Efficient and Scalable Markov Chain Monte Carlo Methods and its Biological Applications

by

Yizhe Zhang

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Alexander Hartemink

_____
Katherine Heller

_____
Scott Schmidler

_____
David Dunson

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program in Computational Biology and Bioinformatics
in the Graduate School of Duke University
2018

ABSTRACT

Efficient and Scalable Markov Chain Monte Carlo Methods and its Biological Applications

by

Yizhe Zhang

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Alexander Hartemink

_____
Katherine Heller

_____
Scott Schmidler

_____
David Dunson

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program in Computational Biology and Bioinformatics
in the Graduate School of Duke University
2018

Abstract

Markov Chain Monte Carlo (MCMC) stands as a fundamental approach for probabilistic inference in many computational statistics problems. Its application to computational biology and bioinformatics has attracted much attention in recent decades. A pivot question in MCMC is to design methods to efficiently draw samples from an unnormalized density function. Two auxiliary-variable sampling schemes, Hamiltonian Monte Carlo (HMC) and the slice sampler, have been introduced for tackling this challenge. Despite the great success of these two methods, little research has been done to investigate their connections, as well as their sampling efficiency.

This thesis first focus on the theoretical connection (chapter 3), the unification and generalization of slice sampling and HMC. Base on these theoretical analysis, I present a generalized HMC that demonstrate efficient exploration of target distribution, especially when the target distribution has multiple modes. The advantage over vanilla HMC is verified theoretically and experimentally. Furthermore, I discussed the tradeoff between mixing efficiency and potential issues of this generalized HMC method. The advances also include potential extensions on utilizing geometric information and higher order numerical integration for better performance.

The second part of the thesis, presented in chapter 4, concerns some advances remedying the practical issues of the generalized sampler, and how to scale up with large datasets. Chapter 4 first develops a novel scalable approximate sampling approach based on the generalized HMC method proposed in chapter 3 and stochastic gradient sampling

methods. This is followed by empirical verification that such an approach can deliver better exploration over complicated multimodal posterior regardless of lack of conjugacy.

The remaining part of this thesis, consisting chapter 5 and chapter 6, discuss advances of scalable Bayesian method for some generic and core Biomedical applications. Two Bayesian inferential tasks involving latent variable model are discussed. Chapter 5 focuses on applying Bayesian inference for discrete time-series biological data. Chapter 6 concerns a non-linear latent topic model with supervised label. These approaches exemplifies the Bayesian inference method in chapter 4, and demonstrate some innovations on maintaining accurate and scalable inference while facilitating model interpretability.

Finally, chapter 7 concludes the dissertation and discussion some potential future studies in both methodology and applications.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathbb{E}$	The expectation operation
$\mathbb{P}$	The probability symbol
$\mathbb{R}$	The set of real numbers
$\mathcal{N}(0, \Sigma)$	The multivariate Gaussian random variable with mean zero and covariance Σ.
$\mathbf{I}$	The identity matrix
$\mathbb{X}$	The slice interval in slice sampling

Abbreviations

HMC	Hamiltonian Monte Carlo
SS	Slice sampler
MGS	Monomial Gamma sampler
MGHMC	Monomial Gamma Hamiltonian Monte Carlo
MG-SS	Monomial Gamma slice sampler
SG-MCMC	Stochastic gradient Markov chain Monte Carlo
SGMGT	Stochastic gradient monomial Gamma thermostats
BPL	Bernoulli-Poisson link
RNN	Recurrent neural network
MLP	Multi-layer perceptron

Acknowledgements

The completion of this dissertation gives the opportunity to present my gratitude for many individuals who influenced me in numerous aspects during my phd study. I wish to first acknowledge my debt of profound gratitude to my supervisor Professor Lawrence Carin. Having the supervision from Prof. Carin is the most precious experience during my study at Duke. It's hard to evaluate how much his support, guidance and encouragement means to me throughout my PhD study. Over the years, I cannot remember how many times he direct me to deviate from wrong track of research direction, and how many times he inspires me when I was stuck in research project – not to mention each and every time he spent hours by hours editing my every paper and helping me correct and proofread all the mistakes from a foreign English writer. I have taken too much for granted for what he did for me. Joining Duke university as a Ph.D. and working with him might be the most important and the best ever decision I've made in the trajectory of my life.

I would like to deliver my sincere gratitude to Professor Alexander Hartemink, for providing kind concern on my research during the rotation and throughout my Ph.d. study. Professor Hartemink directed me for the very first project at Duke, and I really enjoy the enlighting discussion with him every week. His support is so important for my initial exploration on my research field. Thanks also to Professor Katherine Heller. We start collaborating since the end of my first year. She guided me through many classes and projects. She was so kind to treat me as her student and friend, and I really learned a lot from the Heller's group weekly meeting. I would really appreciate the inspiring conversation

with her. Special thanks are also due to Professor Scott Schmidler and Professor David Dunson, for all their constructive suggestions and discussions on my research. Professor Schmidler was the advisor of my master's degree in statistic department. They guided me in various aspects and lead me through this challenging journey.

I would also like to thank several colleagues in the ECE department for their support and valuable discussions. I thank Professor Ricardo Henao, who was a fantastic friend and a patient mentor for me. His help is crucial to my research. I also hope to thank Dr. Changyou Chen, Dr. Qinliang Su, Zhe Gan, Kai Fan, Dr. Xiangyu Wang, Dinghan Shen and Guoyin Wang and Wenlin Wang for their collaborations and supports in each and every time. Thanks to Jianqiao Li and Siyang Yuan for their supportive collaborations on research project. Thanks to all my friends and colleagues in company with me at Duke iid. including but not limited to Liqun Chen, Shuyang Dai and Ruiyi Zhang. We have had a great fun for tennis and board games. I am especially grateful to many other colleague who have also had an important influence on the development of this dissertation. I benefit a lot from the discussion with talented colleagues in Carin's research group. I acknowledge supports of the NSF and NIH. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author and do not necessarily reflect the views of the NSF or NIH. Finally, but certainly not the least, I would like to acknowledge the support of family. I would like to thank my parents. Their moral support of my ambition to earn a PhD degree was a major factor in my ability to do so. I will always be grateful for their incredible generosity.

1

Introduction

Bayesian inference are widely applied in many computational biology problems, for example, regulatory network inference, epidemiology study and sequence analysis. MCMC is the one of the standard tool for tackling Bayesian inference. In many scenarios, when performing sampling, one can only obtain an unnormalized probability density function, where the normalization constant is intractable. One of the most influential algorithms that tackling this issue is Metropolis-Hastings (MH) Metropolis et al. (1953). Despite its great success, the *random walk* nature often delivers inefficient mixing of the Markov chain Robert and Casella (2004). An inappropriate setting of transition kernel would result in either low acceptance ratio or slow move. Such situation is exaggerated in high dimensional cases, where the sample from the chain can be highly correlated. As a consequence, the effective sample size is usually relatively small. A number of adaptations have been proposed to mitigate these issues Haario et al. (2001); Neal (2011), however, achievable improvements are limited if attempting maintaining the Markov property and reversibility of the chain Girolami and Calderhead (2011); Andrieu and Thoms (2008); Neal (1993).

To mitigate the random walk behavior in MH, several approaches have been proposed, such as Hamiltonian Monte Carlo (HMC) Neal (2011); Duane et al. (1987). HMC augments

a target distribution with auxiliary momentum variables, and uses gradient information to propose distant samples, while maintaining ergodic property and detailed balance. The ability of long-range movement with a high acceptance ratio significantly improves mixing performance. However, HMC is sensitive to parameter setting and can only sample continuous distributions. Towards solving these issues, method were proposed to use adaptive leap-frog steps Homan and Gelman (2014), adaptive trajectory length Nishimura and Dunson (2016b) and to relax the discrete distributions sampling tasks to continuous distributions Pakman and Paninski (2013); Zhang et al. (2012); Nishimura et al. (2017).

The improvement can be further boosted by leveraging geometric manifold information Girolami and Calderhead (2011); Nishimura and Dunson (2016a), by considering better numerical integrators Chao et al. (2015), or by relaxing the detailed balance constraint Sohl-Dickstein et al. (2014).

A different direction towards improving sampling performance is the slice sampler Neal (2003). The slice sampler is conceptually related to HMC in the sense that both use auxiliary variables for efficient moves. These moves can be automatically adapted to match the relative scale of the local region being sampled Neal (2003). The sampling procedure alternates between uniformly drawing samples from the target distribution and uniformly drawing the slice variables. Unlike HMC, slice sampling does not require local gradient information. Instead, the primary effort is to locate slice intervals, where the unnormalized density values are greater than the slicing variable. This is typically hard to compute directly, thus requires local search Neal (2003). Further, it is generally less feasible in high-dimensional parameter spaces, because the slice interval is difficult to approximate. For example, using hyper-rectangle estimation may result in high rejection rates Neal (2003). The standard HMC and slice sampler is reviewed in chapter 2.

In advance to the standard HMC and slice sampler, we hope to develop a scalable, yet efficient sampling methods, based on the insight of the intrinsic connection between HMC and slice sampling. This novel sampling method is primarily discussed in chapter

3. Such method may potentially be applied to a wide range of computational statistic problems. We remark that the proposed method may have several potential advantages in Bayesian inferential tasks: 1). No local conjugacy is required. 2). Much lower stationary autocorrelation comparing with standard HMC methods, yielding higher effective sample size. 3) particularly advantageous when applying to complicated multimodal distributions. However, the performance gain is at a cost of numerical difficulty and additional initial convergence efforts. Thereby a practical sampler would require compromising the additional difficulties and theoretical gain by selecting appropriate hyper-parameter. Furthermore, I discussed several potential extensions of the proposed method with higher-order numerical integration or geometrically adaptive methods.

Chapter 4 extend upon chapter 3 by discussing an approximate sampling method towards addressing or mitigating some of above computation and convergence difficulties, and also being beyond the full-batch sampler to compensate the increasing need of scalable sampling method for large datesets or streaming data. In chapter 4, we leverage a stochastic gradient MCMC framework which takes mini-batch of data as input. The asymptotic convergence to the target distribution is theoretically justified. In addition, the numerical difficulties are remedied via a softened version of kinetics. Additional first-order dynamics and stochastic resampling procedure are introduced to alleviate initial convergence issue.

Chapter 5 and 6 builds upon the innovations in chapter 4 to apply Bayesian factor modeling for inhomogeneous spatio-temporal data (chapter 5) and non-linear supervised topic learning (chapter 6). Chapter 5 also discuss another Gibbs sampling approach scaling with the non-zero elements, by utilizing Poisson factor models. These two Bayesian inferential approaches are compared and discussed. Using simulation study and real-data, we demonstrate the effectiveness of the proposed approach in accommodating temporal data with latent factors.

Chapter 6 tackles the interpretability issue with supervised deep models. Traditional deep learning strategies achieved the state-of-the-art results in many real-world problems,

including phenotype predicationPark and Kellis (2015). Thus, it has recently gained increasing attention to formulate various application into deep learning framework. However, deep learning is often criticized to be a black-box inference, where the underlying mechanism can not be directly drawn from the model. We propose a Bayesian strategy towards addressing this issue, by constraining the parameter variables with an informative prior. Consequently, the hidden units in each perceptron layer can be interpreted as topics that composed by a (sparse) weighted abundance of each single features. Additionally, a specific prior is designed to encourage non-overlapping topic assignment. We use the method developed in chapter 4 as our inference tool. Illustrative examples in supervised topic modeling and substantive data analysis in applications in infection source prediction demonstrate such a model can yield accurate prediction while still maintaining interpretability.

A summary of the dissertation, possible extensions and future work are discussed in chapter 7.

2

Preliminaries

This chapter provides some background knowledge for future discussion. The two fundamental auxiliary MCMC methods discussed, namely Hamiltonian (Hybrid) Monte Carlo and slice sampler, are used as core and generic building blocks for many Bayesian inferential tasks.

2.1 Hamiltonian Monte Carlo

Suppose we are interested in sampling a random variable x from an unnormalized density function $f(x) \propto \exp[-U(x)]$, where $U(x)$ is the potential energy function. *Hamiltonian Monte Carlo* (HMC) augments the target density with an auxiliary momentum random variable p, that is independent of x. The distribution of p is specified as $\propto \exp[-K(p)]$, where $K(p)$ is the kinetic energy function. Define $H(x, p) = U(x) + K(p)$ as the Hamiltonian. We have omitted the dependency of $H(\cdot)$, x and p on the system time τ for simplicity. HMC iteratively performs *dynamic evolving* and *momentum resampling* steps, by sampling x_t from the target distribution and p_t from the momentum distribution (Gaussian as $K(p) = p^2$), respectively, for $t = 1, 2, \ldots$ iterations.

Figure 2.1 illustrates two iterations of this procedure. Starting from point $\{x_t(0), p_t(0)\}$

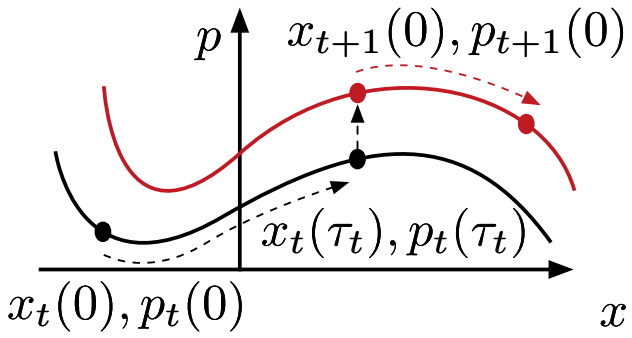


FIGURE 2.1: Representation of HMC sampling.

at the t-th (discrete) iteration, HMC leverages the Hamiltonian dynamics, governed by *Hamilton's equations* in (3.1) to propose the next sample $\{x_t(\tau_t), p_t(\tau_t)\}$, at system time τ_t. The *position* in HMC at iteration $t+1$ is updated as $x_{t+1}(0) = x_t(\tau_t)$ (*dynamic evolving*). A new momentum $p_{t+1}(0)$ is resampled independently from a Gaussian distribution (assuming $K(p) = p^2$), establishing the next initial point $\{x_{t+1}(0), p_{t+1}(0)\}$ for iteration $t+1$ (*momentum resampling*). The latter point corresponds to the initial point of a new trajectory because the Hamiltonian $H(\cdot)$ is commensurately updated. This means that trajectories correspond to distinct values of $H(\cdot)$.

Typically, numerical integrators such as the *leap-frog* method Neal (2011) are employed to numerically approximate the Hamiltonian dynamics. In practice, a random number (uniformly drawn from a fixed range) of discrete numerical integration steps (leap-frog steps) are often used (corresponding to random time τ_t along the trajectory), which has been shown to have better convergence properties than a single leap-frog step Livingstone et al. (2016). The discretization error introduced by the numerical integration is corrected by a Metropolis Hastings (MH) step. However, if the Hamiltonian dynamics can be simulated exactly, which is rarely the case, all samples are accepted and the MH step can be omitted.

The total Hamiltonian is preserved under perfect simulation. However, closed-form dynamic updates are often intractable, thus numerical integrators such as leap-frog method are applied to simulate the Hamiltonian flow. If the integrator is symplectic, by the

Liouville's theorem, the resulting sampler is invariant to the target distribution.

The ability of long-range movement with a high acceptance ratio significantly improves mixing performance. However, HMC is sensitive to parameter setting and can only sample continuous distributions. Towards solving these issues, method were proposed to use adaptive leap-frog steps Homan and Gelman (2014), and to relax the discrete distributions sampling tasks to continuos distributions Pakman and Paninski (2013); Zhang et al. (2012). The improvement can be further boosted by leveraging geometric manifold information Girolami and Calderhead (2011), by considering better numerical integrators Chao et al. (2015), or by relaxing the detailed balance constraint Sohl-Dickstein et al. (2014).

2.2 Slice sampling

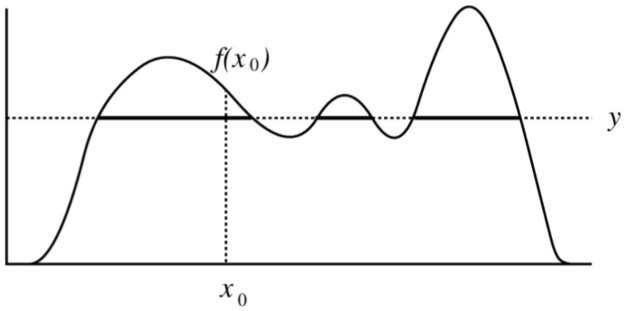


FIGURE 2.2: Representation of slice sampling.

A different direction towards improving sampling performance is the slice sampler Neal (2003). The slice sampler is related to HMC in the sense that both use auxiliary variables for efficient moves. These moves can be automatically adapted to match the relative scale of the local region being sampled Neal (2003). The sampling procedure alternates between uniformly drawing samples from the target distribution and uniformly drawing the slice variables.

Slice sampling is conceptually simpler than HMC. It augments the target unnormalized density $f(x)$ with a random variable y, with joint distribution expressed as $p(x, y) = Z_1^{-1}$, s.t. $0 < y < f(x)$, where $Z_1 = \int f(x)dx$ is the normalization constant, and the marginal

distribution of x exactly recovers the target normalized distribution $f(x)/Z_1$. To sample from the target density, slice sampling iteratively performs a *conditional sampling step* from $p(x|y)$ and *sampling a slice* from $p(y|x)$. This procedure is illustrated in Figure 2.2. At iteration t, starting from x_t, a slice y_t is uniformly drawn from $(0, f(x_t))$. Then, the next sample x_{t+1}, at iteration $t+1$, is uniformly drawn from the *slice interval* $\{x : f(x) > y_t\}$.

HMC and slice sampling both augment the target distribution with *auxiliary variables* and can propose long-range moves with high acceptance probability. Unlike HMC, slice sampling does not require local gradient information. Instead, the primary effort is to locate slice intervals, where the unnormalized density values are greater than the slicing variable. This is typically hard to compute directly, thus requires local search Neal (2003). Further, it is generally less feasible in high-dimensional parameter spaces, because the slice interval is difficult to approximate. For example, using hyper-rectangle estimation may result in high rejection rates Neal (2003).

As a consequence, adaptive methods are often applied (Neal, 2003). Alternatively, one recent attempt to perform efficient slice sampling on latent Gaussian models samples from a high-dimensional elliptical curve parameterized by a single scalar (Murray et al., 2009). It has been shown that in some cases slice sampling is more efficient than Gibbs sampling and Metropolis-Hastings, due to the adaptability of the sampler to the scale of the region currently being sampled (Neal, 2003). However, despite the advance in adaptive slice sampling methods, in general applying slice sampling in high dimensional cases is still often difficult.

Theoretically connecting these two methods will give insight that may help us mitigate some of their shortcomings. For instance, HMC is more convient in practice, since estimating high-dimensional slice intervals in slice sampling is often difficult Neal (2003). However, theoretical analysis is far easier under the slice sampling formalism.

3

Towards improving the efficiency of Hamiltonian Monte Carlo

3.1 Introduction

In this chapter, we hope to develop an efficient sampling methods, based on the insight of the intrinsic connection between HMC and slice sampling, which are two popular auxiliary-variable sampling schemes. Such methods may potentially be applied to a wide range of computational statistic problems. As entailed in Chapter 2, Markov Chain Monte Carlo (MCMC) sampling (Robert and Casella, 2004) stands as a fundamental approach for probabilistic inference in many computational statistical problems. HMC exploits gradient information to propose samples along a trajectory that follows *Hamiltonian dynamics* (Duane et al., 1987), introducing momentum as an auxiliary variable. Extending the random proposal associated with Metropolis-Hastings sampling (Neal, 2003), HMC is often able to propose large moves with acceptance rates close to one (Neal, 2011). Limitations of HMC include being sensitive to parameter tuning and being restricted to continuous distributions. The slice sampler (Neal, 2003) alternates between drawing conditional samples based on a target distribution and a uniformly distributed slice variable (the auxiliary variable). The

problem with the slice sampler is the difficulty of solving for the slice interval, *i.e.*, the domain of the uniform distribution, especially in high dimensions. Despite the success of slice sampling and HMC, little research has been performed to investigate their connections.

In this chapter we use the Hamilton-Jacobi equation from classical mechanics to show that slice sampling is equivalent to HMC with a (simply) *generalized* kinetic function. Further, we also show that different settings of the HMC kinetic function correspond to *generalized* slice sampling, with a *non-uniform* conditional slicing distribution. Based on this relationship, we develop theory to analyze the newly proposed broad family of auxiliary-variable-based samplers. We prove that under this special family of distributions for the momentum in HMC, as the distribution becomes more heavy-tailed, the one-step autocorrelation of samples from the target distribution converges *asymptotically* to zero, leading to potentially decorrelated samples. While of limited *practical* impact, this theoretical result provides insights into the properties of the proposed family of samplers. We also elaborate on the practical tradeoff between the increased computational complexity associated with improved theoretical sampling efficiency. In the experiments, we validate our theory on both synthetic data and with real-world problems, including Bayesian Logistic Regression (BLR) and Independent Component Analysis (ICA), for which we compare the mixing performance of our approach with that of standard HMC and slice sampling.

We remark that the proposed method may have several potential advantages:

1. No local conjugacy is required.

2. Much lower autocorrelation comparing with current methods, yielding higher effective sample size.

3. Can handle streaming data or large scale data.

3.2 Solving Hamiltonian dynamics via the Hamilton-Jacobi equation

A Hamiltonian system consists of a *kinetic* function $K(p)$ with *momentum* variable $p \in \mathbb{R}$, and a potential energy function $U(x)$ with coordinate $x \in \mathbb{R}$. We elaborate on multivariate cases in the appendix A. The dynamics of a Hamiltonian system are completely determined by a set of first-order Partial Differential Equations (PDEs) known as *Hamilton's equations* (Arnol'd, 2013):

$$\frac{\partial p}{\partial \tau} = -\frac{\partial H(x,p,\tau)}{\partial x}\,, \tag{3.1}$$

$$\frac{\partial x}{\partial \tau} = \frac{\partial H(x,p,\tau)}{\partial p}\,, \tag{3.2}$$

where $H(x,p,\tau) = K(p(\tau)) + U(x(\tau))$ is the *Hamiltonian*, and τ is the system time. Solving (3.1) gives the dynamics of $x(\tau)$ and $p(\tau)$ as a function of system time τ. In a Hamiltonian system governed by (3.1), $H(\cdot)$ is a constant for every τ (Arnol'd, 2013). A specified $H(\cdot)$, together with the initial point $\{x(0), p(0)\}$, defines a *Hamiltonian trajectory* $\{\{x(\tau), p(\tau)\} : \forall\tau\}$, in $\{x, p\}$ space.

It is well known that in many practical cases, a direct solution to (3.1) may be difficult (Goldstein, 1965). Alternatively, one might seek to transform the original HMC system $\{H(\cdot), x, p, \tau\}$ to a dual space $\{H'(\cdot), x', p', \tau\}$ in hope that the transformed PDEs in the dual space becomes simpler than the original PDEs in (3.1). One promising approach consists of using the *Legendre* transformation (Arnol'd, 2013). This family of transformations defines a unique mapping between primed and original variables, where the system time, τ, is identical. In the transformed space, the resulting dynamics are often simpler than the original Hamiltonian system.

