributions of trajectories using a reconstruction task on the check-in data. We randomly select 100 test sequences, and randomly hold out half of the observations in each test sequence. The training data consists of the observations that are not held out, i.e., 100 full sequences and 100 half sequences. We run our VB algorithm on this training data for 200 iterations. After each iteration, we reconstruct the held-out observations as follows: given a held-out observation at time

t on test sequence τ, using the maximum-likelihood grid cell to represent each state, we compute the expected grid distance between the true and predicted observations using the estimated posterior $q(v_t)$. The reconstruction error for τ is computed by averaging the grid distances over all held-out observations in τ. The overall reconstruction error is the average reconstruction error over all test sequences. Similarly, we run the MCMC algorithm on the training data for 1000 iterations, and compute the overall reconstruction error after every 10 iterations, using the last 300 iterations to approximate the posterior distribution of the MJP trajectories. We also run an improved variant of the MCMC algorithm, where we use the generalized uniformization scheme (Rao and Teh, 2012) with different Ω_i for each state. This allows coarser discretizations for some states and typically runs faster per iteration.

Figure 9.7(left) shows the evolution of reconstruction error during the algorithms. The error using VB plateaus much more quickly than the MCMC algorithms. The error gap between MCMC and VB is because of slow mixing of the paths and parameters, as a result of the coupling between latent states and observations as well as modeling approximations. Although the improved MCMC takes less time per iteration, it is not more effective for reconstruction in this experiment. Figure 9.4 shows similar results for the synthetic datasets. Figure 9.8 visualizes the posterior distributions of three hidden trajectories with darker shades for higher probabilities.

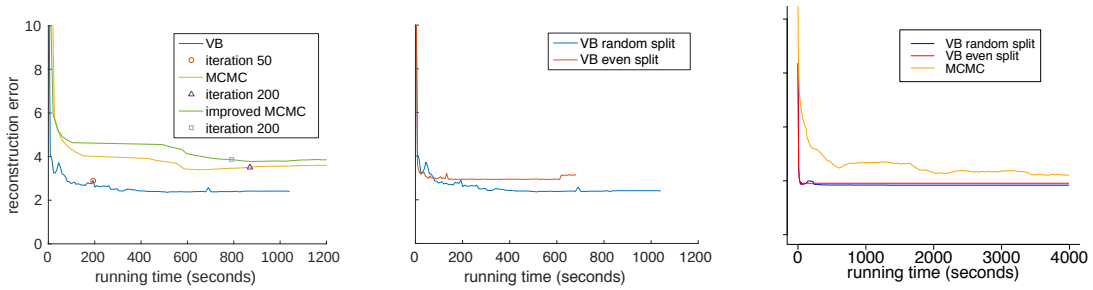

Figure 9.7. (left) reconstruction error of VB and MCMC algorithms; (middle) reconstruction error using random and even splitting; (right) reconstruction error for more iterations

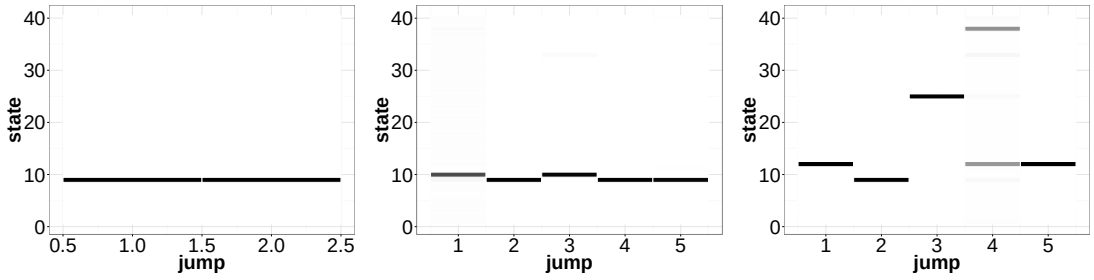

Figure 9.8. Posterior distribution over states of three trajectories in checkin dataset.

We have chosen to split each time interval randomly in our VB algorithm. Another possibility is to simply split it evenly. Figure 9.7(middle) compares the reconstruction error of the two splitting schemes. Random splitting has lower error since it produces more successful splits; on the other hand, the running time is smaller with even splitting due to fewer transitions in the inferred trajectories. In Figure 9.7(right), we resampled the training set and the testing set and ran the experiment for longer. It shows that the error gap between VB and MCMC is closing.