An important property of the *Legendre* transformation is that the form of (3.1) is preserved in the new space (Taylor, 2005), *i.e.*, $\partial p'/\partial\tau = -\partial H'(x', p', \tau)/\partial x'$, $\partial x'/\partial\tau = \partial H'(x', p', \tau)/\partial p'$. To guarantee a valid *Legendre* transformation between the original Hamiltonian system $\{H(\cdot), x, p, \tau\}$ and the transformed Hamiltonian system $\{H'(\cdot), x', p', \tau\}$,

both systems should satisfy the *Hamilton's principle* (Goldstein, 1965), which equivalently express Hamilton's equations (3.1). The form of this *Legendre* transformation is not unique. One possibility is to use a generating function approach (Goldstein, 1965), which requires the transformed variables to satisfy $p \cdot \partial x/\partial\tau - H(x,p,\tau) = p' \cdot \partial x'/\partial\tau - H(x',p',\tau)' + dG(x,x',p',\tau)/d\tau$, where $dG(x,x',p',\tau)/d\tau$ follows from the chain rule and $G(\cdot)$ is a Type-2 generating function defined as $G(\cdot) \triangleq -x' \cdot p' + S(x,p',\tau)$ (Taylor, 2005), with $S(x,p',\tau)$ being the *Hamilton's principal function* (Landau and Lifshitz, 1976), defined below. The following holds due to the independency of x, x' and p' in the previous transformation (after replacing $G(\cdot)$ by its definition):

$$p = \frac{\partial S(x,p',\tau)}{\partial x}\,, \qquad x' = \frac{\partial S(x,p',\tau)}{\partial p'}\,, \tag{3.3}$$

$$H'(x',p',\tau) = H(x,p,\tau) + \frac{\partial S(x,p',\tau)}{\partial \tau}\,. \tag{3.4}$$

We then obtain the desired *Legendre* transformation by setting $H'(x',p',\tau) = 0$. The resulting (3.4) is known as the *Hamilton-Jacobi equation* (HJE). We refer the reader to (Goldstein, 1965; Arnol'd, 2013) for extensive discussions on the *Legendre* transformation and HJE.

Recall from above that the *Legendre* transformation preserves the form of (3.1). Since $H'(x',p',\tau) = 0$, $\{x',p'\}$ are *time-invariant* (constant for every τ). Importantly, the *time-invariant* point $\{x',p'\}$ corresponds to a Hamiltonian *trajectory* in the original space, and it defines the initial point $\{x(0),p(0)\}$ in the original space $\{x,p\}$; hence, given $\{x',p'\}$, one may update the point along the trajectory by specifying the time τ. A new point $\{x(\tau),p(\tau)\}$ in the original space along the Hamiltonian trajectory, with system time τ, can be determined from the transformed point $\{x',p'\}$ via solving (3.4).

One typically specifies the kinetic function as $K(p) = p^2$ (Neal, 2011), and Hamilton's principal function as $S(x,p',\tau) = W(x) - p'\tau$, where $W(x)$ is a function to be determined

(defined below). From (3.4), and the definition of $S(\cdot)$, we can write

$$H(x,p,\tau) + \frac{\partial S}{\partial \tau} = H(x,p,\tau) - p' \tag{3.5}$$

$$= U(x) + \left[\frac{\partial S}{\partial x}\right]^2 - p' \tag{3.6}$$

$$= U(x) + \left[\frac{dW(x)}{dx}\right]^2 - p' = 0\,, \tag{3.7}$$

where the second equality is obtained by replacing $H(x,p,\tau) = U(x(\tau)) + K(p(\tau))$ and the third equality by replacing p from (3.4) into $K(p(\tau))$. From (3.5), $p' = H(x,p,\tau)$ represents the total Hamiltonian in the original space $\{x,p\}$, and uniquely defines a Hamiltonian trajectory in $\{x,p\}$.

Define $\mathbb{X} \triangleq \{x : H(\cdot) - U(x) \geqslant 0\}$ as the *slice interval*, which for constant $p' = H(x,p,\tau)$ corresponds to a set of valid coordinates in the original space $\{x,p\}$. Solving (3.5) for $W(x)$ gives

$$W(x) = \int_{x_{min}}^{x(\tau)} f(z)^{\frac{1}{2}} dz + C\,, \qquad f(z) = \begin{cases} H(\cdot) - U(z), & z \in \mathbb{X} \\ 0, & z \notin \mathbb{X} \end{cases}\,, \tag{3.8}$$

where $x_{min} = \min\{x : x \in \mathbb{X}\}$ and C is a constant. In addition, from (3.4) we have

$$x' = \frac{\partial S(x,p',\tau)}{\partial p'} = \frac{\partial W(x)}{\partial H} - \tau = \frac{1}{2}\int_{x_{min}}^{x(\tau)} f(z)^{-\frac{1}{2}} dz - \tau\,, \tag{3.9}$$

where the second equality is obtained by substituting $S(\cdot)$ by its definition and the third equality is obtained by applying Fubini's theorem on (3.8). Hence, for constant $\{x', p' = H(x,p,\tau)\}$, equation (3.9) *uniquely* defines $x(\tau)$ in the original space, for a specified system time τ.

3.3 Formulating HMC as a Slice Sampler

3.3.1 A slice sampling perspective of HMC

Consider the *dynamic evolving* step in HMC, *i.e.*, $\{x_t(0), p_t(0)\} \mapsto \{x_t(\tau), p_t(\tau)\}$ in Figure 2.1. From Section 3.2, the Hamiltonian dynamics in $\{x,p\}$ space with initial point

$\{x(0), p(0)\}$ can be performed by mapping to $\{x', p'\}$ space and updating $\{x(\tau), p(\tau)\}$ via selecting a τ and solving (3.9). As we show in the appendix A, from (3.9) and in univariate cases[1] the Hamiltonian dynamics has period $\int_{\mathbb{X}}[H(\cdot) - U(z)]^{-\frac{1}{2}} dz$ and is symmetric along $p = 0$ (due to the symmetric form of the kinetic function). Also from (3.9), the system time, τ, is specified uniformly sampled from a half-period of the Hamiltonian dynamics. *i.e.*, $\tau \sim$ Uniform $\left(-x', -x' + \frac{1}{2}\int_{\mathbb{X}}[H(\cdot) - U(z)]^{-\frac{1}{2}}\right)$. Intuitively, x' is the "anchor" of the initial point $\{x(0), p(0)\}$, w.r.t. the start of the first half period, *i.e*, when $\int_{\mathbb{X}}[H(\cdot) - U(z)]^{-\frac{1}{2}} = 0$. Further, we only need consider half a period because for a symmetric kinetic function, $K(p) = p^2$, the Hamiltonian dynamics for the two half-periods are mirrored (Taylor, 2005). For the same reason, Figure 2.1 only shows half of the $\{x, p\}$ space, when $p \geqslant 0$.

Given the sampled τ and the constant $\{x', p'\}$, equation (3.9) can be solved for $x^* \triangleq x(\tau)$, *i.e.*, the value of x at time τ. Interestingly, the integral in (3.9) can be interpreted as (up to normalization constant) a cumulative density function (CDF) of $x(\tau)$. From the inverse CDF transform sampling method, uniformly sampling τ from half of a period and solving for x^* from (3.9), are equivalent to directly sampling x^* from the following density

$$p(x^*|H(\cdot)) \propto [H(\cdot) - U(x^*)]^{-\frac{1}{2}}, \qquad \text{s.t.,}\ H(\cdot) - U(x^*) \geqslant 0\,. \tag{3.10}$$

We note that this transformation does not make the analytic solution of $x(\tau)$ generally tractable. However, it provides the basic setup to reveal the connection between the slice sampler and HMC.

In the *momentum resampling* step of HMC, *i.e.*, $\{x_t(\tau), p_t(\tau)\} \mapsto \{x_{t+1}(0), p_{t+1}(0)\}$ in Figure 2.1, and using the previously described kinetic function, $K(p) = p^2$, resampling corresponds to drawing p from a Gaussian distribution (Neal, 2011).

The algorithm to analytically sample from the HMC (*analytic HMC*) proceeds as follows: at iteration t, momentum p_t is drawn from a Gaussian distribution. The previously

[1] For multidimensional cases, the Hamiltonian dynamics are semi-periodic, yet a similar conclusion still holds. Details are discussed in the appendix A.

sampled value of x_{t-1} and the newly sampled p_t yield a Hamiltonian $H_t(\cdot)$. Then, the next sample x_t is drawn from (3.10). This procedure relates HMC to the slice sampler. To clearly see the connection, we denote $y_t = e^{-H_t(\cdot)}$. Instead of directly sampling $\{p, x\}$ as just described, we sample $\{y, x\}$ instead. By substituting $H_t(\cdot)$ with y_t in (3.10), the conditional updates for this new sampling procedure can be rewritten as below, yielding the *HMC slice sampler* (HMC-SS), with conditional distributions defined as

Sampling a slice:

$$p(y_t|x_t) = \frac{1}{\Gamma(a)f(x_t)}[\log f(x_t) - \log y_t]^{1-a}\,, \quad \text{s.t. } 0 < y_t < f(x_t)\,, \tag{3.11}$$

Conditional sampling:

$$p(x_{t+1}|y_t) = \frac{1}{Z_2(y_t)}[\log f(x_{t+1}) - \log y_t]^{1-a}\,, \quad \text{s.t. } f(x_t) > y_t\,, \tag{3.12}$$

where $a = 1/2$ (other values of a considered below), $f(x) = e^{-U(x)}$ is an unnormalized density, and $Z_1 \triangleq \int f(x)dx$ and $Z_2(y) \triangleq \int_{f(x)>y}[\log f(x) - \log y]^{-\frac{1}{2}}dx$ are the normalization constants.

Comparing these two procedures, analytic HMC and HMC-SS, we see that the *resampling momentum* in analytic HMC corresponds to *sampling a slice* in HMC-SS. Further, the *dynamic evolving* in HMC corresponds to the *conditional sampling* in MG-SS. We have thus shown that HMC can be equivalently formulated as a slice sampler procedure via (3.11) and (3.12).

3.3.2 Reformulating Standard Slice Sampler from HMC-SS

In *standard* slice sampling (described in Section 2.1), both conditional sampling and sampling a slice are drawn from *uniform* distributions. However those for HMC-SS in (3.11) and (3.12) represent *non-uniform* distributions. Interestingly, if we change a in (3.11) and (3.12) from $a = 1/2$ to $a = 1$, we obtain the desired uniform distributions for standard slice sampling. This key observation leads us to consider a generalized form of the kinetic function for HMC, described below.

Consider the *generalized* family of kinetic functions $K(p) = |p|^{1/a}$ with $a > 0$. One may rederive equations (3.5)-(3.12) using this generalized kinetic energy. As shown in the appendix A, these equations remained unchanged, with the update that each isolated 2 in these equations is replaced by $1/a$, and $-1/2$ is replaced by $a - 1$.

Sampling p (for the *momentum resampling* step) with the generalized kinetics, corresponds to drawing p from $\pi(p; m, a) = \frac{1}{2}m^{-a}/\Gamma(a+1)\exp[-|p|^{1/a}/m]$, with $m = 1$. All the formulation in the paper still holds for arbitrary m, see appendix A for details. We denote this distribution the *monomial Gamma* (MG) distribution, MG(a, m), where m is the *mass parameter*, and a is the *monomial parameter*. Note that this is equivalent to the exponential power distribution with zero-mean, described in (Nadarajah, 2005). We summarize some properties of the MG distribution in the appendix A.

Algorithm 1: MG-HMC with HJE

for $t = 1$ *to* T **do**
- Resample momentum: $p_t \sim$ MG(m, a). ;
- Compute Hamiltonian: $H_t = U(x_{t-1}) + K(p_t)$. ;
- Find $\mathbb{X} \triangleq \{x : x \in \mathbb{R}; U(x) \leqslant H_t(\cdot)\}$. ;
- Dynamic evolving: $x_t | H_t(\cdot) \propto [H_t(\cdot) - U(x_t)]^{a-1}$; $x \in \mathbb{X}$. ;

Algorithm 2: MG-SS

for $t = 1$ *to* T **do**
- Sampling a slice: ;
- Sample y_t from (3.11). ;
- Conditional sampling: ;
- Sample x_t from (3.12). ;

To generate random samples from the MG distribution, one can draw $G \sim$ Gamma(a, m) and a uniform sign variable $S \sim \{-1, 1\}$, then $S \cdot G^a$ follows the MG(a, m) distribution. We call the HMC sampler based on the generalized kinetic function, $K(p; a, m)$: *Monomial Gamma Hamiltonian Monte Carlo* (MG-HMC). The algorithm to analytically sample from the MG-HMC is shown in Algorithm 1. The only difference between this procedure and the previously described is the momentum resampling step, in that for *analytic HMC*, p is

drawn Gaussian instead of MG(a, m). However, note that the Gaussian distribution is a special case of MG(a, m) when $a = 1/2$.

Interestingly, when $a = 1$, the *Monomial Gamma Slice sampler* (MG-SS) in Algorithm 2 recovers exactly the same update formulas as in standard slice sampling, described in Section 2.1, where the conditional distributions in (3.11) and (3.12) are both uniform. When $a \neq 1$, we have to iteratively alternate between sampling from non-uniform distributions (3.11) and (3.12), for both auxiliary (slicing) variable y and target variable x.

Using the same argument from the convergence analysis of standard slice sampling (Neal, 2003), the iterative sampling procedure in (3.11) and (3.12), converges to an invariant joint distribution (detailed in the appendix A). Further, the marginal distribution of x recovers the target distribution as $f(x)/Z_1$, while the marginal distribution of y is given by $p(y) = Z_2(y)/[\Gamma(a)Z_1]$.

The MG-SS can be divided into three broad regimes: $0 < a < 1, a = 1$ and $a > 1$ (illustrated in the appendix A). When $0 < a < 1$, the conditional distribution $p(y_t|x_t)$ is skewed towards the current unnormalized density value $f(x_t)$. The conditional draw of $p(x_{t+1}|y_t)$ encourages taking samples with smaller density value (inefficient moves), within the domain of the slice interval $\mathbb{X}$. On the other hand, when $a > 1$, draws of y_t tend to take smaller values, while draws of x_{t+1} encourage sampling from those with large density function values (efficient moves). The case $a = 1$ corresponds to the conventional slice sampler. Intuitively, setting a to be small makes the auxiliary variable, y_t, stay close to $f(x_t)$, thus $f(x_{t+1})$ is close to $f(x_t)$. As a result, a larger a seems more desirable. This intuition is justified in the following sections.

3.4 Theoretical analysis

We analyze theoretical properties of the MG sampler. All the proofs as well as the ergodicity properties of analytic MG-SS are given in the appendix A.

3.4.1 One-step autocorrelation of analytic MG-SS

We present results on the univariate distribution case: $p(x) \propto e^{-U(x)}$. We first investigate the impact of the monomial parameter a on the one-step *autocorrelation function* (ACF), $\rho_x(1) \triangleq \rho(x_t, x_{t+1}) = [\mathbb{E}x_t x_{t+1} - (\mathbb{E}x)^2]/\text{Var}(x)$, as $a \to \infty$, as a first step to understand the limiting behavior of the mixing performance when $a \to \infty$. Theorem 1 characterizes the limiting behavior of $\rho(x_t, x_{t+1})$.

Theorem 1. *For a univariate target distribution, if $U(x)$ is thrice differentiable with bounded third-order derivative, and $\exp[-U(x)]$ has finite integral over $\mathbb{R}$, the one-step autocorrelation of the MG-SS parameterized by a, asymptotically approaches zero as $a \to \infty$, i.e., $\lim_{a\to 0} \rho_x(1) = 0$.*

In the appendix A we also show that, as long as $\exp[-U(x)]$ has finite integral, $\lim_{a\to\infty} \rho(y_t, y_{t+1}) = 0$. In addition, we show that $\rho(y_t, y_{t+h})$ is a non-negative decreasing function of the time lag in discrete steps h.

3.4.2 Effective sample size

The variance of a Monte Carlo estimator is determined by its Effective Sample Size (ESS) (Brooks et al., 2011), defined as $\text{ESS} = N/(1 + 2 \times \sum_{h=1}^{\infty} \rho_x(h))$, where N is the total number of samples, $\rho_x(h)$ is the h-step autocorrelation function, which can be calculated in a recursive manner (described in the appendix A). We prove in the appendix A that $\rho_x(h)$ is non-negative. Further, assuming the MG sampler is uniformly ergodic and $\rho_x(h)$ is monotonically decreasing, it can be shown that $\lim_{a\to\infty} \text{ESS} = N$. When ESS approaches full sample size, N, the resulting sampler delivers excellent mixing efficiency (Girolami and Calderhead, 2011). Details and further discussion are provided in the appendix A.

3.4.3 *Case study*

To examine a specific 1D example, we consider sampling from the exponential distribution, Exp(θ), with energy function given by $U(x) = x/\theta$, where $x \geqslant 0$. This case has analytic $\rho_x(h)$ and ESS. After some algebra (details in the appendix A),

$$\rho_x(1) = \frac{1}{a+1}\,,\ \rho_x(h) = \frac{1}{(a+1)^h}\,,\ \text{ESS} = \frac{Na}{a+2}\,,$$

$$\hat{x}_h(x_0) \triangleq \mathbb{E}_{\kappa_h(x_h|x_0)} x_h = \theta + \frac{x_0 - \theta}{(a+1)^h}.$$

These results are in agreement with Theorem 1 and related arguments of ESS and monotonicity of autocorrelation w.r.t. a. Here $\hat{x}_h(x_0)$ denotes the expectation of the h-lag sample, starting from any x_0. The relative difference $\frac{\hat{x}_h(x_0)-\theta}{x_0-\theta}$ decays exponentially in h, with a factor of $\frac{1}{a+1}$. In fact, the $\rho_x(1)$ for the exponential family class of models introduced in (Roberts and Tweedie, 1996), with potential energy $U(x) = x^\omega/\theta$, where $x \geqslant 0, \omega, \theta > 0$, can be analytically calculated. The result, provided in the appendix A, indicates that for this family, $\rho_x(1)$ decays at a rate of $\mathcal{O}(a^{-1})$.

3.4.4 *MG-HMC mixing performance*

In theory, the *analytic MG-HMC* (the dynamics in (3.9) can be solved exactly) is expected to have the same theoretical properties of the analytic MG-SS for *unimodal cases*, since they are derived from the same setup. However, the mixing performance of the two methods could differ significantly when sampling from a *multimodal* distribution, due to the fact that the Hamiltonian dynamics may get "trapped" into a single closed trajectory (one of the modes) with low energy, whereas the analytic MG-SS does not suffer from this problem as is able to sample from disjoint slice intervals (one per mode). This is a well-known property of slice sampling (Neal, 2003) that arises from (3.11) and (3.12).

Specifically, when the energy is low, there would be more than one close contours associated with the same energy. To be more precise , as shown in Figure 3.1. Left panel

shows the critical value (red) of slicing variable y in sample space, above which the slice interval will be disjoint. Right panel shows the critical value (red) in phase space of the Hamiltonian H, above which the contour will have two disjoint components. Suppose

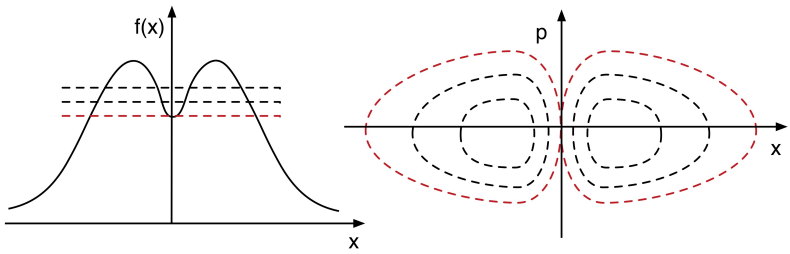


FIGURE 3.1: Left (sample space): critical value (red). Right (phase space): critical value (red)

we are sampling from a bimodal distribution. There must exist a critical value y_T, such that when the slicing variable y exceeds y_T, the slice interval $\mathbb{X}$ will have two disjoint components. The corresponding Hamiltonian, H, will also have a critical value H_T, below which there would be two closed Hamiltonian contours associated with the same energy.

However, if a is large enough, as we show in the appendix A, the probability of getting into a low-energy level associated with more than one Hamiltonian trajectory, which restrict movement between modes, is arbitrarily small. As a result, the analytic MG-HMC with large value of a is able to approach the stationary mixing performance of MG-SS.

3.5 MG sampling in practice

3.5.1 MG-HMC with numerical integrator

In practice, MG-SS (performing Algorithm 2) requires: 1) analytically solving for the slice interval $\mathbb{X}$, which is typically infeasible for multivariate cases (Neal, 2003); or 2) analytically computing the integral $Z_2(y)$ over $\mathbb{X}$, implied by the non-uniform conditionals from MG-SS. These are usually computationally infeasible, though adaptive estimation of $\mathbb{X}$ could be done using schemes like "doubling" and "shrinking" strategies from the slice sampling literature (Neal, 2003).

It is more convenient to perform approximate MG-HMC using a numerical integrator like in traditional HMC, *i.e.*, in each iteration, the momentum p is first initialized by sampling from MG(m, a), then second order Störmer-Verlet integration (Neal, 2011) is performed for the Hamiltonian dynamics updates:

$$\begin{aligned}
\mathbf{p}_{t+1/2} &= \mathbf{p}_t - \tfrac{\epsilon}{2}\nabla U(\mathbf{x}_t)\,, \\
\mathbf{x}_{t+1} &= \mathbf{x}_t + \epsilon \nabla K(\mathbf{p}_{t+1/2})\,, \\
\mathbf{p}_{t+1} &= \mathbf{p}_{t+1/2} - \tfrac{\epsilon}{2}\nabla U(\mathbf{x}_{t+1})\,,
\end{aligned} \tag{3.13}$$

where $\nabla K(\mathbf{p}) = \text{sign}(\mathbf{p}) \cdot \frac{1}{ma}|\mathbf{p}|^{1/a-1}$. When $a = 1$, $[\nabla K(\mathbf{p})]_d = 1/m$ for any dimension d, independent of $\mathbf{x}$ and $\mathbf{p}$. To avoid moving on a grid when $a = 1$, we employ a random step-size ϵ from a uniform distribution within non-negative range (r_1, r_2), as suggested in (Neal, 2011).

3.5.2 No free lunch

With a numerical integrator for MG-HMC, however, the argument about choosing large a (of great theoretical advantage as discussed in the previous section) may face practical issues.

First, a large value of a will lead to a less accurate numerical integrator. This is because as a gets larger, the trajectory of the total Hamiltonian becomes "stiffer", *i.e.*, that the maximum curvature becomes larger. When $a > 1/2$, the Hamiltonian trajectory in the phase space, $(\mathbf{x}, \mathbf{p})$, has at least 2^D (D denotes the total dimension) non-differentiable points ("turnovers"), at each intersection point with the hyperplane $\mathbf{p}^{(d)} = 0, d \in \{1 \cdots D\}$. As a result, directly applying Störmer-Verlet integration would lead to high integration error as D becomes large.

Second, if the sampler is initialized in the tail region of a light-tailed target distribution, MG-HMC with $a > 1$ may converge arbitrarily slow to the true target distribution, *i.e.*, the burn-in period could take arbitrarily long time. For example, with $a > 1$, $\nabla U(x_0)$ can be

very large when x_0 is in the light-tailed region, leading the update $x_0 + \nabla K(p_0 + \nabla U(x_0))$ to be arbitrary close to x_0, *i.e.*, the sampler does not move.

To ameliorate these issues, we provide mitigating strategies. For the first (numerical) issue, we propose two possibilities: 1) As an analog to the *"reflection"* action of (Neal, 2011), in (3.13), whenever the d-th dimension(s) of the momentum changes sign, we "recoil" the point of these dimension(s) to the previous iteration, and negate the momentum of these dimension(s), *i.e.*, $\mathbf{x}_{t+1}^{(d)} = \mathbf{x}_t^{(d)}$, $\mathbf{p}_{t+1}^{(d)} = -\mathbf{p}_t^{(d)}$. 2) Substituting the kinetic function $K(\mathbf{p})$ with a *"softened"* kinetic function, and use importance sampling to sample the momentum. The details and comparison between the *"reflection"* action and *"softened"* kinetics are discussed in the appendix A.

For the second (convergence) issue, we suggest using a step-size decay scheme, *e.g.*, $\epsilon = \max(\epsilon_1 \rho^t, \epsilon_0)$. In our experiments we use $(\epsilon_1, \rho) = (10^6, 0.9)$, where ϵ_0 is problem-specific. This approach empirically alleviates the slow convergence problem, however we note that a more principled way would be adaptively selecting a during sampling, which is left for further investigation.

As a compromise between theoretical gains and practical issues, we suggest setting $a = 1$ (HMC implementation of a slice sampler) when the dimension is relatively large. This is because in our experiments, when $a > 1$, numerical errors and convergence issues tend to overwhelm the theoretical mixing performance gains described in Section 3.4.

3.6 Experiments

3.6.1 Simulation studies

3.6.2 1D unimodal problems

We first evaluate the performance of the MG sampler with several univariate distributions: 1) Exponential distribution, $U(x) = \theta x, x \geqslant 0$. 2) Truncated Gaussian, $U(x) = \theta x^2, x \geqslant 0$. 3) Gamma distribution, $U(x) = -(r-1)\log x + \theta x$. Note that the performance of the sampler does not depend on the scale parameter $\theta > 0$. We compare the empirical $\rho_x(1)$

and ESS of the analytic MG-SS and MG-HMC with their theoretical values. In the Gamma distribution case, analytic derivations of the autocorrelations and ESS are difficult, thus we resort to a numerical approach to compute $\rho_x(1)$ and ESS. Details are provided in the appendix A. Each method is run for 30,000 iterations with 10,000 burn-in samples. The number of leap-frog steps is set to be uniformly drawn from $(100 - l, 100 + l)$ with $l = 20$, as suggested by (Livingstone et al., 2016). We also compared MG-HMC ($a = 1$) with standard slice sampling using doubling and shrinking scheme (Neal, 2003) As expected, the resulting ESS (not shown) for these two methods is almost identical. The experiment settings and results are provided in the appendix A. Figure 3.2 shows the theoretical and

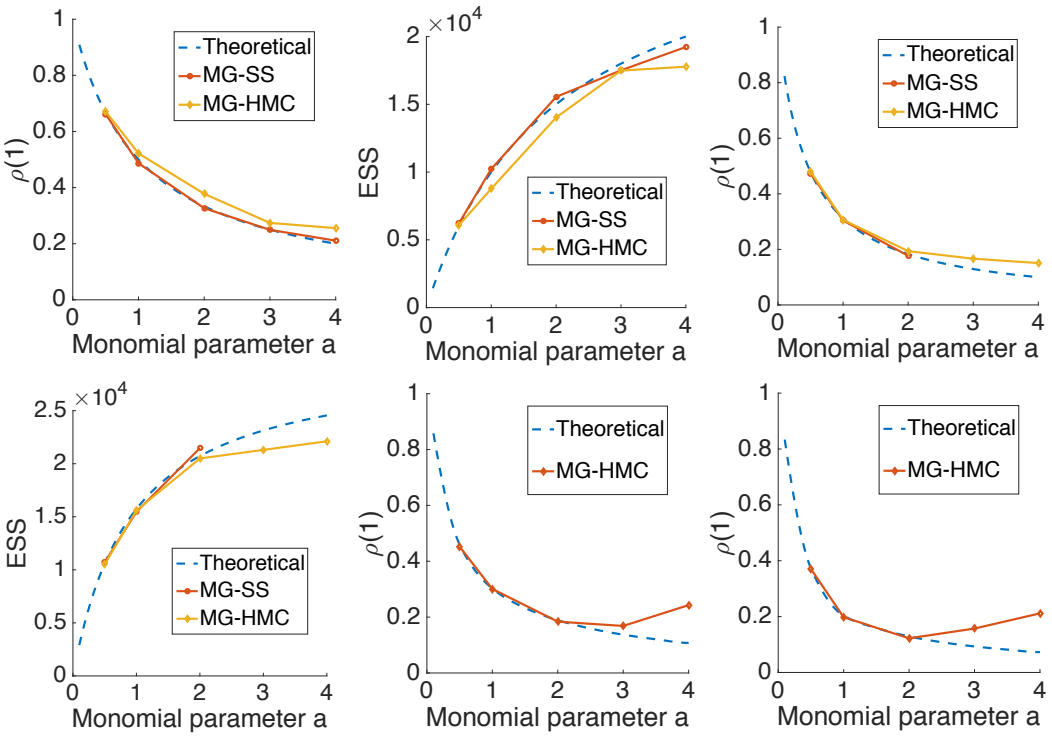


FIGURE 3.2: Theoretical and empirical $\rho_x(1)$ and ESS of toy cases.

empirical $\rho_x(1)$ and ESS of exponential distribution (a,b), $\mathcal{N}_+$ (c,d) and Gamma $r = 2, 3$ (e,f). The acceptance rates decrease from around 0.98 to around 0.77 for each case, when a grows from 0.5 to 4, as shown in Figure 3.2(a)-(d),

The results for analytic MG-SS match well with the theoretical results, however MG-

HMC seems to suffer from practical difficulties when a is large, evidenced by results gradually deviating from the theoretical values. This issue is more evident in the Gamma case (see Figure 3.2(e)), where $\rho_x(1)$ first decreases then increases. Meanwhile, the acceptance rates decreases from 0.9 to 0.5.

3.6.3 1D and 2D bimodal problems

We further conduct simulation studies to evaluate the efficiency of MG-HMC when sampling 1D and 2D multimodal distributions. For the univariate case, the potential energy is given by $U(x) = x^4 - 2x^2$; whereas $U(\mathbf{x}) = -0.2 \times (x_1 + x_2)^2 + 0.01 \times (x_1 + x_2)^4 - 0.4 \times (x_1 - x_2)^2$ in the bivariate case. We show in the appendix A that if the energy functions are symmetric along $\mathbf{x} = C$, where C is a constant, in theory, the analytic MG-SS will have ESS equal to the total sample size. However, as shown in Section 3.4, the analytic MG-HMC is expected to have an ESS less than its corresponding analytic MG-SS, and the gap between the analytic MG-HMC and analytic MG-SS counterpart should decrease with a. As a result, despite numerical difficulties, we expect the MG-HMC based on numerical integration to have better mixing performance with large a.

1D	ESS	$\rho_x(1)$
$a = 0.5$	5175	0.60
$a = 1$	10157	0.43
$a = 2$	24298	0.11

2D	ESS	$\rho_x(1)$
$a = 0.5$	4691	0.67
$a = 1$	16349	0.60
$a = 2$	18007	0.53

Table 3.1: ESS of MG-HMC for 1D and 2D bimodal distributions.

To verify our theory, we run MG-HMC for $a = \{0.5, 1, 2\}$ for 30,000 iterations with 10,000 burn-in samples. The parameter settings and the acceptance rates are detailed in the appendix A. Empirically, we find that the efficiency of HMC is significantly improved with a large a as shown in Table 3.1, which coincides with the theory in Section 3.4.

From Figure 3.3, we observe that the MG-HMC sampler with monomial parameter $a = \{1, 2\}$ performs better at jumping between modes of the target distribution, when compared to standard HMC, which confirms the theory in Section 3.4. We also compared

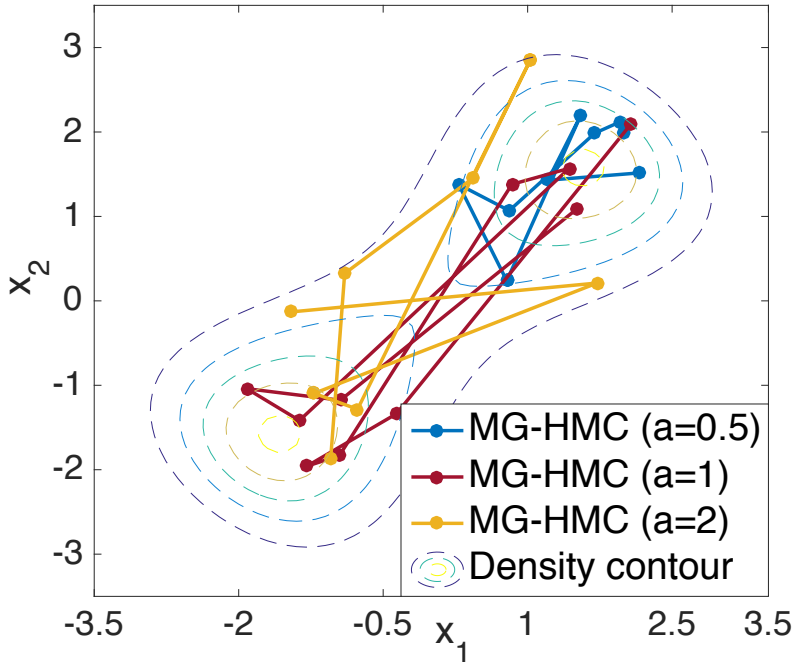


FIGURE 3.3: 10 MC samples by MG-HMC from a 2D distribution and different a.

MG-HMC ($a = 1$) with standard SS (Neal, 2003). As expected, in the 1D case, the standard SS yields ESS close to full sample size, while in 2D case, the resulting ESS is lower than MG-HMC ($a = 1$) (details are provided in the appendix A).

3.6.4 Real-world Bayesian analysis

3.6.5 Bayesian logistic regression

We evaluate our methods on 6 real-world datasets from the UCI repository (Bache and Lichman, 2013): German credit (G), Australian credit (A), Pima Indian (P), Heart (H), Ripley (R) and Caravan (C) (Van Der Putten and van Someren, 2000). Feature dimensions range from 7 to 87, and total data instances are between 250 to 5822. All datasets are normalized to have zero mean and unit variance. Gaussian priors $\mathcal{N}(\mathbf{0}, 100\mathbf{I})$ are imposed on the regression coefficients. We draw 5000 iterations with 1000 burn-in samples for each experiment. The leap-frog steps are set to be uniformly drawn from $(100 - l, 100 + l)$ with $l = 20$. Other experimental settings (m and ϵ) are provided in the appendix A.

Results in terms of minimum ESS are summarized in Table 3.2. Prediction accuracies

estimated via cross-validation are almost identical all across (reported in the appendix A). It can be seen that MG-HMC with $a = 1$ outperforms (in terms of ESS) the other two settings with $a = 0.5$ and $a = 2$, indicating increased numerical difficulties counter the theoretical gains when a becomes large. This can be also seen by noting that the acceptance rates drop from around 0.9 to around 0.7 as a increases from 0.5 to 2. The dimensionality also seems to have an impact on the optimal setting of a, since in the high-dimensional dataset Cavaran, the improvement of MG-HMC with $a = 1$ is less significant compared with other datasets, and $a = 2$ seems to suffer more of numerical difficulties. Comparisons between MG-HMC ($a = 1$) and standard slice sampling are provided in the appendix A. In general, standard slice sampling with adaptive search underperforms relative to MG-HMC ($a = 1$).

Dataset (dim)	A (15)	G (25)	H (14)	P (8)	R (7)	C (87)
$a = 0.5$	3124	3447	3524	3434	3317	33
$a = 1$	**4308**	**4353**	**4591**	**4664**	**4226**	**36**
$a = 2$	1490	3646	4315	4424	1490	7

ICA (25)
2677
3029
1534

Table 3.2: Minimum ESS for each method (dimension in parenthesis). Left: BLR; Right: ICA

3.6.6 ICA

We finally evaluate our methods on the MEG (Vigário et al., 1998) dataset for Independent Component Analysis (ICA), with 17,730 time points and 25 feature dimension. All experiments are based on 5000 MCMC samples. The acceptance rates for $a = (0.5, 1, 2)$ are $(0.98, 0.97, 0.77)$. Running time is almost identical for different a. Settings (including m and ϵ) are provided in the appendix A. As shown in Table 3.2, when $a = 1$, MG-HMC has better mixing performance compared with other settings.

3.7 Future study

To further improve the mixing efficiency and alleviate practical issues of the method proposed above, I suggest several future studies by either incorporating geometric information to adaptively move over parameter probability space with different scale, or utilizing higher order numerical integrator to reduce numerical error. Hopefully these remedies could allow extending the limit of our proposed method.

3.7.1 Riemann manifold adaptation

Although HMC methods often outperform other popular MCMC methods, they may mix slowly if there are strong correlations between variables in the target distribution. Neal (2011) showed that HMC can mix faster if M is not the identity matrix. Intuitively, such a M acts like a preconditioner. However, if the curvature of target posterior varies greatly, a global preconditioner can be inadequate.

For this reason, recent work, notably that on Riemannian manifold HMC (RMHMC) Girolami and Calderhead (2011), has considered non-separable Hamiltonian methods, in which $\mathrm{M}\boldsymbol{\theta}$ varies with position $\boldsymbol{\theta}$, so that $\boldsymbol{\theta}$ and momentum $\boldsymbol{p}$ are no longer independent in joint augmented posterior. The resulting Hamiltonian is called a non-separable Hamiltonian. For example, for Bayesian inference problems, Girolami and Calderhead (2011) proposed using the Fisher Information Matrix (FIM) of log posterior of $\boldsymbol{\theta}$, which is the metric tensor of posterior manifold. However, for a non-separable Hamiltonian, the simple leapfrog dynamics do not yield a valid MCMC method, as they are no longer reversible. Simulation of general non-separable systems requires the generalized leapfrog integrator (GLI) Girolami and Calderhead (2011), which requires computing higher order derivatives to solve a system of non-linear differential equations. The computational cost of GLI in general is $\mathcal{M}D^3$ where D is the number of parameters, which is prohibitive for large d. Girolami and Calderhead (2011) showed that the mixing performance can be improved by

leveraging geometric manifold information.

HMC is known to require heavy parameter tuning. Previous methods were proposed to use adaptive leap-frog steps Homan and Gelman (2014) or automatic stepsize Mohamed et al. (2013). Intuitively, conducting Hamiltonian dynamic update over Riemann manifold would allow adaptively choosing appropriate stepsize for each dimension.

The MGHMC with large value of a would exaggerate the relative scale of each dimension, comparing with standard HMC. Presumably, the local geometric information would dramatically improve the mixing efficiency. However, the trade-off between computation complexity and performance gain should be further evaluated.

For a total of D dimension, we have

$$\begin{aligned} H(\boldsymbol{x}, \boldsymbol{p}) &= E(\boldsymbol{x}) + \frac{1}{2} V(\boldsymbol{p})^T G(\boldsymbol{x})^{-1} V(\boldsymbol{p}) \\ &\quad + \frac{D}{a} \log \Gamma(a+1) + \frac{D(1-a)}{a} \log 2 + \frac{1}{a} \log |G(\boldsymbol{x})| \end{aligned}$$

Where $V(\boldsymbol{p}) = \left(|p_1|^{\frac{1}{2a}}, \ldots, |p_D|^{\frac{1}{2a}} \right)^T$. The resulting Hamiltonian dynamics will be,

$$\begin{aligned} \dot{x}_i &= G(\boldsymbol{x})^{-1} V(\boldsymbol{p}) \frac{\partial V(\boldsymbol{p})}{\partial p_i} \\ &= \text{sign}(p_i) \cdot \left(\frac{1}{2a} G(\boldsymbol{x})^{-1} \boldsymbol{p}^{\frac{1}{a}-1} \right)_i \\ \dot{p}_i &= -\frac{\partial E(\boldsymbol{x})}{\partial x_i} - \frac{1}{a} \text{Tr} \left[G(\boldsymbol{x})^{-1} \frac{\partial G(\boldsymbol{x})}{\partial x_i} \right] + \frac{1}{2} V(\boldsymbol{p})^T G(\boldsymbol{x})^{-1} \frac{\partial G(\boldsymbol{x})}{\partial x_i} G(\boldsymbol{x})^{-1} V(\boldsymbol{p}) \end{aligned}$$

For cases where the Fisher information matrix is expensive or intractable, one may consider using approximated pre-conditioner to substitute $G(\boldsymbol{x})^{-1}$. How to ensure the approximated FIM can still maintain the invariate target distribution is an interesting topic for further investigation.

3.7.2 High order numerical integrator

Another potential approach is to apply high order numerical method to reduce the integration error Striebel et al. (2011); Jiang and Cong (2015) For example, symmetric Partitioned Runge-Kutta (SPRK) Striebel et al. (2011) would yield numerical error in order of $O(\varepsilon^4)$, with a cost of 4 more times of functional evaluations. Such methods have advantage when the stepsize is small. One caveat is that the symmetric/symplectic property of such numerical methods is required for the sampler to be invariant to target distribution, since this will guarantee the Jacobi of the dynamic updates have determinant of 1 (i.e. Liouville's theorem holds).

Specifically,

$$\begin{aligned}
\begin{pmatrix} x_{t+1} \\ p_{t+1} \end{pmatrix} &= \begin{pmatrix} x_t \\ p_t \end{pmatrix} + \frac{\varepsilon}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
f\left[\begin{pmatrix} x \\ p \end{pmatrix}\right] &= \begin{pmatrix} \nabla_p K(p) \\ -\nabla_x E(x) \end{pmatrix} \\
k_1 = f\left[\begin{pmatrix} x_t \\ p_t \end{pmatrix}\right] \quad &, \quad k_2 = f\left[\begin{pmatrix} x_t \\ p_t \end{pmatrix} + \frac{\varepsilon}{2}k_1\right] \\
k_3 = f\left[\begin{pmatrix} x_t \\ p_t \end{pmatrix} + \frac{\varepsilon}{2}k_2\right] \quad &, \quad k_4 = f\left[\begin{pmatrix} x_t \\ p_t \end{pmatrix} + \varepsilon k_3\right]
\end{aligned}$$

One caveat is that this naive implementation may give worse result than leap-frog, due to a violation of the Liouville theorem. As a consequence, the detailed balance will no longer hold. To solve this, one should use a symmetric Runge-Kutta instead. How to design such a method requires further examinations.

3.8 Summary comments

We demonstrated the connection between HMC and slice sampling, introducing a new method for implementing a slice sampler via an augmented form of HMC. With few modifications to standard HMC, our MG-HMC can be seen as a drop-in replacement for

any scenario where HMC and its variants apply, for example, Hamiltonian Variational Inference (HVI) (Salimans et al., 2014). We showed the theoretical advantages of our method over standard HMC, as well as increasing numerical difficulties arise in align with such performance improvement. As a consequence, the optimal settings of parameters in our approach involves the counteraction between these two factors. Several future extensions can be explored to mitigate numerical issues, *e.g.*, performing MG-HMC on the Riemann manifold (Girolami and Calderhead, 2011) so that step-sizes can be adaptively chosen, and using a high-order symplectic numerical method (Striebel et al., 2011; Jiang and Cong, 2015) to reduce the discretization error introduced by the integrator. Presumably, such strategies would bring substantial improvement to the MG-HMC scheme with large a, because the difference of contour scale between dimensions can be amplified when a is large.

4

Scalable MCMC inference with generalized kinetics

4.1 Introduction

In chapter 3, we introduced a generalized framework of Hamiltonian Monte Carlo, which demonstrated that a generalized kinetic function typically improves the stationary mixing efficiency of HMC, especially when the target distribution has multiple modes. However, this advantage comes with numerical difficulties, and convergence problems due to poor initialization. Aided by gradient information, HMC is able to move efficiently in parameter space, thus greatly improving exploration. However, the emergence of big datasets poses a new challenge for MCMC, as evaluation of gradients on whole datasets becomes computationally demanding, if not prohibitive, in many cases. This development of increasingly large dataset for Bayesian models in modern machine learning has accentuated the need for both scalable and efficient generation of asymptotically exact samples from complex posterior distributions.

To scale HMC methods to big data, recent advances in Stochastic Gradient MCMC (SG-MCMC) have subsampled the dataset into *minibatches* in each iteration, to decrease

computational burden (Welling and Teh, 2011a; Chen et al., 2014; Ding et al., 2014a; Ma et al., 2015). Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011a) was first proposed to generate approximate samples from a posterior distribution using minibatches. Since then, research has focused on leveraging the minibatch idea while also providing theoretical guarantees. For instance, Teh et al. (2014) showed that by appropriately injecting noise while using a stepsize-decay scheme, SGLD is able to converge asymptotically to the desired posterior. Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014) extended SGLD with auxiliary momentum variables, akin to HMC, and introduced a *friction* term to counteract the stochastic noise due to subsampling. However, exact estimation of such noise is needed to guarantee a correct SGHMC sampler. To alleviate this issue, the Stochastic Gradient Nosé-Hoover Thermostat (SGNHT) (Ding et al., 2014a) algorithm introduced so-called *thermostat* variables to adaptively estimate stochastic noise via a thermal-equilibrium condition.

One standing challenge of SG-MCMC methods is inefficiency when exploring complex multimodal distributions. This limitation is commonly found in latent variable models with a multi-layer structure. Inefficiency is manifested because sampling algorithms have difficulties moving across modes, while traveling along the surface of the distribution. As a result, it may take a very large number of iterations (posterior samples) to cover more than one mode, greatly limiting scalability.

In this chapter, we investigate strategies for improving mixing in SG-MCMC. We propose the Stochastic Gradient Monomial Gamma Thermostat (SGMGT), building upon the Monomial Gamma Sampler (MGS) Zhang et al. (2016d) from chapter 3. We also provide more principled remedies for the numerical difficulties, and convergence problems due to poor initialization, when the monomial parameter a becomes large in MGS. By defining a *smooth* version of this generalized kinetic function, we can leverage its mixing efficiency, while satisfying the required conditions for stationarity of the corresponding stochastic process, as well as alleviating numerical difficulties arising from differentiability

issues. To ameliorate the convergence issues, we further introduce i) a sampler with an underlying elliptic stochastic differential equation system and ii) a resampling scheme for auxiliary variables (momentum and thermostats) with theoretical guarantees. The result is an elegant framework to improve stationary mixing performance on existing SG-MCMC algorithms augmented with auxiliary variables.

4.2 Stochastic Gradient MCMC

SG-MCMC is desirable when the dataset, X, is too large to evaluate the potential $U(\theta)$ using all N samples. The idea behind SG-MCMC is to replace $U(\theta)$ with an unbiased *stochastic likelihood*, $\tilde{U}(\theta)$, evaluated from a subset of data (termed a minibatch)

$$\tilde{U}(\theta) = -\frac{N}{N'}\sum_{i=1}^{N'} \log p(x_{\tau_i}|\theta) - \log p(\theta)\,, \tag{4.1}$$

where $\{\tau_1, \cdots, \tau_{N'}\}$ is a random subset of $\{1, 2, \cdots, N\}$ of size $N' \ll N$. SG-MCMC algorithms are typically driven by a continuous-time Markov stochastic process of the form (Chen et al., 2015)

$$d\Gamma = V(\Gamma)dt + D(\Gamma)dW\,, \tag{4.2}$$

where Γ denotes the parameters of the *augmented* system, *e.g.*, p and θ, $V(\cdot)$ and $D(\cdot)$ are referred as *drift* and *diffusion* vectors, respectively, and W denotes a standard Wiener process.