10. SUMMARY AND FUTURE WORK

10.1 Summary

In this thesis, we described a novel symmetrized Metropolis Hastings algorithm for the parameter inference for Markov jump processes. We marginalized out the state information after using uniformization to sample the candidate transition times. We computed the MH acceptance rate by FFBS algorithm. The symmetrization proposing scheme avoids the dependency between candidate transition times and the MJP parameters, leading to a efficient Metropolis Hastings sampling algorithm for the parameter inference. We also show our sampler inherits geometric ergodicity property from an ideal sampler that is computationally much more expensive.

We also described a novel collapsed variational inference algorithm for Bayesian inference in Markov jump processes. We reparameterized the Markov jump processes based the idea of uniformization. Our variational inference algorithm marginalizes out the MJP parameters, and maintains a point estimate of the discretization of time, which leads to computational savings, and thus is more efficient.

10.2 Future work

For the problem of parameter inference for Markov jump processes, there are a number of interesting directions for future research. Our focus was on Metropolis-Hastings algorithms for typical settings, where the parameters are low dimensional. It is interesting to investigate how our ideas extend to schemes like Hamiltonian Monte Carlo (Neal, 2010) suited for higher-dimensional settings. Another direction is to develop and study similar schemes for more complicated hierarchical models like mixtures of MJPs or coupled MJPs. While we focused only on Markov jump

processes, it is also of interest to study similar ideas for algorithms for more general processes (Rao and Teh, 2012). Also, it is interesting to apply the idea of the generalized uniformization. We can extend our method by treating the uniformization rate $\Omega(\theta)$ as a trajectory dependent random variable. Moreover, in this thesis, we assume the state space of the MJP is finite, which allows us to apply the FFBS algorithm. However, if the state space of the MJP is infinite, which is common in practice. For example, the queuing models without capacity have infinite state space $\mathbb{N} = \{0, 1, \cdots\}$. Such an infinite state space prevents us to use the FFBS algorithm. Fortunately, in practice, it is possible to choose a large number N and work on the trimmed state space $\{0, 1, \cdots N\}$. However, it will introduce bias and the sampler is not exact anymore. It is important to study if we can extend our methods to the MJPs with infinite state space without any approximation. It is also important to investigate how similar ideas apply to deterministic algorithms like variational Bayes (Opper and Sanguinetti, 2007; Pan et al., 2017). From a theoretical viewpoint, our proof required the uniformization rate to satisfy $\Omega(\theta) \geq k_1 \max_s A_s(\theta) + k_0$ for $k_1 > 1$. We believe our result still holds for $k_1 = 1$, and for completeness, it would be interesting to prove this.

For the problem of variational inference for Markov jump processes, there are a number of interesting extensions worth studying. First is to consider more structured variational approximations (Wang and Blunsom, 2013), than the factorial approximations we considered here. Also of interest are extensions to more complex MJPs, with infinite state-spaces (Saeedi and Bouchard-Côté, 2011) or structured state-spaces (Opper and Sanguinetti, 2007). It is also interesting to look at different extensions of the schemes we proposed in this paper: different choices of split-merge proposals, and more complicated posterior approximations of the parameter Ω. Finally, it is instructive to use other real-world datasets to compare our approaches with more traditional MCMC approaches.

REFERENCES

REFERENCES

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725, 2009.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

S. Asmussen. *Applied Probability and Queues*, volume 51. Springer, 2003.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

L. Breuer. *From Markov jump processes to spatial queues*. Springer, 2003.

C. K. Carter and R. Kohn. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83(3):589–601, 1996.

E. Çinlar. *Introduction to Stochastic Processes*. Prentice Hall, 1975.

T. El-Hay, N. Friedman, and R. Kupferman. Gibbs sampling in factorized continuous-time Markov processes. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 169–178, 2008.

R. Elliott and C. J. Osakwe. Option pricing for pure jump processes with Markov switching compensators. *Finance and Stochastics*, 10(2):250–275, 2006.

P. Fearnhead and C. Sherlock. An exact Gibbs sampler for the Markov-modulated Poisson process. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(5):767–784, 2006.

J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *The Annals of Applied Probability*, 1(1):62–87, 1991.

Sylvia Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, 1994.

H. Gao, J. Tang, and H. Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM conference on Information and knowledge management*. ACM, 2012.

D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular biology and evolution*, 11(5):725–736, 1994.

D. Gross, J.F. Shortle, J.M. Thompson, and C.M. Harris. *Fundamentals of Queueing Theory*. Wiley Series in Probability and Statistics. Wiley, 2011.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

A. Hobolth and E. Stone. Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. *The Annals of Applied Statistics*, 3(3):1204–1231, 2009.