In SGHMC (Chen et al., 2014), the resulting stochastic dynamic process is governed by the following Stochastic Differential Equations (SDEs) (with $M = I$):

$$\begin{aligned} d\theta &= pdt\,, \\ dp &= -[\nabla\tilde{U}(\theta) + Ap]dt + \sqrt{2(AI - \hat{B}(\theta))}dW\,, \end{aligned} \tag{4.3}$$

where $\Gamma = \{\theta, p\}$, $V(\Gamma)$ is a function of $\{p, \nabla_\theta\tilde{U}, A\}$, and $D(\Gamma)$ is a function of $\{A, \hat{B}(\theta)\}$. $\nabla\tilde{U}(\theta)$ is modeled as $\nabla\tilde{U}(\theta) = \nabla U(\theta) + \sqrt{2B(\theta)}\nu$, where $\nu \sim \mathcal{N}(0, 1)$ and h is the

discretization stepsize. $\hat{B}(\theta)$ is an estimator of $B(\theta)$, A is a user-specified *diffusion* factor and I is the identity matrix. Chen et al. (2014) set $\hat{B}(\theta) = 0$ for simplicity. The reasoning is that the injected noise $\mathcal{N}(0, 2Ah)$ will dominate as $h \to 0$ (A remains as a constant), whereas $B(\theta)$ goes to zero. Unfortunately, the covariance function, $B(\theta)$, of the stochastic noise, ν, is difficult to estimate in practice.

Recently, SGNHT (Ding et al., 2014a) considered incorporating additional auxiliary variables (thermostats). The resulting SDEs correspond to

$$dp = -[\nabla\tilde{U}(\theta) + \xi \odot p]dt + \sqrt{2AI}dW\,, \tag{4.4}$$

$$d\theta = pdt\,,\ d\xi = (p \odot p - 1)dt\,, \tag{4.5}$$

where $\odot$ represents the Hadamard (element-wise) product and ξ are thermostat variables. Note that the diffusion factor, A, is decoupled in (4.4), thus ξ can adaptively fit to the unknown noise from the stochastic gradient $\nabla\tilde{U}(\theta)$.

4.3 Stochastic Gradient Monomial Gamma Sampler

We now consider i) a more efficient (generalized) kinetic function, ii) adapting the proposed kinetics to satisfy stationary requirements and alleviate numerical difficulties, iii) incorporating an additional first-order stochastic process to (4.4) and iv) stochastic resampling of the momentum and thermostats to lessen convergence issues.

Generalized kinetics The statistical physics literature traditionally considers a quadratic form of the kinetics function, and a Gaussian distribution for the thermostats in (4.4), when analyzing the dynamic system of a canonical ensemble (Tuckerman, 2010). Inspired by this, one typical assumption in previous SG-MCMC work is that the marginal distribution for the momentum and thermostat is Gaussian (Ding et al., 2014a; Li et al., 2016a). However, this assumption, while convenient, does not necessarily guarantee an optimal sampler.

In recent work, Lu et al. (2016) extended the standard (Newtonian) kinetics to a more general form inspired by relativity theory. By bounding the momentum, their *relativistic*

Monte Carlo can lessen the problem associated with large potential gradients, $\nabla U(\theta_t)$, thus resulting in a more robust alternative to standard HMC. Further, Zhang et al. (2016d) demonstrated that adopting non-Gaussian kinetics delivers better mixing and reduces sampling autocorrelation, especially for cases where the posterior distribution has multiple modes.

These ideas motivate a more general framework to characterize SG-MCMC, with potentially non-Gaussian kinetics and thermostats. As a relaxation of SGNHT (Ding et al., 2014a; Ma et al., 2015), we consider a Hamiltonian system defined in a more general form

$$H = K(p) + U(\theta) + F(\xi)\,, \tag{4.6}$$

where $K(\cdot)$ and $F(\cdot)$ are any valid potential functions, inherently implying that $\exp[-K(\cdot)]$ and $\exp[-F(\cdot)]$, define valid probability density functions.

We first consider the SDEs of SGNHT with generalized kinetics $K(p)$. The system can be obtained by generalizing $K(p) = p^T p/2$ (with identity mass matrix M for simplicity) in (4.4) with arbitrary $K(p)$, thus

$$\begin{aligned} d\theta &= \nabla K(p)dt\,, \\ dp &= -[\nabla \tilde{U}(\theta) + \xi \odot \nabla K(p)]dt + \sqrt{2AI}dW\,, \\ d\xi &= (\nabla K(p) \odot \nabla K(p) - \nabla^2 K(p))dt\,. \end{aligned} \tag{4.7}$$

However, if we set $K(p)$ as $K(p) = |p|^{1/a}$ with $a \geqslant 1$, the dynamics governing the SDEs in (4.7) will often fail to converge. This is because the sufficient condition to guarantee that the Itô process governed by the SDEs in (4.7) converge to a stationary distribution generally requires the Fokker-Planck equation to hold (Risken, 1984). Further, the existence and uniqueness of the solutions to the Fokker-Planck equation require Lipschitz continuity of drift and diffusion vectors in (4.2) (Bris and Lions, 2008). Unfortunately, this is not the case for the drift vectors in (4.7) when $a \geqslant 1$, as $\nabla K(p)$ is non-differentiable at the origin, *i.e.*, $p = 0$.

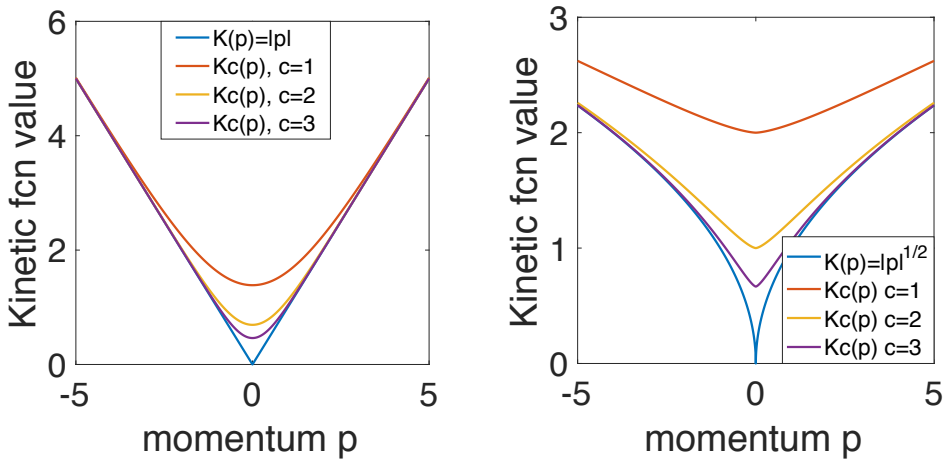


FIGURE 4.1: Softened *vs.* stiff kinetics (1D). Left: $a = 1$. Right: $a = 2$.

Softened kinetics The above limitation can be avoided by using a *softened* kinetic function $K_c(p)$. However, to keep the performance benefits from the original *stiff* kinetics, we must ensure that $K_c(p)$ has the same tail behavior. We propose that for $a = \{1, 2\}$, the softened kinetics are (for clarity we consider 1D case, however higher dimensions still apply)

$$K_c(p) = \begin{cases} -p + 2/c \log(1 + e^{cp}), & a = 1 \\ |p|^{1/2} + \frac{4}{c(1+e^{c|p|^{1/2}})}, & a = 2 \end{cases}, \tag{4.8}$$

where $c > 0$ is a *softening* parameter. Note that $K_c(p)$ is (infinitely) differentiable for any c and asymptotically approaches the stiff kinetics as $c \to \infty$. A comparison between stiff kinetics, $K(p)$, and softened kinetics $K_c(p)$ is shown in Figure 4.1, for different values of c. Discussion and formulation of the softened kinetics for arbitrary a (and M) are provided in the appendix B.

To generate samples of the momentum variable, p, from the density with softened kinetics, which is proportional to $\exp[-K_c(p)]$, we use a coordinate-wise rejection sampling, *i.e.*, the proposed p_d for the d-th dimension is rejected with probability $1 - \exp[K(p_d) - K_c(p_d)]$.

In practice, setting c to a relatively large value would still make the gradient $\nabla K_c(p)$ ill-posed close to $p = 0$, thus causing high *integration error* when simulating the Hamiltonian dynamics. Conversely, setting c to a small value will cause a *high approximation* error w.r.t. the original $K(p)$, thus resulting in a less efficient sampler. Consequently, c has to be

determined empirically as a trade-off between integration and approximation errors.

Additional First Order Dynamics Inspired by Ma et al. (2015), we consider adding Brownian motion to θ and ξ in (4.4), with variances σ_θ and σ_ξ, respectively, while maintaining the stochastic process (asymptotically) converging to the correct marginal distribution of θ. Specifically, we consider the following SDEs:

$$\begin{aligned} d\theta =& -\sigma_\theta \nabla \tilde{U}(\theta)dt + \nabla K_c(p)dt + \sqrt{2\sigma_\theta}dW\,, \\ dp =& -(\sigma_p + \gamma \nabla F(\xi)) \odot \nabla K_c(p)dt \\ & - \nabla \tilde{U}(\theta)dt + \sqrt{2\sigma_p}dW\,, \\ d\xi =& \gamma \left[\nabla K_c(p) \odot \nabla K_c(p) - \nabla^2 K_c(p)\right] dt \\ & - \sigma_\xi \nabla F(\xi)dt + \sqrt{2\sigma_\xi}dW\,. \end{aligned} \tag{4.9}$$

The variances $\{\sigma_\theta, \sigma_p, \sigma_\xi\}$ control the Brownian motion for $\{\theta, p, \xi\}$, respectively, and $\gamma > 0$ denotes a rescaling factor for the friction term of momentum updates. The additional terms $-\sigma_\theta \nabla \tilde{U}(\theta)dt + \sqrt{2\sigma_\theta}dW$ and $-\sigma_\xi \nabla F(\xi)dt + \sqrt{2\sigma_\xi}dW$ can be understood as first-order Langevin dynamics (Welling and Teh, 2011a). The variance term, σ_θ, controls the contribution of $\nabla \tilde{U}(\theta)$ to the update of θ w.r.t. $\nabla K_c(p)$. This is analogous to the hyperparameter balancing $\nabla \tilde{U}(\theta)$ and p in the SGD-with-momentum algorithm (Rumelhart et al., 1988). Derivation details for $\nabla K_c(p)$ and $\nabla^2 K_c(p)$ in (4.8), as well as other values of a, are provided in the appendix B.

The following theorem, proven in the appendix B, shows that under regularity conditions, the SDEs in (4.9) lead to posterior samples from the invariant joint distribution $p(\Gamma) \propto \exp[-H(\Gamma)]$, yielding the desired marginal distribution w.r.t. θ as $p(\theta) \propto \exp[-U(\theta)]$.

Theorem 1. *The stochastic process governed by* (4.9) *converges to a stationary distribution* $p(\Gamma) \propto \exp[-H(\Gamma)]$*, where* $H(\Gamma)$ *is as defined in* (4.6)*, and* $\Gamma = \{\theta, p, \xi\}$.

The reasoning behind increasing *stochasticity* in the SDEs is two-fold. First, the additional Langevin dynamics are crucial to SG-MCMC with generalized kinetics for large

a. For instance, for $\sigma_\theta = 0$, the update for θ from (4.7) is $\theta_{t+1} = \theta_t + \nabla K(p_t)h$. When $a > 1$ and $|p_t|$ is large, $\nabla K(p) = \frac{1}{a}|p|^{1/a-1}$ will be close to zero, thus θ_{t+1} (the next sample) will be close to θ_t, *i.e.*, the sampler moves arbitrarily slow. As discussed by Zhang et al. (2016d), this can happen when θ moves to a region where the gradient $\nabla U(\theta)$ takes a large absolute value, *e.g.*, near the low-density regions in a light-tailed distribution. Fortunately, the additional Langevin dynamics in (4.9), $-\sigma_\theta \nabla \tilde{U}(\theta)dt + \sqrt{2\sigma_\theta}dW$, compensate for the weak updating signal from $\nabla K(p)$, by an immediate gradient signal $\nabla \tilde{U}(\theta)$. Additionally, when $\tilde{U}(\theta)$ becomes small, $\nabla K(p)$ will become large. As a result, these two updating signals $\nabla K(p)$ and $\nabla \tilde{U}(\theta)$ compensate each other, thereby delivering a stable updating scheme. Likewise, the immediate gradient $\nabla F(\xi)$ in (4.9) can provide complementary updating signal for the thermostat variables, ξ, to offset the weak deterministic update $\nabla K_c(p) \odot \nabla K_c(p) - \nabla^2 K_c(p)$, when p is large.

Second, (4.9) has noise components on all parameters, $\{\theta, p, \xi\}$, making the corresponding SDEs *elliptic*. From a theoretical perspective, ellipticity/hypoellipticity are necessary conditions to guarantee existence of bounded solutions for a particular partial differential equation related to the diffusion's *infinitesimal generator*, which lies in the core of most recent SG-MCMC theory (Teh et al., 2014; Vollmer et al., 2016; Chen et al., 2015). Ellipticity is characterized by a noise process (Brownian motion) covering all components of the system, via the diffusion, $D(\Gamma)$, in (4.2). This means $D(\Gamma)$ is block diagonal, thus a positive definite matrix (Mattingly et al., 2010). In a typical hypoelliptic case, the noise process is imposed on a subset of Γ. However, hypoellipticity also requires the noise to be able to spread through the system via the drift term, $V(\Gamma)$, which may not be true for general $V(\Gamma)$. For instance, in (4.4), $\Gamma = \{\theta, p, \xi\}$ and $D(\Gamma)$ is block diagonal with entries $\{0, \sqrt{2A}I, 0\}$, *i.e.*, θ and ξ are not explicitly influenced by the noise process, W, thus hypoellipticity cannot be guaranteed.

To the authors' knowledge, for existing SG-MCMC algorithms, only SGLD where $d\theta = -\nabla_\theta \tilde{U}(\theta)dt + \sqrt{2}dW$, satisfies the ellipticity property, while other algorithms such

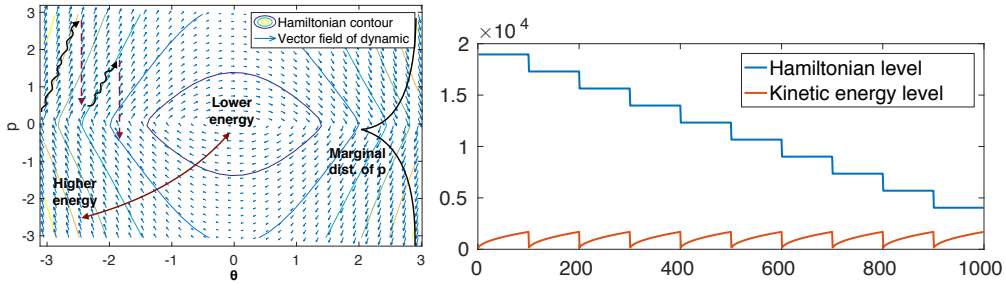


FIGURE 4.2: Momentum resampling. Resampling occurs every 100 iterations.

as SGHMC and SGNHT assume hypoellipticity, thus their corresponding $D(\Gamma)$ are not positive definite.

One caveat of (4.9) is that if σ_θ and σ_ξ are too large, the updates will be dominated by first-order dynamics, thus losing the convergence benefits from second-order dynamics (Chen et al., 2014). In practice, σ_θ and σ_ξ are problem-specific, thus need to be tuned, *e.g.* , by cross-validation.

Stochastic resampling When generating samples from the stochastic process in (4.9), we resample momentum and thermostats from their marginal distribution with a fixed frequency, instead of every iteration from their conditionals. Since the momentum and thermostats are drawn from the independent marginals of stationary distribution $p(\Gamma) \propto \exp[-H(\Gamma)]$, it can be shown that reconstructing the stochastic process with the solution of the SDEs will still leave the stochastic process invariant to the target stationary distribution (Brunick et al., 2013).

To simplify the discussion, consider a stochastic process of a particle $\{\theta, p\}$ as in (4.7) with fixed ξ. As show in Figure 4.2(top), stochastic process with resampling helps sampler move quickly to a lower Hamiltonian contour. Suppose the initial value of θ is far from the maximum *a posteriori* value. The dynamics governed by (4.7) will stochastically move along the Hamiltonian contour. The total Hamiltonian energy level is affected by the joint effect of the stochastic diffusion and momentum refraction (*i.e.*, -ξpdt), which changes continuously over time.

From previous discussions, moving on a high Hamiltonian contour when $a > 1$ is less efficient because the absolute value of the momentum, $|p|$, will get increasingly large, slowing down the movement of θ. Resampling of momentum according to its marginal will enable the sampler to immediately move to a lower Hamiltonian energy level.

At the burn-in stage, this *momentum-accumulation/energy-drop* cycle seen in Figure 4.2(bottom) via resampling momentum can happen several times, until equilibrium is found. In other words, resampling decreases energy step-wise during burn-in stage. In practice, the resulting energy level is often much lower than initially, thereby delivering a more efficient and accurate dynamic updating.

The frequency of resampling from the marginal of the stationary distribution can have a direct impact on the mixing performance. Setting the frequency too high will result in a random-walk behavior. Conversely, with a low frequency resampling, the random-walk behavior is suppressed at a cost of fewer jumps between trajectories associated with different energy levels. It is advisable to increase the resampling frequency if the sampler is initialized on low-density (*e.g.* light-tailed) region.

The resampling step on p and ξ plays a role that is similar to adding a Langevin component to θ, in the sense that both improve convergence for $a > 1$. However, these two strategies (resampling and Langevin) are fundamentally different. We empirically observe that resampling is most helpful during burn-in, while the additional Langevin-style updates are more helpful with mixing during stationary sampling.

SGMGT The specifications described above constitute an SG-MCMC method for the SDEs in (4.7), which we call Stochastic Gradient Monomial Gamma Thermostat (SGMGT). We denote the SG-MCMC method with additional Brownian motion on θ and ξ in (4.9) as SGMGT-D (Diagonal), *i.e.*, with $\sigma_\theta > 0$ and $\sigma_\xi > 0$. The complete update scheme, with Euler integrator, for SGMGT is presented in the appendix B. Note that with $a = 1/2, \sigma_\theta = 0, \sigma_\xi \to 0, c \to \infty$, SGMGT-D recovers SGHMC as in Chen et al. (2014). Moreover, when

$a = 1/2, \sigma_\theta = 0, c \to \infty$, it becomes SGNHT as in Ding et al. (2014a).

We note that SGMGT-D improves upon SGNHT in three respects: (i) we introduce generalized kinetics, which provably yield lower autocorrelations than standard HMC, especially in multimodal cases; (ii) the additional stochastic noise on thermostat variables yields more efficient mixing; (iii) we use stochastic resampling to allow for faster interchange between different energy levels, thus alleviating sampling *stickiness*.

To the authors' knowledge, despite existing analysis for Langevin Monte Carlo (Bubeck et al., 2015; Dalalyan, 2016), rigorous analysis and comparison of the mixing performance of general SG-MCMC is very difficult, thus not yet established. Toward understanding the mixing performance of SGMGT-D, we argue that as the minibatch size increases, and the contribution of the diffusion in (4.2) decreases, the SGMGT-D will approach MGHMC, in which case, a large a will result in high stationary mixing performance, especially when sampling multimodal distribution, as theoretically shown by Zhang et al. (2016d). Although our experiments support our intuition, a more formal theoretical justification is needed. We leave this as interesting future work.

We observe empirically that when increasing the value of a, SGMGT-D may not always achieve superior mixing performance. One possible reason for this is a larger value of a induces "stiffer" behavior of $\exp[-K(p)]$ at $p = 0$, which typically requires a higher level of softening, thus higher rejection rates during the rejection sampling step. Also, when the dimensionality of p is higher, the rejection rate of the rejection sampling step increases (proportional to p). In such cases, the efficiency of the sampler decreases with large a. For these reasons, we limit our experiments to $a = \{1, 2\}$.

We clearly have more hyperparameters than SGNHT. In practice, we fix $M = I$, $a = \{1, 2\}$, and set the resampling frequency $T_p = T_\xi = 100$, which provides robust performance. Thus, only two additional hyperparameters are employed (σ_θ and σ_ξ) compared to SGNHT, and these parameters require further tuning. We use either validation or a hold-out set in our experiments.

More accurate numerical integrators Using a first-order Euler integrator to approximate the solution of the continuous-time SDEs in (4.9), leads to $O(h)$ errors in the approximate samples (Chen et al., 2016a). Alternatively, we can use the symmetric splitting scheme of Chen et al. (2016a) to reduce the order of the approximate error to $O(h^2)$. Details of the splitting used in this work are provided in the appendix B.

Convergence properties The SGMGT framework, as an instance of SG-MCMC, enjoys the same convergence properties of general SG-MCMC algorithms studied in Chen et al. (2015). It's worth to mention that on challenging problems the posterior may not be densely sampled to yield ideal posterior computation, and the asymptotic theory is being used as a useful heuristic. Specifically, it is of interest to quantify how fast the sample average, $\hat{\phi}_T$, converges to the true posterior average, $\bar{\phi} \triangleq \int \phi(\theta)\pi(\theta|X)\mathrm{d}\theta$, for $\hat{\phi}_T \triangleq \frac{1}{T}\sum_{t=1}^{T}\phi(\theta_t)$, where T is number of iterations. Here we make the same assumptions of Chen et al. (2015), and further assume that a first-order Euler integrator and a fixed stepsize are used.

Proposition 2. *For the proposed SGMGT and SGMGT-D algorithms, if a fixed stepsize h is used, we have:*

$$\textit{Bias: } \left|\mathbb{E}\hat{\phi}_T - \bar{\phi}\right| = O\left(1/(Th) + h\right) ,$$

$$\textit{MSE: } \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 = O\left(1/(Th) + h^2\right) .$$

This proposition indicates that with larger number of iterations and smaller step sizes, smaller bias and MSE bounds can be achieved. We note that these bounds have similar rates compared to other SG-MCMC algorithms such as SGLD, however, as we demonstrate below in the experiments, SGMGT and SGMGT-D usually converge faster than existing SG-MCMC methods.

In addition, for stochastic resampling, we can extend Proposition 2 to the following complementary results:

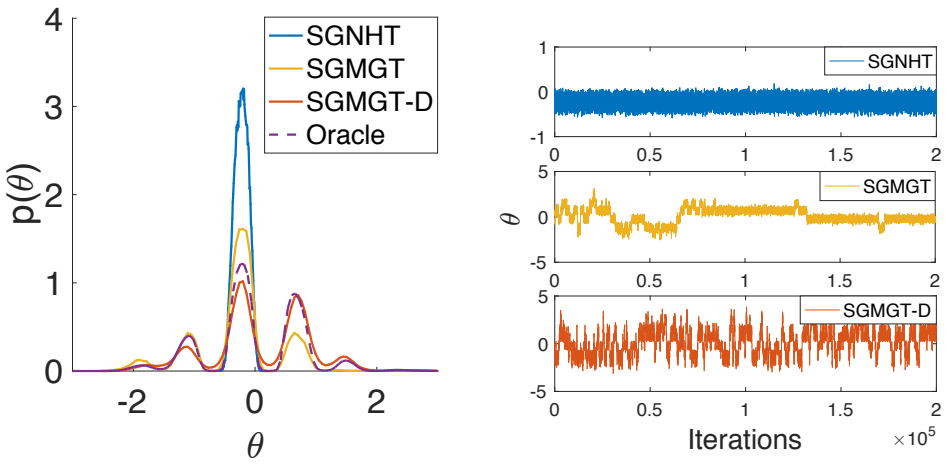


FIGURE 4.3: Synthetic multimodal distribution. Left: empirical distributions. Right: traceplot.

Lemma 3. *Let π_h be the stationary distribution of SGMGT-D. The stationary distribution of SGMGT-D with momentum resampling is the same as π_h.*

Lemma 4. *The optimal finite-time bias and MSE bounds for SGMGT-D with momentum replacement remain the same as SGMGT-D.*

Proofs of Lemma 1 and Lemma 2 are provided in the appendix B. The proposed SGMGT framework has a strong connection with second-order stochastic optimization methods, leading to a sampling scheme with minibatches with similar mixing performance as slice sampling (Neal, 2003). We discuss the details of this in the appendix B.

4.4 Experiments

4.4.1 Multiple-well Synthetic Potential

We first evaluate the mixing efficiency of SGMGT and SGMGT-D for a synthetic problem, to generate samples from a complex multimodal distribution. The distribution is shown in Figure 4.3(left). See appendix B for the definition of its potential energy. The modes are almost isolated with a low-density region connecting each other. Consequently, traversing

AUROC (D)	A (15)	G (25)	H (14)	P(8)	R (7)	C (87)
SGNHT	0.89	0.75	0.90	0.86	0.95	0.65
SGMGT(a=1)	0.92	0.78	0.91	0.86	0.87	0.70
SGMGT-D(a=1)	0.95	0.86	0.95	**0.93**	**0.98**	**0.73**
SGMGT(a=2)	0.93	0.79	0.93	0.88	0.86	0.62
SGMGT-D(a=2)	**0.95**	**0.90**	**0.95**	0.90	0.97	0.69
ESS (D)	A (15)	G (25)	H (14)	P(8)	R (7)	C (87)
SGNHT	869	941	1911	2077	1761	1873
SGMGT-D(a=1)	**3147**	**2131**	2448	**4244**	1494	**3605**
SGMGT-D(a=2)	2700	1989	**2768**	3430	**2265**	2969

Table 4.1: Average AUROC and median ESS. Dataset dimensionality in parenthesis.

between modes is difficult. In order to simulate the noise of the gradient estimates, we set $\nabla\tilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, 2B)$, similar to Ding et al. (2014a), where $B = 1$.

We compare SGNHT with SGMGT and SGMGT-D with monomial parameter $a = 2$ and fix $\gamma = 1$. For all three algorithms, we try a number of hyperparameter settings, *e.g.*, stepsize h, $\{\sigma_\theta, \sigma_p, \sigma_\xi\}$, and the soft parameter c, and present the best results in Figure 4.3. Standard SGNHT fails to escape from one of the modes of the distribution. For SGMGT with $a = 2$, the generated samples reached 3 modes. For SGMGT-D with $a = 2$, the sampler identified all 5 modes. In Figure 4.3(right), SGMGT-D adequately moves across different modes and yields rapid mixing performance, unlike SGMGT which exhibits stickier behavior.

4.4.2 Bayesian Logistic Regression

We evaluated the mixing efficiency and accuracy of SGMGT and SGMGT-D using Bayesian logistic regression (BLR) on 6 real-world datasets from the UCI repository (Bache and Lichman, 2013): German credit (G), Australian credit (A), Pima Indian (P), Heart (H), Ripley (R) and Caravan (C). The data dimensionality ranges from 7 to 87, and total observations vary between 250 to 5822. Gaussian priors are imposed on the regression coefficients. We set the minibatch size to 16. Other hyperparameters are provided in the

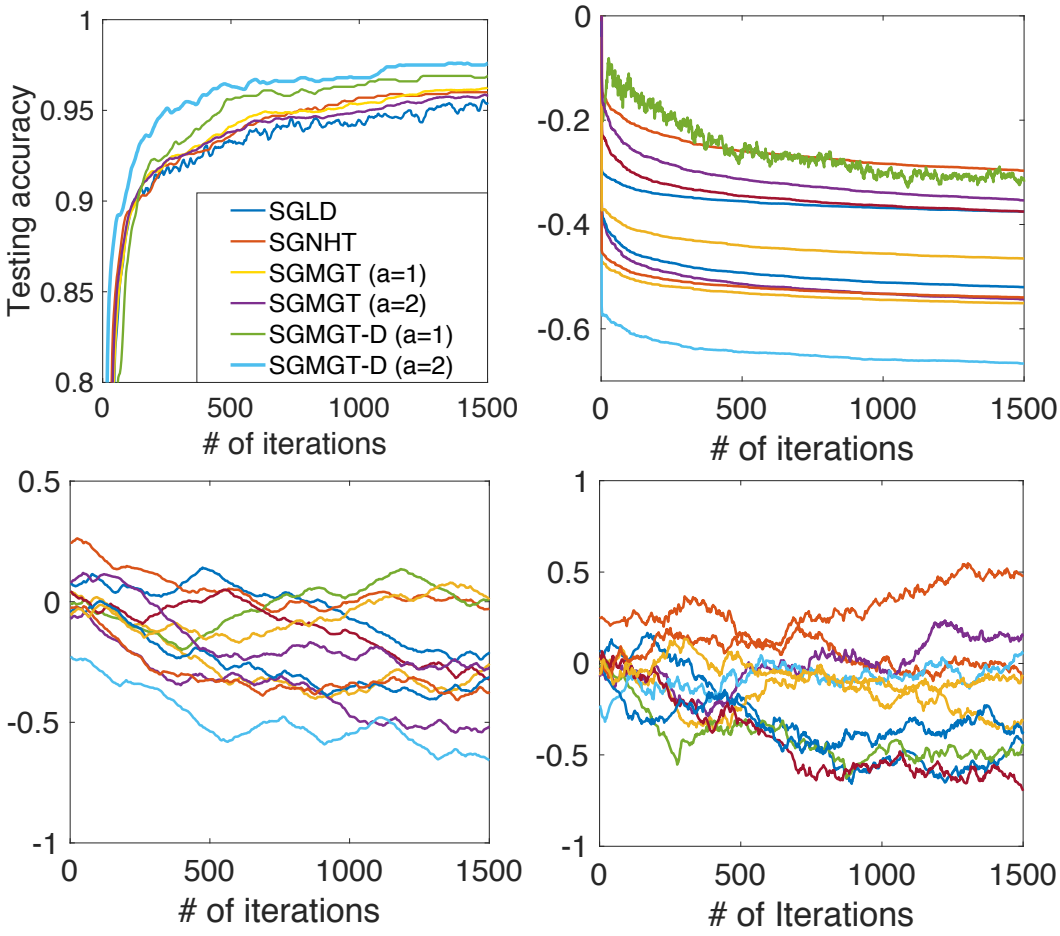


FIGURE 4.4: Experimental results for DRBM. Left: testing accuracies. Middle-left through right: traceplots.

appendix B. For each experiment, we draw 5000 iterations with 1000 burn-in samples.

Results in terms of median Effective Sample Size (ESS) and prediction accuracies as Area Under Receiver Operating Characteristic (AUROC) are summarized in Table A.10. All the results are averages over 5 independent runs with random initialization. In general, SGMGT-D performs better than SGMGT. For higher-dimensional dataset Cavaran, the performance of $a = 2$ decreases significantly, indicating numerical difficulties. The performance gap between SGMGT and SGMGT-D with $a = 1$ or $a = 2$ is usually larger than the gap between SGNHT ($a = 0.5$). Presumably when a is greater than 1, SGMGT-D has better convergence.

stepsize	0.04	0.05	0.06	0.07	0.08	0.09	0.1
SGLD	1058	1054	1058	1067	**1037**	1048	1057
SGNHT	1104	1144	1039	**1024**	1043	1067	1107
SGMGT(a=1)	996	988	990	**986**	996	998	997
SGMGT-D(a=1)	987	**983**	996	996	992	1013	1029
SGMGT(a=2)	1024	1029	1030	**1013**	1030	1022	1043
SGMGT-D(a=2)	968	994	973	957	961	**954**	970

Table 4.2: The test perplexity with varying stepsize.

4.4.3 Latent Dirichlet Allocation

We also test our methods on Latent Dirichlet Allocation (LDA) (Blei et al., 2003). Details of LDA and our implementation are provided in the appendix B. We use the ICML dataset (Chen et al., 2013), which contains 765 documents corresponding to abstracts of ICML proceedings from 2007 to 2011. After stopword removal, we obtain a vocabulary size of 1918 and about 44K words. We use 80% of the documents for training and the remaining 20% for testing. The number of topics is set to 30, resulting in 57,540 parameters. We use a symmetric Dirichlet prior with concentration $\beta = 0.1$. All experiments are based on 5000 MCMC samples with 1000 burn-in rounds. We set the minibatch size to 16. Other hyperparameter settings are provided in the appendix B.

Table 4.2 shows the test perplexities for SGMGT and SGMGT-D for different stepsizes. For each method we highlight the best perplexity. The SGMGT-D with $a = 2$ outperforms other methods, however SGMGT with $a = 2$ fails to achieve a comparable result with SGMGT with $a = 1$, probably because a good initialization is hard to achieve for a high-dimensional distribution.

4.4.4 Discriminative RBM

We applied our SGMGT to the Discriminative Restricted Boltzmann Machine (DRBM) (Larochelle and Bengio, 2008) on MNIST data. We choose DRBM instead of RBM because it provides explicit stochastic gradient formulas.

We evaluated our methods empirically and compare them with SGNHT. We use one hidden layer with 500 units. For each method we performed 1500 iterations with 200 burn-in samples. The minibatch size is set to 100. Details of the hyperparameter settings for SGMGT and SGMGT-D are provided in the appendix B. As shown in Figure 4.4(right-most 3 panels), we observe that SGMGT-D with $a = 2$ yields a superior mixing performance. For SGMGT-D with $a = 2$, the posterior samples demonstrated both rapid local mixing, and long-range movement. In contrast, SGLD seems trapped into a local mode after around 500 iterations.

Figure 4.4(left) shows that SGMGT-D with $a = 2$ delivers the fastest convergence with the highest test accuracy, 0.976. The SGMGT-D improves over SGMGT, while performance of SGMGT-D seems to increase with a large value of a. We observed that the stochastic resampling played a crucial role for SGMGT, as removing the resampling step resulted in a large drop in testing performance and mixing efficiency.

4.4.5 Recurrent Neural Network

We test our framework on Recurrent Neural Networks (RNNs) for sequence modeling (Gan et al., 2015d). We consider two tasks: (*i*) polyphonic music prediction; and (*ii*) word-level language modeling, detailed below. Additional details of the experiment are provided in the appendix B.

Polyphonic music prediction We use four datasets: Piano-midi.de (Piano), Nottingham (Nott), MuseData (Muse) and JSB chorales (JSB) (Boulanger-Lewandowski et al., 2012). Each of these are represented as a collection of 88-dimensional binary sequences, that span the whole range of piano from A0 to C8.

We use a one-layer LSTM (Hochreiter and Schmidhuber, 1997) model, and set the number of hidden units to 200. The total number of parameters is around 200K. Each model is trained for 100 epochs. We perform early stopping, while selecting the stepsize

and other hyperparameters by monitoring the performance on validation sets. Updates are performed using minibatches from one sequence.

Language modeling The Penn Treebank (PTB) corpus (Marcus et al., 1993) is used for word-level language modeling. We adopt the standard split (929K training words, 73K validation words, and 82K test words). The vocabulary size is 10K. We train a two-layer LSTM model on this dataset. The total number of parameters is approximately 6M. Each LSTM layer contains 200 units.

Algorithms	Piano	Nott	Muse	JSB	PTB
SGLD	11.37	6.07	10.83	11.25	127.47
SGNHT	9.00	4.24	7.85	9.27	131.3
SGMGT (a=1)	7.90	4.35	8.42	8.67	120.6
SGMGT (a=2)	10.17	4.64	8.51	8.84	250.5
SGMGT-D (a=1)	**7.51**	**3.33**	7.11	8.46	113.8
SGMGT-D (a=2)	7.53	3.35	**7.09**	**8.43**	**109.0**
SGD	11.13	5.26	10.08	10.81	120.44
RMSprop	7.70	3.48	7.22	8.52	120.45
SGD-M	8.37	4.46	8.13	8.71	120.44
ADAM	8.00	3.70	7.56	8.51	120.45

Table 4.3: Test negative log-likelihood on music datasets and test perplexities on PTB.

Results are shown in Table 4.3. The best log-likelihood results on the test set are achieved by using SGMGT-D with either $a = 1$ or $a = 2$ (depending on the dataset). To compare with optimization-based methods, we also include results for SGD (Bottou, 2010), SGD with momentum, ADAM(Kingma and Ba, 2014) and RMSprop (Tieleman and Hinton, 2012). A more comprehensive comparison is provided in the appendix B.

Learning curves for Nott and PTB datasets are shown in Figure 4.5. We omit the SGLD results since they are not comparable with other methods. For both datasets, we observe that SGMGT-D delivers fastest convergence. The best negative log-likelihood is achieved by SGMGT-D $a = 1$. The difference between $a = 1$ and $a = 2$ is small, though SGMGT-D with $a = 2$ seems to decrease slightly faster after 20 epochs for PTB data.

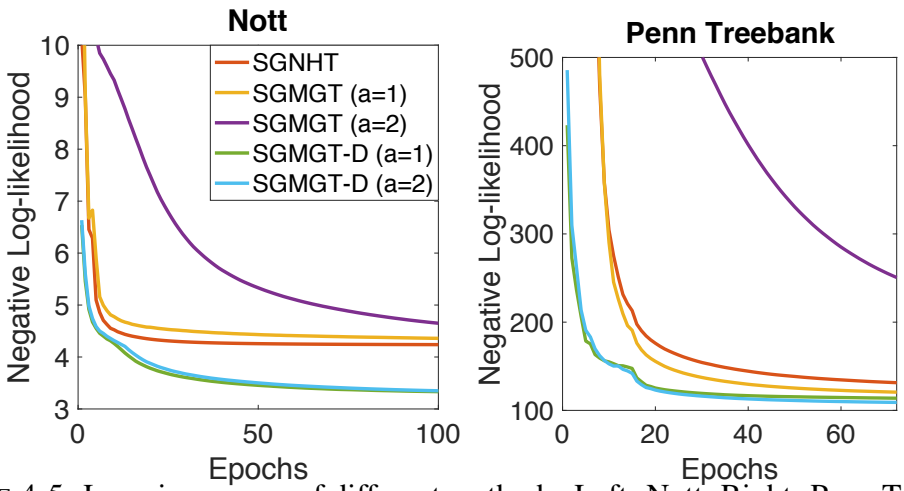


FIGURE 4.5: Learning curves of different methods. Left: Nott. Right: Penn Treebank.

We also observe that the SGMGT with $a = 2$ seems suboptimal compared with SGMGT with $a = 1$ and SGNHT. We hypothesize that numerical difficulties hinder the success of SGMGT with $a = 2$, especially in higher-dimensional cases, and without the additional Langevin components of SGMGT-D.

4.5 Summary comments

We improve upon existing SG-MCMC methods with several generalizations. We employed a more-general kinetic function, which we have shown to have better mixing efficiency, especially for multimodal distributions. Since practical use of the generalized kinetics is limited by convergence issues during burn-in, we injected additional Langevin dynamics and incorporated a stochastic resampling step to obtain generalized SDEs that alleviate the convergence issues. Possible areas of future research include designing an algorithm in a slice-sampling fashion, which maintains the invariant distribution by leveraging the connections between HMC and slice sampling (Zhang et al., 2016d). In addition, it is desirable to design an algorithm that can adaptively choose the monomial parameter a, thereby achieving better mixing while automatically avoiding numerical difficulties.

5

Scalable Bayesian analysis for discrete time-series biological date

5.1 Introduction

Probabilistic models for high-dimensional time series have long been an area of significant interest in machine learning. In this chapter, we introduce a novel dynamic model for discrete time-series data, in which the temporal sampling may be nonuniform. The applicability of this model spans data from different domains, such as music sequences, text streams and time-series of molecular data in *computational biology*.

Among this prior work, Hidden Markov Models (HMMs) (Rabiner and Juang, 1986) and the Linear Dynamical System (LDS) (Kalman, 1963) are particularly well understood. However, in some cases these models are limited by the type of dynamic structures they can capture. In fact, real-world time-series data are usually described by complex nonlinear temporal dependencies, while traditional LDS models are restricted to latent representations described by linear dynamics. Models with discrete latent spaces, such as the HMM, are often specified as mixture models that represent the history of a time-series using multinomial distributions.

A newer class of time-series models, better suited to model complex (nonlinear) probability distributions over high-dimensional sequences, rely either on Recurrent Neural Networks (RNNs) (Hermans and Schrauwen, 2013; Martens and Sutskever, 2011; Pascanu et al., 2013; Sutskever et al., 2013), Restricted Boltzmann Machines (RBMs) (Taylor et al., 2006; Sutskever and Hinton, 2007; Sutskever et al., 2009; Boulanger-Lewandowski et al., 2012; Mittelman et al., 2014) or Sigmoid Belief Networks (SBNs) (Gan et al., 2015a). The Temporal Restricted Boltzmann Machine (TRBM) (Sutskever and Hinton, 2007) and the Temporal Sigmoid Belief Network (TSBN) (Gan et al., 2015a), for instance, consist of a sequence of RBMs or SBNs, respectively, where the state of the current RBM or SBN is stochastically determined by previous RBMs or SBNs. Inference approaches for these models are non-trivial. Approximate procedures based mainly on Variational Bayes principles have been proposed and scale well to large datasets (Sutskever and Hinton, 2007; Mittelman et al., 2014; Gan et al., 2015a).

In the context of count data, multi-layer directed models are becoming increasingly popular (Mnih and Gregor, 2014; Gan et al., 2015e; Henao et al., 2015a; Gan et al., 2015a). In these models, the likelihood function connecting latent variables to observations is often specified in terms of Poisson distributions or softmax link functions, whereas latent (often deep) layers of the model are described by binary variables, capturing nonlinear dependencies and often modeled via SBNs (Mnih and Gregor, 2014; Gan et al., 2015e). For time-series data, a similar idea has been leveraged, where a discrete transition model akin to HMMs connects current binary latent variables to previous ones (at earlier time points), but deviates from HMMs by specifying transitions via SBNs (Gan et al., 2015a), not multinomial distributions.

In this context , we propose a model for time-series data where both emission and transition models are specified in terms of Poisson distributions, borrowing ideas from deep Poisson factor models (Henao et al., 2015a). In particular, the Poisson-based transition model treats transition *strengths* as latent counts that are transformed to binary variables

via the Bernoulli-Poisson Link (BPL) (Zhou, 2015), a recently proposed alternative to the sigmoid link function. The BPL yields efficient learning and inference algorithms (Zhou, 2015; Henao et al., 2015a). In particular, the key advantage of modeling transitions with the BPL is that learning and inference scales with the number of non-zero latent variables, as opposed to the number of states (like is the case of TRBM or TSBN models), where the sigmoid link function is employed (Sutskever and Hinton, 2007; Gan et al., 2015a).

We derive efficient inference via *augmented gibbs sampling*, which scales with the number of non-zeros in the data and latent binary states, as well as *stochastic gradient monomial Gamma sampler* (SGMGT) as developed in chapter 4. These scalable MCMC methods yield significant acceleration compared to related models. In comparison, the augmented gibbs sampling yields higher level of accuracy, while SGMGT enjoys better scalability and computational efficiency. The model is derived for count data but can be readily modified for binary observations. Experimental results on benchmark data show the proposed model achieves state-of-the-art predictive performance. The experiments on microbiome data demonstrate applicability of the proposed model to interesting problems in computational biology where interpretability is of utmost importance.

The highlights of this model are:

1. A dynamic model for times-series data with count observations based on Poisson factor analysis. The Bernoulli-Poisson link formulation can be readily accommodated to time-series with binary observations.

2. Unlike most previously proposed models, our formulation easily allows for modeling data *nonuniformly* sampled in time.

3. Two efficient and scalable inferences methods were applied.

   (a) First, an efficient exact augmented sampling inference procedure is developed, scaling with the number of non-zeros in the data and binary latent variables.

This allows our implementation to benefit from significant parallelization using GPUs (demonstrated in the experiments).

(b) Second, SGMGT is applied and compared with above exact method. Trade-offs are discussed.

4. Results on dynamic microbiome dataset highlight the benefits of our modeling strategy from the standpoints of performance and interpretability.

5.2 Dynamic Poisson factor analysis

Assume observed counts for N time-series, where the vector of counts at each time point is of dimension M. The vector of counts at time t for the nth time series is denoted as $\mathbf{x}_{nt} \in \mathbb{N}^M$. Akin to HMMs, we model the dynamics of the time-series by imposing a *transition* model on latent variables, $\mathbf{h}_{nt} \in \{0,1\}^K$, where the state of the current latent variable (at time t) depends on the previous state (at time $t-1$), and K is the number of latent variables. The model allows 2^K different realizations of $\mathbf{h}_{nt}$, yielding 2^K states. However, the proposed model characterizes transition dynamics by more than just these discrete states (moving beyond an HMM), yielding improved modeling flexibility and results, as detailed below.

The joint probability of the nth observation at time t is

$$p(\mathbf{X}_n, \mathbf{H}_n|\boldsymbol{\Xi}, \boldsymbol{\Omega}) = p(\mathbf{h}_{n0}) \prod_{t=1}^{T_n} p_{\boldsymbol{\Xi}}(\mathbf{x}_{nt}|\mathbf{h}_{nt})\, p_{\boldsymbol{\Omega}}(\mathbf{h}_{nt}|\mathbf{h}_{nt-1})\,, \tag{5.1}$$

where $\mathbf{X}_n = [\mathbf{x}_{n1}, \ldots, \mathbf{x}_{nT_n}]$, $\mathbf{H}_n = [\mathbf{h}_{n1}, \ldots, \mathbf{h}_{nT_n}]$ and T_n is the length of the nth time-series. Further, $p(\mathbf{h}_{n0})$ is the prior for the initial value of the latent variables, $p_{\boldsymbol{\Xi}}(\mathbf{x}_{nt}|\mathbf{h}_{nt})$ is the *emission* model with parameters $\boldsymbol{\Xi}$ and $p_{\boldsymbol{\Omega}}(\mathbf{h}_{nt}|\mathbf{h}_{nt-1})$ is the transition model with parameters $\boldsymbol{\Omega}$.

5.2.1 *Emission model*

For the observed vector $\mathbf{x}_{nt}$, containing counts of M entities (*e.g.*, a vocabulary of M words), we impose the following *emission* model

$$\mathbf{x}_{nt} \sim \text{Poisson}\left(\mathbf{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt})\right) , \tag{5.2}$$

where $\mathbf{\Psi} \in \mathbb{R}_+^{M \times K}$ is the *global* factor loadings matrix with K factors, shared by all time-series and time points; $\boldsymbol{\theta}_{nt} \in \mathbb{R}_+^K$ and $\mathbf{h}_{nt} \in \{0,1\}^K$ are *local* variables representing factor intensities and activations, respectively. Note that entries of $\mathbf{h}_{nt}$ indicate which factors are active for observation n at time t, *i.e.*, they define the *state* to which $\mathbf{x}_{nt}$ belongs. Symbol $\circ$ represents the element-wise (Hadamard) product.

Note that the Poisson emission parameters for data n are not just dependent on the state at time t, $\mathbf{h}_{nt}$. Binary activations, $\mathbf{h}_{nt}$, impose which columns of $\mathbf{\Psi}$ are employed to represent the Poisson rate, *and* the nonnegative intensities, $\boldsymbol{\theta}_{nt}$, provide temporal- and data-dependent scaling; in our experiments we have found this modeling flexibility to be important, particularly on real data, *e.g.*, the motivating microbiome data.

The model in (5.2) may be rewritten as

$$\begin{aligned} x_{mnt} = \textstyle\sum_{k=1}^K x_{mknt} \, , x_{mknt} \sim \ \text{Poisson}(\lambda_{mknt}) \, , \\ \lambda_{mknt} = \ \psi_{mk}\theta_{knt}h_{knt} \, , \end{aligned} \tag{5.3}$$

where x_{mnt} is component m of vector $\mathbf{x}_{nt}$, ψ_{mk} is component m of $\boldsymbol{\psi}_k$, $\boldsymbol{\psi}_k$ is column k of $\mathbf{\Psi}$, θ_{knt} is component k of vector $\boldsymbol{\theta}_n$, and h_{knt} is component k of vector $\mathbf{h}_n$. In (5.3) we have used the additive property of the Poisson distribution to decompose the mth observed count of $\mathbf{x}_{nt}$ as K latent counts, $\{x_{mknt}\}_{k=1}^K$. This decomposition allows derivation of efficient inference for the entire model, as discussed in Section 5.3.