J. H. Huggins, K. Narasimhan, A. Saeedi, and V. K. Mansinghka. Jump-means: Small-variance asymptotics for Markov jump processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pages 693–701, 2015.

M. C. Hughes, W. T. Stephenson, and E. B. Sudderth. Scalable adaptation of state complexity for nonparametric hidden Markov models. In *Advances in Neural Information Processing Systems 28*, pages 1198–1206, 2015.

A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skand. Aktuarietiedskr.*, 36:87–91, 1953.

T. H. Jukes and C. R. Cantor. *Evolution of Protein Molecules*. Academy Press, 1969.

P. A. W. Lewis and G. S. Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979.

J. S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89 (427):958–966, 1994a.

J. S. Liu. The fraction of missing information and convergence rate for data augmentation. *Computing Science and Statistics*, pages 490–496, 1994b.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2nd edition, 2009.

B. Miasojedow and w. Niemiro. Geometric ergodicity of Rao and Teh's algorithm for Markov jump processes and CTBNs. *Electronic Journal of Statistics*, 11(2): 4629–4648, 2017.

J. Møller, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2):451–458, 2006.

I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 359–366. AUAI Press, 2006.

R. M. Neal. Taking bigger Metropolis steps by dragging fast variables. Technical report, Department of Statistics, University of Toronto, 2004.

R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

U. Nodelman, C.R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 378–387, 2002.

M. Opper and G. Sanguinetti. Variational inference for Markov jump processes. In *Advances in Neural Information Processing Systems 20*, 2007.

J. Pan, V. Rao, P. K. Agarwal, and A. E. Gelfand. Markov-modulated marked Poisson processes for check-in data. In *Proceedings of The 33rd International Conference on Machine Learning(ICML 2016)*, pages 2244–2253, 2016.

J. Pan, B. Zhang, and V. Rao. Collapsed variational Bayes for Markov jump processes. In *Advances in Neural Information Processing Systems 30*, pages 3749–3757, 2017.

O. Papaspiliopoulos, G. O. Roberts, and M. Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, 22(1):59–73, 2007.

H. Philippe, N. Lartillot, and N. Rodrigue. Uniformization for sampling realizations of Markov processes: applications to Bayesian implementations of codon substitution models. *Bioinformatics*, 24(1):56–62, 2007.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3:4–16, 1986.

V. Rao and Y. W. Teh. Gaussian process modulated renewal processes. In *Advances in Neural Information Processing Systems 24*, pages 2474–2482, 2011.

V. Rao and Y. W. Teh. MCMC for continuous-time discrete-state systems. *Advances in Neural Information Processing Systems 25*, pages 701–709, 2012.

V. Rao and Y. W. Teh. Fast MCMC sampling for Markov jump processes and extensions. *Journal of Machine Learning Research*, 14:3295–3320, 2013.

A. Saeedi and A. Bouchard-Côté. Priors over recurrent continuous time processes. In *Advances in Neural Information Processing Systems 24*, pages 2052–2060, 2011.

P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. In *Advances in Neural Information Processing Systems 18*, pages 1145–1152. 2006.

S. L. Scott and P. Smyth. The Markov modulated Poisson process and Markov Poisson cascade with applications to web traffic modeling. *Bayesian Statistics*, 7: 1–10, 2003.

Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 19*, pages 1353–1360, 2006.

P. Wang and P. Blunsom. Collapsed variational Bayesian inference for hidden Markov models. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 599–607. PMLR, 2013.

J. Xu and C. R. Shelton. Intrusion detection using continuous time Bayesian networks. *Journal of Artificial Intelligence Research*, 39:745–774, 2010.

B. J. Yoon. Hidden Markov models and their applications in biological sequence analysis. *Current Genomics*, 10:402–415, 2009.

Y. Yu and X. Meng. To center or not to center: That is not the question-an ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011.

B. Zhang and V. Rao. Efficient MCMC for parameter inference for Markov jump processes. arXiv preprint arXiv:1704.02369, 2018.

APPENDIX

A. APPENDIX

In the appendix, we describe the particle MCMC algorithm for Bayesian inference for Markov jump processes. We evaluate its performance and compare it with other algorithms in Chapter 6.

Particle MCMC for Bayesian inference for Markov jump processes

Particle MCMC (Andrieu et al., 2010) uses particle filtering to get an unbiased estimate of the marginal $P(X|\theta)$. Plugging this into the Metropolis-Hastings acceptance probability results in an MCMC sampler that targets the correct posterior. The resulting scheme does not exploit the structure of the MJP, and we show that it is quite inefficient in the experiments in Chapter 6.