We specify prior distributions for the model in (5.2) as previously described (Zhou et al., 2012), *i.e.*,

$$\begin{aligned} \boldsymbol{\psi}_k \sim \text{Dirichlet}(\eta_\psi \mathbf{1}_M) \, , \theta_{knt} \sim \ \text{Gamma}(r_k, b_\theta) \, , \\ h_{knt} \sim \ \text{Bernoulli}(\pi_{knt}) \, , \end{aligned} \tag{5.4}$$

where $\mathbf{1}_M$ is an M-dimensional vector of all-ones. Favoring simplicity, we let $\eta_\psi = 1/K$, $b_\theta = 0.5$ and $r_k \sim \text{Gamma}(1, 1)$. Prior distributions for η_ψ and b_θ that result in closed form conditionals exist, and can be used if desired; see for instance (Escobar and West, 1995) for η_ψ, and (Zhou and Carin, 2015) for b_θ.

The hierarchical model implied by (5.2) and (5.4), known as Poisson Factor Analysis (PFA), corresponds to the emission model, $p_{\boldsymbol{\Xi}}(\mathbf{x}_{nt}|\mathbf{h}_{nt})$, succinctly expressed in (5.1), with parameters $\boldsymbol{\Xi} = \{\boldsymbol{\Psi}, \boldsymbol{\theta}_{nt}, r_k\}$. These parameters can be interpreted in the context of topic modeling as follows: $\boldsymbol{\Psi}$ is a loadings matrix whose columns encode K topics (distributions over M words), that capture the correlation structure of observed variables; binary latent activations $\mathbf{h}_{nt}$ select which topics are used in time-series n at time t; and $\boldsymbol{\theta}_{nt}$, encode the intensities with which each topic is manifested in observation $\mathbf{x}_{nt}$. Interestingly, the PFA is closely related to other well-known topic modeling approaches, such as latent Dirichlet allocation, hierarchical Dirichlet processes and focused topic models (Zhou and Carin, 2015).

5.2.2 Transition model

The most unique aspect of the proposed model is how state transitions are modeled, and how this allows nonuniform temporal sampling. In order to specify our model for latent variable transitions with respect to time, we first introduce the Bernoulli-Poisson link (Zhou, 2015), a recently proposed probabilistic link function particularly useful at relating binary and count variables. Specifically, for a binary vector $\mathbf{h}_{nt}$ with elements h_{knt},

$$h_{knt} = 1\left(z_{knt} > 0\right), z_{knt} \sim \text{Poisson}\left(\tilde{\lambda}_{knt}\right), \tag{5.5}$$

where z_{knt} is a latent count associated with binary variable h_{knt}, parameterized by a Poisson distribution with rate $\tilde{\lambda}_{knt}$. The function $1(\cdot)$ is defined as $1(\cdot) = 1$ if the argument holds, and $1(\cdot) = 0$, otherwise. The model in (5.5), denoted here for short as $\mathbf{h}_{nt} \sim \text{BPL}(\tilde{\boldsymbol{\lambda}}_{nt})$,

for $\tilde{\boldsymbol{\lambda}}_{nt} \in \mathbb{R}_+^K$ with elements $\tilde{\lambda}_{knt}$, has the interesting property that

$$p(h_{knt} = 1) = \text{Bernoulli}(\pi_{knt}) \, , \pi_{knt} = \, 1 - \exp\left(-\tilde{\lambda}_{knt}\right) \, . \tag{5.6}$$

The result in (5.6) can be shown by marginalizing out latent counts, z_{knt}, in (5.5). In fact, to sample h_{knt} we do not need to instantiate latent count z_{knt}, but the rate of its underlying Poisson distribution, $\tilde{\lambda}_{knt}$.

The distribution implied by (5.6) is reminiscent of the complementary log-log link function (Piegorsch, 1992; Collett, 2002), where $\tilde{\lambda} = \exp(-u)$ and $u \in \mathbb{R}$. The logistic link function used in RBMs and SBNs is symmetric around the origin, $u = 0$, with $p(h = 1) = \text{Bernoulli}(\pi)$, where $\pi = 1/(1 + \exp(-u))$; by contrast, the proposed BPL link is *asymmetric*, which makes it appropriate for very sparse settings, where the proportion of zeros is large. In our case, this is particularly useful, because we can increase the number of latent variables without forcing the model to increase the number of states *a priori*.

Having defined the Bernoulli-Poisson link, it becomes clear how to specify a transition model using the same Poisson factor analysis framework used for the emission model. Specifically, we write

$$\begin{aligned} \mathbf{h}_{nt} &\sim \text{BPL}\left(\tilde{\boldsymbol{\lambda}}_{nt}\right) \, , \\ \boldsymbol{\phi}_k &\sim \text{Dirichlet}\left(\eta_\phi \mathbf{1}_K\right) \, , w_{knt-1} \sim \, \text{Gamma}(s_k, b_w) \, , \end{aligned} \tag{5.7}$$

where $\tilde{\boldsymbol{\lambda}}_{nt} = \tau_{nt}^{-1} \boldsymbol{\Phi}(\mathbf{w}_{nt-1} \circ \mathbf{h}_{nt-1}) + \tilde{\boldsymbol{\lambda}}_0$ as in (5.5) and (5.6), $\boldsymbol{\phi}_k$ is a column of $\boldsymbol{\Phi}$ (transition factor matrix), w_{nkt-1} is an element of $\mathbf{w}_{nt-1}$ (*local* variable representing transition factor intensity), and η_ϕ, b_w and s_k are specified in a similar fashion to the emission model in (5.4). Bias term, $\tilde{\boldsymbol{\lambda}}_0$, controls the base rate of the Poisson distribution in (5.5) and is specified below. Parameter τ_{nt} is the time difference between observations t and $t-1$, for time-series n; τ_{nt} can vary with n and t, allowing *nonuniform* temporal sampling.

The specification in (5.7) corresponds to the transition model, $p_{\boldsymbol{\Omega}}(\mathbf{h}_{nt}|\mathbf{h}_{nt-1})$, in (5.1), with parameters $\boldsymbol{\Omega} = \{\boldsymbol{\Phi}, \mathbf{w}_{nt}, s_k, \tilde{\boldsymbol{\lambda}}_0\}$, for $n = 1, \ldots, N$, $t = 1, \ldots, T_n$ and $k = 1, \ldots, K$.

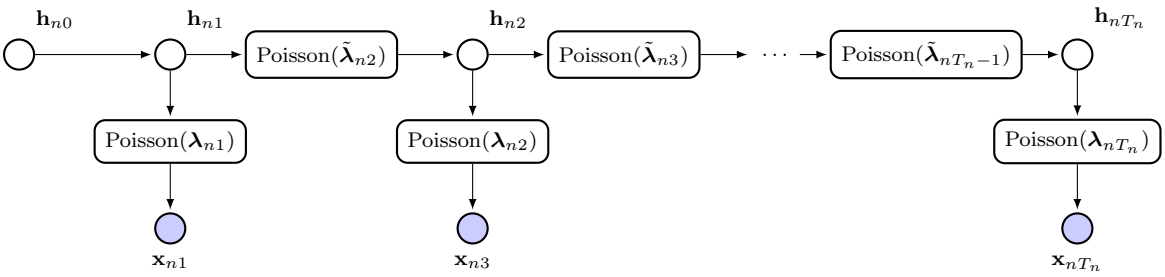


FIGURE 5.1: Graphical model for dynamic Poisson factor analysis.

These parameters have clear interpretations. For instance, $\boldsymbol{\Phi}$ is a transition matrix whose columns, $\boldsymbol{\phi}_k$, encode distinct transition *templates*. These templates are K-dimensional probability vectors and can be viewed as distributions over latent binary variable activations, $\mathbf{h}_{nt}$. Each template defines a particular activation pattern; some latent variables are co-active with high probability, while others are jointly absent, *i.e.*, they define correlation structure among elements of $\mathbf{h}_{nt}$. Interestingly, templates at time t are selected by previous activations $\mathbf{h}_{nt-1}$, and regulated (weighted) by previous intensities $\mathbf{w}_{nt-1}$. This implies that at time t the transition statistics are not only dependent on the 2^K discrete states $\mathbf{h}_{nt}$, but scale these states with weights $\mathbf{w}_{nt-1}$, adding modeling flexibility analogous to that in the emission model discussed above. In addition, the correlation between two adjacent time-points decays inversely with proximity in time, *i.e.*, as τ_{nt} increases, the dependency between latent variables $\mathbf{h}_{nt}$ and $\mathbf{h}_{nt-1}$ decreases. When τ_{nt} is large enough, binary activations loose their time dependency and effectively become a stochastic function of $\tilde{\boldsymbol{\lambda}}_0$, in fact, from (5.6)

$$p(h_{knt} = 1) = \text{Bernoulli}\left(1 - \exp(\tilde{\lambda}_{k0})\right) ,$$

where $\tilde{\lambda}_{k0}$ is an element of $\tilde{\boldsymbol{\lambda}}_0$.

Note that the fact that intensities are time-dependent adds flexibility to the model compared to a simplified specification where these intensities are global parameters, *i.e.*, w_{kn} instead of w_{knt}. We observed empirically that the simplified model (with time-independent weights) does not perform as well as the specification in (5.7), however it is worth mention-

ing that time-dependent intensities may undermine the contribution of τ_{nt} to the transition model.

The emission model is completed by specifying a prior distribution for the initial state of the latent variables of the dynamic model, $p(\mathbf{h}_{n0})$, in (5.1). We let

$$h_{kn0} \sim \mathrm{BPL}\left(\tilde{\lambda}_{k0}\right), \tilde{\lambda}_{k0} \sim \mathrm{Gamma}(a, b). \tag{5.8}$$

For simplicity, we let $a = b = 1$, so that elements of $\mathbf{h}_{n0}$ are approximately uniform. Note that (5.8) is a special case of (5.7) because since we do not have past information about $\mathbf{h}_{n0}$, conceptually, $\tau_{n0} \to \infty$.

In summary, the joint distribution for our dynamic Poisson factor model in (5.1) is fully specified by the emission model implied by (5.2) and (5.4), the transition model implied by (5.5) and (5.7), and the initial-state probability as in (5.8). Figure 5.1 shows a graphical representation of the model in terms of emission and transition rates. For the transitions, $\boldsymbol{\lambda}_{nt} = \boldsymbol{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt})$, and for the transitions, $\tilde{\boldsymbol{\lambda}}_{nt} = \tau_{nt}^{-1}\boldsymbol{\Phi}(\mathbf{w}_{nt-1} \circ \mathbf{h}_{nt-1}) + \tilde{\boldsymbol{\lambda}}_0$, as defined in (5.2) and (5.7), respectively. Filled and empty nodes represent observed and latent variables, respectively.

5.2.3 Binary data

We can easily extend the dynamic Poisson factor model described above to model binary time-series data, by leveraging the same type of construction used for the transition model, *i.e.*, for $\mathbf{x}_{nt} \in \{0, 1\}^M$, we can let

$$\mathbf{x}_{nt} \sim \mathrm{BPL}\left(\boldsymbol{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt})\right),$$

as in (5.7) but with prior distributions defined as in (5.4).

5.3 Learning and Inference

5.3.1 Augmented Gibbs sampling

The dynamic Poisson factor model defined in the previous section has the convenient property of having all its conditional posteriors available in closed form, due to local conjugacy. In this paper, we focus on Markov Chain Monte Carlo (MCMC) via Gibbs sampling for learning and inference. Stochastic variational inference may be also readily implemented using ideas from (Henao et al., 2015a). Other alternatives for scaling up inference based on gradient-based approaches (Welling and Teh, 2011b; Chen et al., 2014; Ding et al., 2014b) are also amenable to our model specification, however beyond the scope of this paper.

During the *learning* phase, Gibbs sampling for the model in (5.2), (5.4), (5.5), (5.7) and (5.8) involves sampling in sequence from the conditional posterior of all the global parameters of the model, namely, $\{\boldsymbol{\Psi}, r_k, \boldsymbol{\Phi}, s_k, \tilde{\lambda}_{k0}\}$, for $k = 1, \ldots, K$. During the *inference* phase, we sample from the conditional posterior of all local parameters, while also sampling from the *fixed* conditional posterior of the global parameters given the training data (obtained during learning). The local parameters include all latent variable activations, $\{\mathbf{h}_{nt}\}$, intensities for the emission model, $\boldsymbol{\theta}_{nt}$, and intensities for the transition model, $\mathbf{w}_{nt}$, where $n = 1, \ldots, N$, $t = 1, \ldots, T_n$. For prediction tasks, we perform learning on a training set, then perform inference on the test set, while sampling from the learned conditional posterior of the global parameters conditioned only on training data. In this way, we can benefit form model averaging at test time.

The hyperparameters of the model are set to fixed values: $\eta_\psi = \eta_\phi = 1/K$, $b_\theta = b_w = 0.5$ and $a_0 = b_0 = 1$. Note that priors for η, b, a_0 and b_0 exist that result in Gibbs-style updates, and can be readily incorporated into the model if desired; however, our priority is keeping the model simple without sacrificing flexibility. The updates for conditional posteriors are shown in the Appendix.

An important property from our dynamic Poisson factor model is that inference does not scale with the size of the data, $\{M, N, T_n\}$, for $n = 1, \ldots, N$, and the number of factors, K, but as a function of their non-zero elements, which is tremendously advantageous in cases where the data is sparse, which is often the case. In order to show that this scaling behavior holds, it is enough to see that by construction, from (5.3), if $x_{mnt} = \sum_{k=1}^{K} x_{mknt} = 0$ (or z_{mnt}), thus $x_{mkn} = 0$, $\forall k$ with probability 1. From (5.5) we see that if $h_{knt} = 0$ then $z_{knt} = 0$ with probability 1. As a result, update equations for all parameters of the model except for binary activations $\mathbf{h}_{nt}$, depend only on non-zero elements of $\mathbf{x}_{nt}$ and $\mathbf{z}_{nt}$. Updates for the binary variables can be cheaply obtained in block from $h_{knt} \sim \text{Bernoulli}(\pi_{knt})$ via λ_{knt}, as previously described in (5.6).

The most computationally expensive operation in our inference procedure is sampling from the multinomial distribution, to obtain latent counts for x_{mknt} (and z_{knt}), $\forall m, k, n, t$. Fortunately, conditioned on data $\mathbf{x}_{nt}$ and emission rates $\boldsymbol{\lambda}_{nt}$ (and $\tilde{\boldsymbol{\lambda}}_{nt}$ for transitions), $\forall n, t$, these can be sampled in block using a heavily parallelized implementation of the multinomial sampler via GPUs. Results of this efficient implementation are shown in the experiments.

5.3.2 SGMGT-embedded Gibbs sampling

Alternatively, the SGMGT algorithm introduced in chapter 4 can be readily applied to the dynamic Poisson factor model to yield additional computational saving. We consider employing SGMGT for conditional posterior sampling within a block Gibbs sampling procedure, due to the fact that SGMGT cannot address discrete variable sampling. Specifically, after each iteration of sampling hidden variables $\mathbf{h}, \mathbf{z}$, a fixed-step SGMGT procedure is applied for generating samples of $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\theta}, \boldsymbol{\Psi}, \mathbf{w}, \boldsymbol{\Phi}, \tilde{\boldsymbol{\lambda}}_0\}$. The *conditional* posterior can be written as

$$\pi(\boldsymbol{\Theta}|\mathbf{h}, \mathbf{z}) \propto p(\boldsymbol{\Theta}) p(\mathbf{x}|\mathbf{h}, \boldsymbol{\Theta}) p(\mathbf{h}, \mathbf{z}|\boldsymbol{\Theta}).$$

Where $p(\mathbf{\Theta})$ denotes the prior distribution. $p(\mathbf{x}|\mathbf{h}, \mathbf{\Theta})$ and $p(\mathbf{h}, \mathbf{z}|\mathbf{\Theta})$ is given by (5.2) and (5.7), respectively. Omitting constant,

$$\log p(\mathbf{x}|\mathbf{h}, \mathbf{\Theta}) = \sum_{n,t} -\mathbf{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt}) + x_{nt} \log[\mathbf{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt})]$$

$$\log p(\mathbf{h}, \mathbf{z}|\mathbf{\Theta}) = \sum_{n,t} -\tilde{\boldsymbol{\lambda}}_{nt} + z_{knt} \log[\tilde{\boldsymbol{\lambda}}_{nt}]$$

where $\tilde{\boldsymbol{\lambda}}_{nt}$ is defined in (5.7). Note that the conditional log-posterior is fully differentiable. By leveraging the posterior gradient, A fixed-step SGMGT updates for approximate sampling $\pi(\mathbf{\Theta}|\mathbf{h}, \mathbf{z})$ using a randomly sampled sub-collection of training data is performed. The sampling can also be performed over the longitude dimension of the data. We use 20 SGMCMC steps in our experiments, which is still significantly faster than applying conditional sampling. We observed that the SGMGT sample give very low 20-lag autocorrelation, which indicates the final sample has low correlation with the initialization point.

5.4 Comparison to previous work

Our work focuses on directed generative models. Most of the existing work on directed models is based on the Sigmoid Belief Network (SBN) (Neal, 1992), which only recently has shown potential for building large multi-layer (deep) and dynamic models (Mnih and Gregor, 2014; Gan et al., 2015c,d,a). Our model is most related with the Temporal SBN (TSBN) of (Gan et al., 2015a), in which the emission model is an SBN or a softmax belief network for binary and count time-series, respectively, and the transition model is an SBN. The TSBN delivers fast inference via the Neural Variational Inference and Learning (NVIL) algorithm (Mnih and Gregor, 2014). Our model is different from TSBN in three ways: (i) We use Bernoulli-Poisson links, not sigmoid links, for fast inference. (ii) We model observed counts directly using Poisson distributions, as opposed to observed count proportions like in TSBN, via softmax link. (iii) Our inference approach scales with the number of non-zeros in the data and binary latent variables, which is not possible for models

based on SBNs (or RBMs).

The work presented here is closely related to the deep Poisson factor model (Henao et al., 2015a), a multi-layer (deep) topic model in which adjacent layers are connected via BPLs similar to our model. The main differentiator between the deep PFA and our dynamic PFA is that in the former, BPLs are introduced as a way to model correlation across latent variables whereas in the latter (ours), BPLs provide us with a way to model correlation across observations, by coupling adjacent time-points (transition model).

To the best of our knowledge there is only one existing approach for time-series modeling based on Poisson factor analysis (Acharya et al., 2015). In their work, dynamics are imposed using a linear model on the intensities, θ_{nt}, in terms on previous intensities θ_{nt-1} via a gamma distribution specification, $w_{knt} \sim \mathrm{Gamma}(w_{knt-1}, b_\theta)$. Our model is different in that we learn correlations across binary latent variables using $\mathbf{\Phi}$, whereas in (Acharya et al., 2015) latent variables are *i.i.d. a priori* and restricted to linear dynamics. Unlike ours their model does not model latent variable activations, *i.e*, their model is *dense*. Note that our model does not impose dynamic structure on the latent intensities, however, we verified empirically that adding a specification like that of (Acharya et al., 2015) to our model does not improve the performance. We believe that this is the case because our model already allows for capturing complex nonlinear temporal dynamics, thus the addition of linear dynamics to the intensities does not meaningfully impact the model representation abilities.

All prior work on dynamic models using either RBMs, SBNs and Poisson factor analysis assumed uniform temporal sampling. The approach to nonuniform sampling introduced here specifies the dynamics of the transition model in terms of Poisson rates $\tilde{\boldsymbol{\lambda}}_{nt} = \tau_{nt}^{-1}\mathbf{\Phi}(\mathbf{w}_{nt-1} \circ \mathbf{h}_{nt-1}) + \tilde{\boldsymbol{\lambda}}_0$; the explicit dependence on sampling delay τ_{nt} and scaling $\mathbf{w}_{nt-1}$ moves well beyond only depending on the 2^K variants of the states $\mathbf{h}_{nt-1}$.

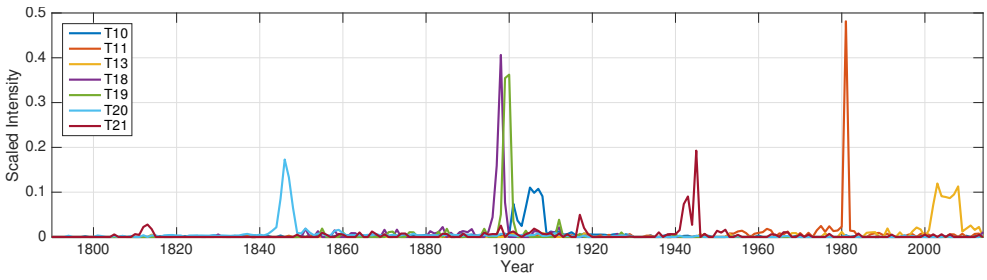
5.5 Experiments

We present extensive experiments on artificial dataset and several standard datasets for evaluation and comparison. Here we present the state of the union dataset to exemplify the effectiveness of our methods, the details of other experiments are provided in the appendix C. Finally, we consider a microbiome dataset composed of measurements of human gut microbiota over time from 6 subjects spanning 3 different studies (David et al., 2014) (sampled nonuniformly in time). The source code of our dynamic Poisson factor model will be available upon publication.

5.5.1 State of the Union

This dataset contains transcripts of $T = 225$ US presidential State of the Union addresses, ranging from years 1790 to 2014. We consider the dataset as a single time-series ($N = 1$) with 225 time-points, where each transcript is a document, thus one document per year ($\tau_{nt} = 1$, $\forall t$ and $n = 1$). We preprocess the data lightly by removing stop words and terms that occur less than 7 times in one document or less that 20 times overall, which results in a vocabulary of size $M = 2375$ terms. This preprocessing scheme was previously used by (Gan et al., 2015a) in their experiments with TSBNs.

Quantitative evaluation For prediction tasks, we exclude the last year (2014) from the learning phase of the model. For all the other documents ranging from year 1790 to 2013, we randomly partition words from each document into a 80%/20% split. The learning phase of the model is performed on the 80% subset, whereas the remaining 20% observations are used during inference to make predictions at each year. The predictions from both held-out sets are ranked according to their average predicted counts using the emission model in (5.2). For exact Gibbs inference we run 900 iterations of the Gibbs sampler during learning and average predictions over 100 collection samples during inference. For SGMGT we performed 20 steps, the minibatch size is set to be 128. The rest of the setting



T10	T11	T13	T18	T19	T20	T21
law	energy	iraq	cuba	islands	mexico	war
business	administration	terrorists	spanish	island	texas	enemy
conditions	percent	america	spain	phillipines	war	forces
labor	nuclear	terror	island	products	mexican	production
matter	programs	women	claim	convention	army	fighting
national	development	afghanistan	international	imperial	peace	japanese
industrial	policy	al	april	military	territory	german

FIGURE 5.2: Selected topics learned from the State of the Union dataset.

are identical to exact Gibbs sampler. We verified empirically that increasing the number of Gibbs iterations does not significantly change results.

To evaluate the prediction performance, we calculate the precision @top-L as in [31], which is defined as the fraction of the top-L words, predicted by the model, that match the true ranking of the observed word counts. We use $L = 50$ as in (Gan et al., 2015a). We compare our dynamic Poisson model against three recently proposed and related models: Gamma Process Dynamic Factor Analysis (GP-DPFA) (Acharya et al., 2015), Dynamic Rank Factor Model (DRFM) (Han et al., 2014) and Temporal Sigmoid Belief Network (TSBN) (Gan et al., 2015a). The parameters of each of these models were selected to maximize performance. For DRFM and TSBN we used 25 latent variables and for GP-DPFA and our model, 100 latent variables. Note that unlike GP-DPFA, our model has latent binary activations that allow for model size selection, thus $K = 100$ can be seen as a lower bound on the total number of active latent variables.

For this dataset we observed that the model estimated non-trivial activations for an average of 24 latent variables; see qualitative evaluation below for further details.

We summarize predictive performance results in Table 5.1. Mean precision was computed on held-out subsets of each year. Predictive precision was computed on the last year,

Model	Mean Precision	Predictive Precision
Dynamic PFA (Gibbs)	0.382	0.520
Dynamic PFA (SGMGT)	0.367	0.442
TSBN	0.327	0.353
GP-DPFA	0.223	0.189
DRFM	0.217	0.177

Table 5.1: Average predictive precision for STU dataset.

2014. We report Mean Precision over all documents (years) in the dataset, restricted to the 20% held-out sets, one per year. We also report Predictive Precision for the final year, 2014. We see that our model significantly outperforms all the other methods in terms of mean precision and is comparable to TSBN in terms of predictive precision. In terms of the inference method, the exact Gibbs sampler achieved highest testing prediction precision, while the SGMGT embedded Gibbs sampler uses around 1/3 of the running time of exact Gibbs sampler to reach a plateau of testing predictive precision.

Qualitative evaluation We now examine some of the parameters learned by the model to highlight the interpretability of our model. By looking at the conditional posterior of r_k, the global shape of intensities, θ_{kn}, we verified that the model only uses 24 latent variables from the $K = 100$ set *a priori*, see Figure 5.2(Top-left), which shows posterior mean of the shape parameter, r_k, for latent intensities, θ_{kn}. We show the top-30 largest values sorted in decreasing order.

This means that the reminding 76 latent variables are not active, $h_{knt} = 0$, or their intensities are close zero, $\theta_{knt} \approx 0$, at any given time point. In Figure 5.2(Top-right) we show scaled intensity traces (θ_{knt}, $\forall t$) and top words (largest weights of $\boldsymbol{\psi}_k$), respectively, from 7 of the 24 active topics estimated by the model. Scaling was done only to improve visualization. For figure 5.2(Bottom), we show top words from selected topics. Each column represents the top-10 largest weights from 7 columns, $\boldsymbol{\psi}_k$, of loadings matrix, $\boldsymbol{\Psi}$, with corresponding intensity traces shown above. We observe very relevant topics, tightly localized in time. We see for example that Topic 10 is related to the organized labor

movement, Topic 11 with the National Energy Program, Topics 13, 18, 19 and 20 focus on the Middle East, Spanish, Phillipines and Mexican wars, respectively. Finally Topic 21 is related to the world wars. Topics not shown are less localized in time and range from foreign policy to internal economic affairs.

5.6 Latent dynamic factor analysis of gut microbiome data

We finally applied our method on a microbiome dataset composed of longitudinal measurements of human gut microbiota over time, from 6 subjects spanning 3 different studies, with 2 subjects per study. Details about sample collection and processing are detailed in (David et al., 2014). Data are produced by DNA sequencing of microbiota samples, followed by processing and mapping of raw DNA reads into Operational Taxonomic Units (OTUs). Each OTU defines a species or a group of species, and is commonly used as unit of microbial diversity and is represented in the data as a count, which is a proxy for OTU concentration. The total number of OTUs per subject is shown in Table 5.2. The sparsity level of these data is on average 85% (most OTUs are not observed at a given point in time), and this sparsity is leveraged in the proposed model (via the BPL link) to yield significant computational acceleration.

Importantly, these data are sampled *nonuniformly* in time, and this complexity motivated several aspects of the proposed model (as detailed when presenting the model). All of the previous models against which we compared in the previous examples are not applicable here, as they assume uniform temporal sampling. Hence, in this experiment we focus on results based upon our proposed model. As in the other experiments, we run 900 Gibbs iterations during learning, 100 collections samples during inference and we set the number of latent variables to $K = 50$. For SGMGT we use 20 fixed step with 64 minibatch. Each dataset, has time-series of different time lengths, sampling intervals and OTU resolutions (see Table 5.2). The results are shown in terms of correlation. Error bars for dynamic PFA

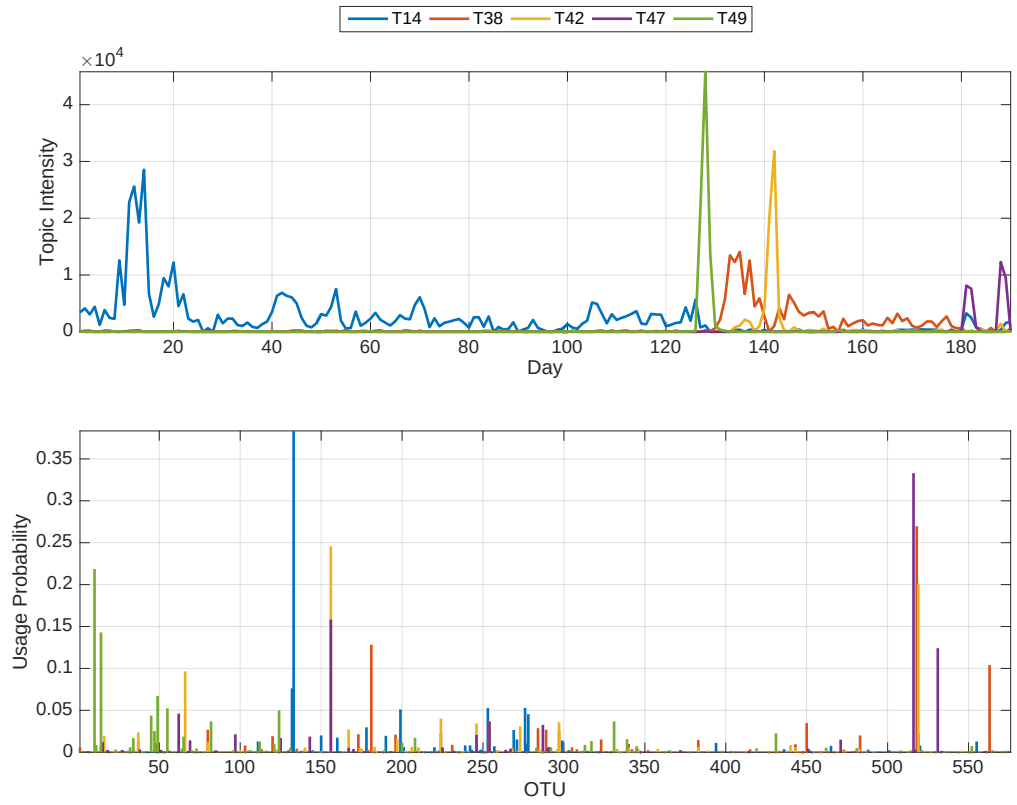


FIGURE 5.3: Selected topics learned from microbiome data and particular to subject S5.

correspond to correlation averages over 100 posterior collection samples during inference. In particular, data for which $T = 30$ were sampled once a day, whereas all the others where sampled nonuniformly over a total period of a year (David et al., 2014).

Sample	M	T	Dynamic PFA(gibbs)	Dynamic PFA(SGMGT)	Naive
S1	5432	321	0.880±0.008	0.866±0.011	0.8614
S2	5432	189	0.755±0.044	0.613±0.067	0.378
S3	9371	30	0.989±0.003	0.951±0.005	0.990
S4	9371	30	0.964±0.006	0.966±0.009	0.960
S5	33750	332	0.943±0.003	0.940±0.007	0.935
S6	33750	129	0.975±0.002	0.980±0.005	0.943

Table 5.2: One-step ahead forecasting results on microbiome data.

We have highlighted above aspects of the transition-statistics model that are motivated by nonuniform temporal sampling. We here also emphasize components of the emission statistics that are driven by the motivating microbiome example. Recall the emission model,

$\mathbf{x}_{nt} \sim \text{Poisson}\left(\mathbf{\Psi}(\boldsymbol{\theta}_{nt} \circ \mathbf{h}_{nt})\right)$, where $\mathbf{h}_{nt}$ defines which of the 2^K states characterizes time point t in sample n, with $\boldsymbol{\theta}_{nt}$ providing a time- and sample-dependent *scaling* of the factors that are "on" (those having non-zero values in $\mathbf{h}_{nt}$). In microbiome data, the *absolute* counts of OTUs may depend on the sample size and other aspects of the samples, and hence the scaling flexibility provided by $\boldsymbol{\theta}_{nt}$ is important. This is analogous to analyzing a corpus of documents for which the absolute size of the documents may vary widely.

We also observed that the Bernoulli-Poisson link is particularly useful in our microbiome dataset analysis. One well-known fact in omics studies in general is that less enriched microbiome species may have greater impact to the host, compared with species with high abundance (Qin et al., 2010). In many cases, the *"existence"* of a species is more important than the exact value representing its *"abundance"*. Similar to logit and probit link functions, the non-linearity manifested by BPL latent variables accounts for the saturation effect found in omics data, by diminishing the probability improvement per increment of abundance counts, as counts become large. As a consequence, the model is sensitive to low abundant species while is robust under the scenario where the difference in abundance among species is large. Meanwhile, unlike the logit and probit link, the BPL is asymmetric, which is beneficial when the probability of high-abundant events and low-abundant events has large discrepancies. The probability π increases sharply near $p = 0$ but slowly near $p = 1$, which makes it ideal when the probability of an event is very small. We observed that in our microbiome data, this is usually the case.

We first evaluate whether our model can make predictions at the last time-point, *i.e.*, one-step ahead forecasting, that correlate better with the ground truth measurements, compared to a *naive* approach in which we make predictions by assuming that observed OTUs at time T and $T-1$ do not change at all, *i.e.*, $\mathbf{x}_{nT} = \mathbf{x}_{nT-1}$, which is known to be a good assumption in some cases. Table 5.2 shows Spearman correlations indicating that for 5 out of 6 time-series, modeling the dynamics of OTU changes over time has actual predictive value. SGMGT-embedded sampler achieves around 2 times faster convergence to a stable

testing predictive comparing to exact MCMC. We see these results on forecasting of OTU concentrations as interesting preliminary results that need to be further investigated.

Figure 5.3 show five selected topics from the model learned for subject S2, respectively. Intensities, $\boldsymbol{\theta}_{nt}$, and OTU weights, ϕ_{mk}, for selected topics (T14, T38, T42, T47 and T49), are shown in Top and Bottom panels, respectively. The bottom pannel of the figure shows weights for 5 different topics (colors), where each bar represent an element of ψ_k, a column of $\boldsymbol{\Psi}$. The proportion of non-zero intensities is 64%. We see that latent variable intensities, $\boldsymbol{\theta}_{nt}$, are nicely localized in time. We verified that Topic 49 is consistent with the onset of a *Salmonella* infection suffered by the subject (see (David et al., 2014)), while Topic 38 is related to its recovery period. Interestingly we see that Topic 14, stably present up to the time of infection does not reappear even after recovery has taken place. Also interesting is that the five selected topics in Figure 5.3 only account for about 10% of the total OTUs in the sample ($\phi_{mk} > 10^{-4}$), which indicates that topics are not only localized in time but in OTU space. In particular, Topic 49 is enriched for *Proteobacteria*, and Topics 14 and 38 are enriched for *Firmicutes*, while Topic 42 is enriched for *Tenericutes*. All these results are consistent with the findings of (David et al., 2014), which were derived using a completely different, more biologically targeted approach.

5.7 Summary comments

We have introduced a dynamic time-series model based on Poisson factor analysis. The model allows for count and binary data, as well as for nonuniformly sampled time-series. Efficient inference methods are developed, that either scales with the number of non-zeros in the data and binary latent variables, or scale with the size of small subset of data. Extensive results on benchmark data demonstrate the excellent performance of our simple yet elegant specification. Results on real microbiome data highlight the applicability of our model to interesting problems in modern computational biology. Finally, stochastic gradient MCMC

embedded method achieve relatively comparable results with full-batch gibbs sample, while the computational time is dramatically reduced.

As future work, we would like to explore the possibility of specifying a deep version of our model, building upon ideas of deep PFA models (Henao et al., 2015a) for extended flexibility and representation ability. We are also considering working on a model specification where multiple observations can occur at the same point in time, but share a common transition *backbone* model. This could be very useful for the analysis of electronic medical records data. In modeling perspective, how to derive a stochastic gradient MCMC that can sample high dimensional discrete distribution would be a challenging but intriguing future work.

6

Supervised neural topic Bayesian analysis for gene expression data

6.1 Introduction

Our ability to identify the etiology of febrile illness is limited by nonspecific clinical presentations, low sensitivity and prolonged time to positivity of current pathogen specific methods, and in some cases difficulty in determining whether presence of pathogen alone indicates invasive disease. Consequently, there is a compelling need for novel diagnostic approaches to accurately discriminate between viral and bacterial etiology versus non-infectious causes of febrile illness. The host response to pathogens uses pattern recognition receptors tied to specific transcriptional responses that can be measured at the mRNA and protein levels. We hypothesize that we can harness this response to generate robust classifiers and high-fidelity assays to distinguish bacterial and viral illnesses that are inclusive of high consequence pathogens of relevance to the modern warfighter.

The objective is to use our existing biorepository of exquisitely phenotyped specimens from our global collaborations to refine our existing bacterial and viral classifiers and expand their capabilities to include a number of high consequence pathogens. We hope

to use RNAseq and proteomic methods to develop and validate host-based assays specific for pathogen class. We have sufficient samples in our current repository to allow for both refinement of existing bacterial and viral RNA classifiers and independent validation. The host response RNA and protein classifiers will be transferred to more clinically feasible platforms (e.g., RT-PCR for RNA and Multiple Reaction Monitoring (MRM) for proteins); thus, providing platforms for new, rapid, high fidelity clinical diagnostics for global pathogens.

Deep learning has advanced rapidly in recent decade and demonstrates state-of-the-art performance in various fields (LeCun et al., 2015), including bioinformatics(Park and Kellis, 2015; Angermueller et al., 2016). Consequently, application of deep learning in biomedical data to gain insight has been one recent hotspot in both academia and industry. Nevertheless, despite the great success in performance for various tasks, there remains many potential challenges, including imbalanced or limited data, model interpretability, and architecture and hyperparameters sensitivity. Thereby, to fully exploit the capabilities of deep learning, how to prudently develop tailored strategies regarding theses issues discussed herein would be key to the success of future deep learning approaches in bioinformatics.

In this chapter we focused on developing deep learning strategy which not only demonstrate great success in achieve accurate prediction, but also to shed light upon the regulatory network mechanism, which is of prime interest. Specifically, we aim at identifying interpretable "topics" (the functionality groups of the essayed gene) that represents a small collection of transcripts that works in a synergistic manner and contributes importantly to the classifier. Analyzing the composition of these "topics" could be an important step towards understanding the regulatory mechanism of different types of infection. Furthermore, we leverage SGMCMC to perform approximate Bayesian inference. Bayesian methods has been reported to have advantages and yield excellent and robust performance in the scenarios where data is limited and/or latent factor model is involved (Hoff, 2009), thus could be important in our application.

We highlight that our framework differs from traditional topic modeling strategy, such

as LDA, in several aspects.

- We consider a *non-linear* topic decomposition approach, each of the topic represents a complicated non-linear composition of vocabulary.
- The topics are explicitly selected according to *supervised* signals, which characterize key features facilitating downstream supervised tasks.
- A two-way Dirichlet prior is introduced for the topic loading weight matrix, to induce sparsity, non-negativity and non-overlapping properties.
- Instead of using Gibbs sampling or variational inference method, we consider the SGMCMC method proposed in chapter 4 for inference.

6.2 Supervised neural topic model

Let $d_i \in \mathbf{R}^V$ denote the i-th document and y_i denote the document label. Given a finite topic number K, the i-th document d_i is first abstracted into a topic vector $h_i \in \mathbf{R}^K$. The generative process can be described as following. Each topic h_{ik} is achieved by a weighted summation (with offset) of original data. The weight random vector W_k, lives on the simplex, is drawn from a sparsity-inducing prior distribution (we use Dirichlet distribution). Followed by this linear interpolation, a non-linear saturation function (we use hyperbolic tangent function) is applied to account for non-linearity relationship between word abundance and topic intensity. After this, the probability of label y_i is formulated as the softmax output of applying another neural perceptron layer to all the latent topic units. In specific, the model can be written as

$$h_i = \tanh(Wd_i + b) \ , \ p(y_i) = \text{softmax}(Uh_i + c), \tag{6.1}$$

$$W \sim \text{Dirichlet}(\cdot) \ , \ U, b, c \sim \mathcal{N}(\cdot) \tag{6.2}$$

The motivation for formulating W as Dirichlet is two-fold. First, such a specification would naturally induce sparsity over the topic-word composition distribution by leveraging a

small concentration parameter of Dirichlet. Second, this specification constrain the weights to be a probabilistic density, which gives direct interpretation of the weight value as the percentage contribution of the corresponding word to the topic intensity. Note that in the original data each word is first normalized to have unit L-2 norm over data samples, in order to eliminate the effect of the scale of word abundances to the absolute values of the weights.

6.2.1 Two-way Dirichlet prior

In many applications, encouraging non-redundant topics that can cover distinct word subsets, rather than allowing them to overlap could be helpful to obtain a clear interpretation. Motivated by this, we further consider a two-way Dirichlet prior for the weight matrix W, to encourage both row-wise and column-wise sparse structure. This can be achieved by assigning Dirichlet distribution both row-wisely and column-wisely and re-normalizing the distribution. i.e.,

$$p(W_{mn}) \propto A_{mn} B_{mn} \ , \ A_m \sim \text{Dirichlet}(\alpha) \ , \ B_n \sim \text{Dirichlet}(\beta), \tag{6.3}$$

where A_m is a row vector and B_n is a column vector. To constrain all A_i and B_j live in the simplex, we leverage an additional softmax layer, and directly consider the unnormalized $\tilde{A_m}$ and $\tilde{B_n}$

$$A_m = \text{softmax}(\tilde{A_m}) \ , \ B_n = \text{softmax}(\tilde{B_n}) \tag{6.4}$$

We use each gene as independent feature, the raw transcripts abundance are collapsed into each gene by summation. We denote this model as *gene-level model*. The model is illustrated as in figure 6.1

6.2.2 Transcript-level composition inference

Insofar we formulate our method as identifying topic in document. In the context of analysing transcript data, we consider the contribution of each transcript in a finer hierarchy

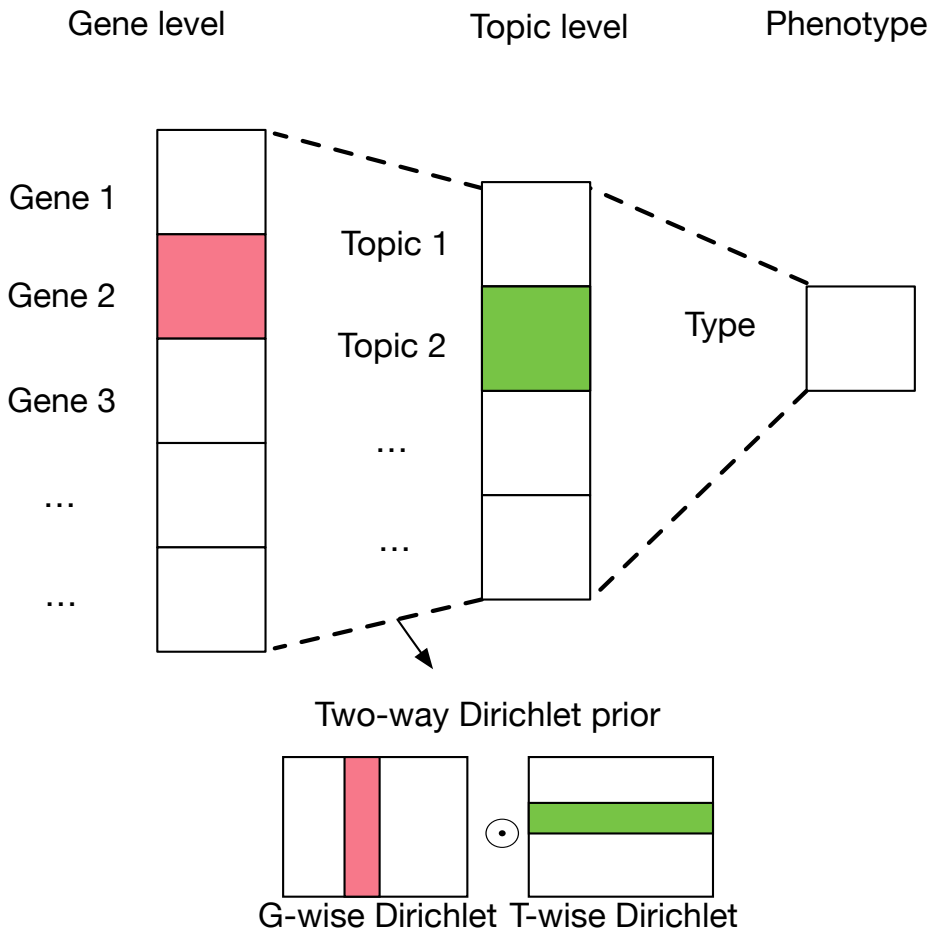


FIGURE 6.1: Neural topic model with two-way Dirichlet

. Specifically, we model the impact of single transcript as first through the gene that encoded this transcript, then affect the model prediction through gene. Thus the hierarchical structure from the top to the bottom would be *gene topics/groups* → *genes* → *transcripts*. From modeling perspective, we add another transcript-gene layer which resembles the gene-topic layer:

$$g_i = \sigma(Vd_i + a), h_i = \sigma(Wg_i + b),$$
$$p(y_i) = \text{softmax}(Uh_i + c) \tag{6.5}$$

where W, b, U, c follows the setup described in (6.5)-(6.4). a is specified as Gaussian distributed. V is specified to have a *masked* two-way Dirichlet distribution:

$$p(V_{mn}) \propto C_{mn} D_{mn} M_{mn} \ , \ C_m \sim \text{Dirichlet}(\gamma) \ , \ D_n \sim \text{Dirichlet}(\eta) \tag{6.6}$$

The mask M indicates whether n-th transcript is decoded from m-th gene. Similar to gene-level model, C_m is a row vector and D_n is a column vector. This model, denoted as *transcript(Isoform)-level model*, is illustrated as in Figure 6.2. Conversely, we note that in the we use eta greater than 1 to discourage sparsity for gene-transcript composition. The motivation for this is due to the fact that we are more interested to learn a weighted summation representation of all different corresponding isoforms, rather than selecting a single represent from these isoforms.