A sequential Monte Carlo algorithm for MJPs inference

We first describe a sequential Monte Carlo algorithm for MJPs inference that underlies particle MCMC. Denote by $S_{[t_1',t_2']}$ the MJP trajectory from time t_1' to time t_2'. Our target is to sample an MJP trajectory $S_{[0,t_{end}]}$ given n noisy observations $X = (x_1, x_2, ..., x_n)$, at time $t_1^X, t_2^X, ..., t_n^X$. The initial value of the Markov jump process trajectory can be simulated from its initial distribution over states: $S(0) \sim \pi_0$. $S_{[t_i^X, t_{i+1}^X]}$, its values over any interval $[t_i^X, t_{i+1}^X]$ can be simulated by Gillespie's algorithm (see algorithm 1). For the ith observation x_i at time t_i^X, denote the likelihood for $S(t_i^X)$ as $P(x_i|S(t_i^X))$.

Algorithm 9 The SMC sampler for MJP trajectories

Input: Prior π_0, n observations X, Number of particles N, rate-matrix A.

Output: New MJP trajectory $S'(t) = (s'_0, S', T')$.

1: Define $t_0^X = 0$ and $t_{n+1}^X = t_{end}$.

2: Sample initial states for N particles $S^k(0)$ from π_0, $k = 1, ..., N$.

3: **for** $i = 1, ..., n+1$: **do**

4: (a) For $k = 1, 2, ..., N$, update particle k from $[0, t_{i-1}^X]$ to $[0, t_i^X]$ by forward simulating $S^k_{[t_{i-1}^X, t_i^X]} | S^k(t_{i-1}^X)$ via Gillespie's algorithm.

5: (b) Calculate the weights $w_i^k = P(x_i | S^k(t_i^X))$ and normalize $W_i^k = \frac{w_i^k}{\sum_{k=1}^N w_i^k}$.

6: (c) Sample $J_i^k \sim \text{Multi}(\cdot | (W_i^1, \ldots, W_i^N))$, $k = 1, 2, ..., N$.

7: (d) Set $S^k_{[0, t_i^X]} := S^{J_i^k}_{[0, t_i^X]}$.

8: **end for**

The SMC algorithm gives us an estimate of the marginal likelihood $P_\theta(X_{1:n})$.

$$\hat{P}_\theta = \hat{P}_\theta(X_1) \prod_{i=2}^{n} \hat{P}_\theta(X_i | X_{1:i-1}) = \prod_{i=1}^{n} \left[\sum_{k=1}^{N} \frac{1}{N} w_i^k \right].$$

Particle MCMC algorithm for inference over MJP trajectory and parameters

Algorithm 10 outlines the Particle MCMC algorithm for the inference. The acceptance probability involves the marginal likelihood returned by the SMC algorithm we described.

Algorithm 10 The particle marginal MH sampler for MJP trajectories

Input: The observations X, the MJP path $S(t) = (s_0, S, T)$, number of particles N, parameter θ and π_0, $P(\theta)$ prior of θ, proposal density $q(\cdot|\cdot)$.

Output: New MJP trajectory $S'(t) = (s_0', S', T')$.

1: Sample $\theta^* \sim q(\cdot|\theta)$.

2: Run the SMC algorithm above targeting $P_{\theta^*}(\cdot|X_{1:n})$ to sample $S^*(t)$ from $\hat{P}_{\theta^*}(\cdot|X_{1:n})$ and let $\hat{P}_{\theta^*}$ denote the estimate of the marginal likelihood.

3: Accept $\theta^*, S^*(t)$ with probability

$$\mathsf{acc} = 1 \wedge \frac{\hat{P}_{\theta^*} P(\theta^*)}{\hat{P}_{\theta} P(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}.$$

VITA

VITA

Boqian Zhang was born in 1992 in Anhui province, and grew up in Hainan province, the southernmost province of China. He received a Bachelor of Science degree in Statistics at the School of Mathematical Sciences, Peking University in July 2014. He also received a Bachelor of Arts degree in Economics at China Center for Economic Research, Peking University in the same year. Then he joined the Department of Statistics at Purdue University and started his graduate study in August 2014. He received a Master of Science degree in Mathematical Statistics in December 2015 and earned a doctoral degree in Statistics in May 2019. His research interests include machine learning, computational statistics, Markov chain Monte Carlo methods and point processes. His Ph.D. research focused on efficient path and parameter inference for Markov jump processes. He had internship experience at J.P. Morgan and Bank of America Merrill Lynch. After graduating, he would join the Quantitative Strategies Group in Bank of America Merrill Lynch as a Quantitative Associate in New York City. He would like to pursue a professional career in quantitative trading industry.