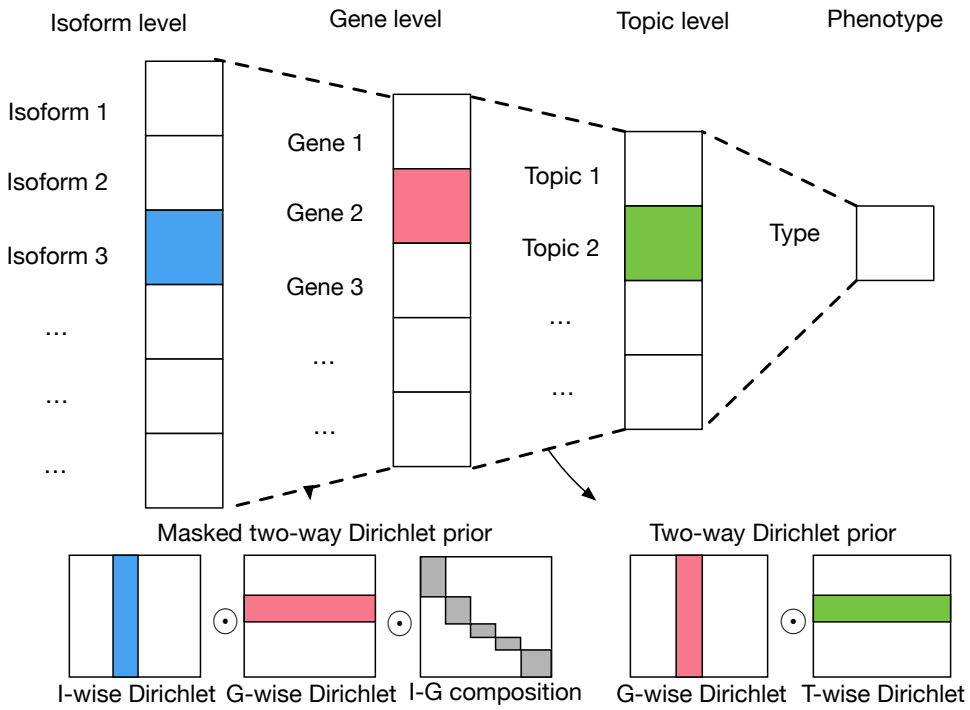


FIGURE 6.2: Transcript level model

6.2.3 *Model learning*

Due to the lack of conjugacy, it would be difficult to directly apply Gibbs sampling. Instead we resort to SGMGT for scalable approximate posterior modeling inference. The SGMGT can be readily applied with the gradient of log-likelihood. Omitting constant the log-likelihood objective can be written as:

$$\text{Gene-level model:} \quad \mathcal{L} = \text{NLL} + (1-\alpha)\sum_{ij}\log A_{ij} + (1-\beta)\sum_{ij}\log B_{ij} \tag{6.7}$$

$$\begin{aligned}\text{Transcript-level model:} \quad \mathcal{L} = \text{NLL} &+ (1-\alpha)\sum_{ij}\log A_{ij} + (1-\beta)\sum_{ij}\log B_{ij} \\ &+ (1-\gamma)\sum_{ij}\log C_{ij} + (1-\eta)\sum_{ij}\log D_{ij}\end{aligned} \tag{6.8}$$

6.2.4 *Experiment*

We evaluated our method on duke clinical data to discriminate between viral and bacterial etiology versus non-infectious causes of febrile illness. The label y_i contains 3 pathogen classes (bacterial, viral, non-infectious). In total 55,688 transcripts and 21,203 genes is measured, and 212 subjects are involved. The inference is performed for 5000 burn-in rounds and 100 posterior samples are collected for testing. Testing accuracy is performed via a 10-fold cross validation where within each fold 10 random initializations are performed. For baseline we consider group lasso method (Friedman et al., 2010). The results are summarized in table 6.2.4. Surprisingly, we empirically found that by use only 3 topic units, our model achieved high testing accuracy. We set the α,β and γ to be 0.01, the η is set to be 2.

Model	Group Lasso	Ours (Two-way Dir)
Gene-level model	0.187$\pm$0.050	**0.162$\pm$0.083**
Transcript-level model	0.177 $\pm$ 0.066	**0.149$\pm$0.075**

Table 6.1: Error rate on testset with 10 fold cross-validation

The traceplot of 3 weight parameters are shown in figure 6.3. We empirically observed

that the parameters demonstrate exploration ability even when the validation performance has converged comparing to directly optimizing (6.7) using Adam (Kingma and Ba, 2014). We further examine the non overlapping and sparsity property of our model. The topic-gene

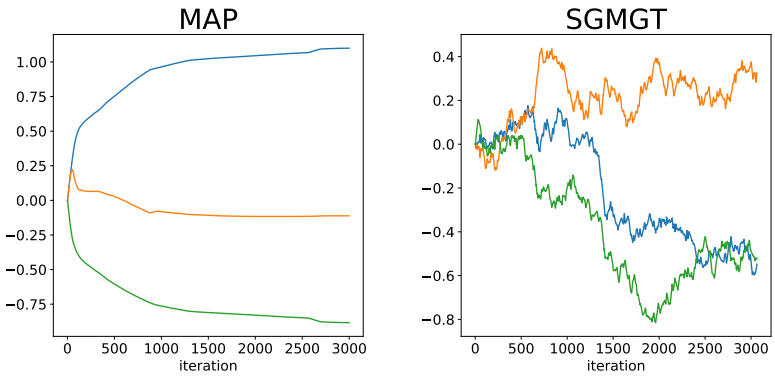


FIGURE 6.3: Traceplot of baseline method and our method.

composition matrix of our model are demonstrated in figure 6.4. The parameter value of two-way Dirichlet model is shown via posterior mean. It can be observed that only a small collection of gene is included in each topic, and the topics seem containing disjoint gene sets. The visualization of topic-gene composition is shown in the upper panel of figure 6.4 and figure 6.5, where in figure 6.4 lighter color indicate higher intensity. Each column represents one topic and each row represents one gene. Only top 10000 genes are shown. It can be observed that in group lasso, many genes are shared by two or more topics, whereas the two-way Dirichlet prior induces a strong separation among topics. This separation is not necessarily advantageous in prediction, however it gives a clear interpretation each topic for further functionality analysis. Interestingly, the topics identified by our model, illustrated in figure 6.5, is verified by domain experts as biologically meaningful. Each of the topic roughly represents a key functional group that is involved in each type of infection (figure 6.6). Additionally, as shown in figure 6.5, the results indicates that the contribution of gene can come from different transcripts either relatively evenly or rather exclusively. The lower panel of figure 6.4 shows the sparsity of each model. It can be seen that both group lasso and two-way Dirichlet render high-level of sparsity for inference in this noisy

data. In group lasso the parameters could be negative, whereas in two-way Dirichlet the parameters can only range from 0 to 1.

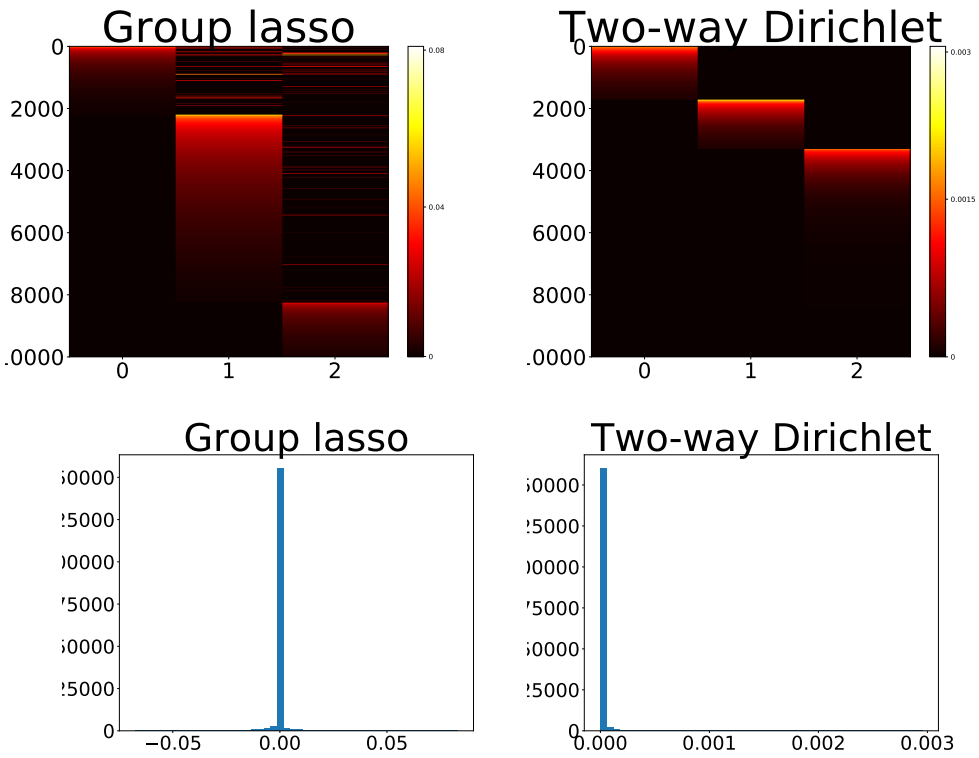


FIGURE 6.4: Topic-gene composition and sparsity

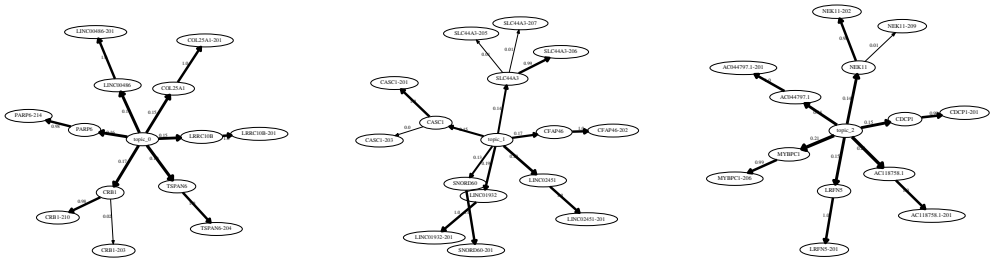


FIGURE 6.5: Identified topics from our model.

6.3 Future study

Learning the gene expression data is not trivial and may requires several rounds of iterations between experimental intervention and computational analysis. As a future research,

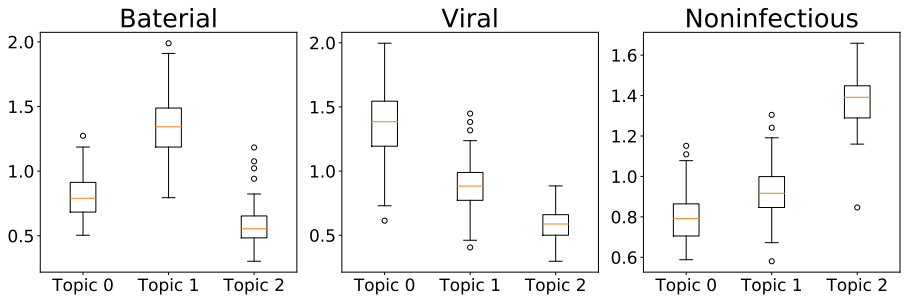


FIGURE 6.6: Topic intensity level for each group of infection type

we will assess the performance of our existing gene expression classifiers of viral and bacterial infection when applied to subjects infected with pathogens of global importance and warfighter relevance. We will then refine these classifiers to be broadly applicable to these pathogens in order to deliver a gene expression-based biomarker panel capable of accurately diagnosing and discriminating between pathogen classes. We will then transition the classifier to a clinically tractable RT-PCR platform and evaluate the assays performance in existing phenotyped patients with samples in our biorepository as we have done with other classifiers. Lastly, we will migrate the RT-PCR assay developed with human targets to non-human primate orthologs for validation in relevant studies at USAMRIID.

7

Conclusions

This dissertation presented the development of efficient and scalable MCMC inference for Bayesian modeling. The first part of the dissertation examines the connection between HMC and slice sampling, and develops a novel HMC method beyond the standard HMC in stationary mixing efficiency, albeit additional numerical difficulties apply with approximate integration of the underlying Hamiltonian contour that characterized by the density function. Such a method provides a generic replacement for many scenarios where HMC applies.

The second part of the dissertation discusses about scalable MCMC inference as an extension of the first part. The motivation for this research arises from the increasing trend and interests of large datasets analysis. Comparing to optimization method which gives a single point estimation of the target posterior, the proposed MCMC method enjoys many advantages of Bayesian approaches, such as better exploration of parameter space rather than falling in local minimum solutions. Additionally, the use of additional Langevin dynamics and scheduled resampling procedure alleviate many practical issues that possessed by the method proposed in the first part. Extensive simulation and real-word problems demonstrate outstanding performance of the proposed method comparing to previous stochastic gradient methods (SGLD, SGHMC, SGNHT) and popular optimization method

such as ADAM and RMSprop. Notably the computation speed is at a constant factor (less than 5) comparing with SGD.

The remaining parts provide two scenarios for representing typical computational problems in biomedical applications. Bayesian inference is increasingly popular in biomedical analysis with latent factors in that it presents an elegant and simply way for characterizing uncertainty level of estimation and requires less efforts for computing the complicated integrals. Besides, biomedical datasets is often data-demanding while the feature dimension is dramatically larger than the data sample size, where Bayesian approaches seem to be a natural recipe for these scenario. However standard MCMC is inherently slow and cannot scale. Toward this end we considers the methods we consider the stochatstic gradient MCMC method we proposed in last part. The first scenario concerns understanding the underlying trend represented by latent statistical process with spatial-temporal intensity profiles. We embed the SGMCMC procedure in conditional posterior sampling within block gibbs sampling. Comparing with the original augmented Gibbs sampling approach, the predictive performance does not decrease much however the computational time is dramatically reduced.

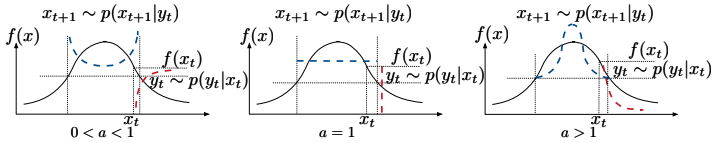
For the second scenario, we consider a typical supervised task for infection type detection. This is one key step towards many prevalent problems in personalized and precise medication, thus is of great interest. Models employ complicated multi-layer neural networks received great success however lack of interpretation with the underlying mechanism, which limits their advantages. We consider a Bayesian formalism employing task-specific sparsity-inducing prior. Due to the lack of conjugacy, the inference is performed via the SGMCMC method we propose in previous. Our empirical analysis demonstrate not only outstanding predicative performance empowered by Bayesian model averaging, but also identification of topics that corresponding to each infection types which can be verified by domain experts. This research shed light upon the interpretable deep Bayesian learning approaches.

As of future study, we note that our use of heavy-tailed kinetic distribution is inspired by the exponential family class of model introduced by (Roberts and Tweedie, 1996), which have the potential energy with form $U(x) = x^{\omega}$. The major reasons we use this setup are that 1) this family of distribution (in our notation, monomial Gamma distributions) is easy to directly sampling from 2) theoretical analysis is convenient. Nevertheless the optimality may not necessarily be reached with such a specification. We note that several theoretical analysis in chapter 3 is established on univariate cases. Extending these theories into higher dimension is non-trivial and would be interesting and important. Extensions such as leveraging geometric information or using higher order numerical integration could be helpful for mitigating numerical issues, however they are at a cost of additional computation. Probably approximate approaches that trade off between computation and numerical issue would be promising. The scalable inference method in chapter 4 is of great potential since it requires comparable computational resources as optimization method which is prevalent in current machine learning research, while demonstrates ability to explore and traverse over multimodal probability landscape and complex surface (which represents the advantage of Bayesian inference). Besides, it's flexible to use. Probably it could bring additional vitality of Bayesian learning into the new era of data-intensive machine learning.

Appendix A

Additional Theories and results for chapter 3 (MGS)

A.1 Illustration of MG-SS with different monomial parameters a



a	HMC space	Canonical (SS) space
0.5	***Standard HMC***	MG-SS
1	MG-HMC	***Standard SS***
Otherwise (a>0)	MG-HMC	MG-SS

FIGURE A.1: MG-HMC and equivalent MG SS.

The MG-SS for $0 < a < 1, a = 1$ and $a > 1$ are illustrated in Figure A.1. Red and blue dashed lines denote the conditionals $p(y_t|x_t)$ and $p(x_{t+1}|y_t)$, respectively. When $0 < a < 1$, the conditional distribution $p(y_t|x_t)$ is skewed towards the current unnormalized density value $f(x_t)$. The conditional draw of $p(x_{t+1}|y_t)$ encourages taking samples with smaller density value (small moves), within the domain of the slice interval $\mathbb{X}$. On the other hand, when $a > 1$, draws of y_t tend to take smaller values, while draws of x_{t+1} encourage sampling from those with large density function values (large moves). Intuitively, setting a to be small makes the auxiliary variable, y_t, stay close to $f(x_t)$, thus $f(x_{t+1})$ is close to $f(x_t)$. As a result, a larger a seems more desirable.

A.2 Monomial Gamma distribution

Several useful observations can be drawn from monomial Gamma distribution:

1) The mean and variance for it is 0 and $\frac{\Gamma(3a+1)}{3\Gamma(a+1)}m^{2a}$, respectively.

2) Scaling. For any $\lambda > 0$,

$$Y \sim \text{MG}(a, m) \quad \Rightarrow \quad \lambda Y \sim \text{MG}(a, |\lambda|^{1/a} m)$$

3) As $a \to \infty$, the distribution, regardless of scale, becomes more heavy-tailed.

A.3 Periodicity of Hamiltonian flow and higher dimensional HMC equivalents

First note that $\lim_{x\to\pm\infty} U(x) \to \infty$, since the integral $\int \exp(-U(x))$ is finite. From definition $\lim_{p\to\pm\infty} K(p) \to \infty$. Given above conditions, if the target distribution has one dimension, the Hamiltonian flow is periodic, and the Hamiltonian contour is closed (Ekeland and Lasry, 1980).

In (3.9), $\int_{x_{min}}^{x(\tau)} f(z)^{a-1}dz \in [0, \int_{\mathbb{X}}[H - U(z)]^{a-1}dz]$. For one dimensional problems, the Hamiltonian dynamics described in (3.9) has a period $T \triangleq 2a\int_{\mathbb{X}}[H - U(z)]^{a-1}dz$. Since the $K(p)$ has symmetric form, the contour is symmetric along $p = 0$. In the second half of the period, the particle x simply reverse the motion of the first half period.

However, if the dimensionality D is higher than one, the periodicity assumption will almost never be true, the flow will typically be quasi-periodic as the periods of each 1D component would not exactly match. In those cases, the hamiltonian trajectory is a one-dimensional manifold in high-dimensional space. If uniformly sample a time τ from an interval with width much larger than $\prod_d T_d$, where T_d is the period for d-th dimension, the hamiltonian trajectory will behave like a *dynamic billiard* (Goldstein, 1965). With infinite evolutionary time, the trajectory will almost certainly cover each point in a hyper-rectangle

[1] $\mathbb{Y} = \{x : L_d \leqslant x^{(d)} \leqslant R_d, \quad \text{for all} \quad , d \in \{1, \cdots, D\}\}$, which is one of the maximum hyper-rectangles that lives within the slice interval $\mathbb{X} = \{x : U(x) \leqslant H\}$. For each dimension d, the boundary of the hyper-rectangle L_d and R_d are determined by the last sample point $\mathbf{x}_{t-1}$.

The Hamiltonian trajectory and corresponding slice interval are shown in Figure A.2. As

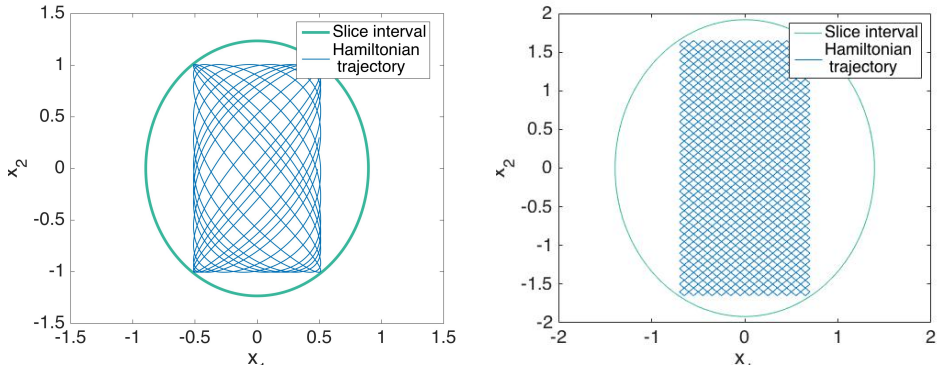


FIGURE A.2: Hamiltonian trajectory and corresponding slice interval when $a = 0.5$ (left) and $a = 1$ (right).

in univariate cases, when $a = 0.5$, the Hamiltonian dynamic corresponds to a conditional density with less probability mass in the region with large $f(x)$. When $a = 1$, the Hamiltonian dynamic corresponds to a uniform density. However, in each of the case the density is constraint in the hyper-rectangle $\mathbb{Y}$. Thereby, in cases more than one dimension, even the MG-HMC with $a = 1$ is not exactly recovering standard slice sampling, but rather a *generalized slice sampler*. For simplicity, suppose the mass matrix is $m\mathbf{I}$, it can be shown that $K(p) \sim \text{Gamma}(D/a, m)$, thus, this generalized slice sampler has iterative procedure (A.2) as below (Figure A.3). Red and blue dashed lines denote the conditionals $p(y_t|x_t)$ and $p(x_{t+1}|y_t)$, respectively. $x^{(1)}$ and $x^{(2)}$ denote the first and second dimension of target distribution.

[1] suppose the $U(x)$ is decomposable over dimensions, if not the hyper-rectangle will become hyper-diamond in the high-dimensional space.

$$p(y_t|x_t) \propto [\log f(x_t) - \log y_t]^{D/a-1}, s.t. 0 < y_t < f(x_t) \tag{A.1}$$

$$p(x_{t+1}|y_t) \propto [\log f(x_{t+1}) - \log y_t]^{a-1}, s.t. x_{t+1} \in \mathbb{Y} \tag{A.2}$$

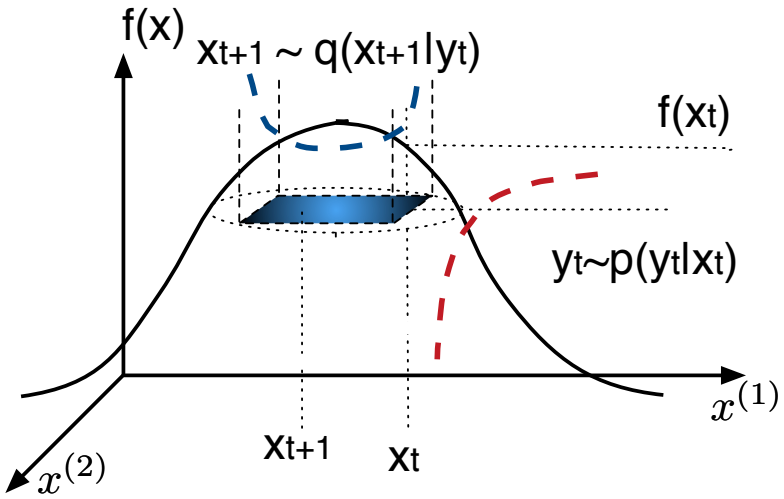


FIGURE A.3: 2D equivalent generalized slice sampler for MG-HMC $a = 1/2$.

A.4 Connecting HMC with generalized kinetics and slice sampling

We show in 3.2 and 3.3 that the generalized kinetic form $K(p) = |p|^{1/2}, a > 0$ lead to MG-SS 2. In fact, for the generalized kinetic form with mass parameter and monomial parameters, $K(p) = |p|^{1/a}/m, a, m > 0$, the conclusion still holds. To see this, one may rederive equations (3.5)-(3.12) using this generalized kinetic energy.

The (3.5) becomes

$$U(x) + |\frac{dW(x)}{dx}|^{1/a}/m - p' = 0\,. \tag{A.3}$$

Solving (A.3) for $W(x)$ gives

$$W(x) = \int_{x_{min}}^{x(\tau)} [mf(z)]^a dz + C\,, \qquad f(z) = \begin{cases} H(\cdot) - U(z), & z \in \mathbb{X} \\ 0, & z \notin \mathbb{X} \end{cases}, \tag{A.4}$$

Hence, from (3.4) we have

$$x' = m^a a \int_{x_{min}}^{x(\tau)} f(z)^{a-1} dz - \tau \,, \tag{A.5}$$

For the (A.5), Hamiltonian dynamics with generalized kinetics $K(p) = |p|^{1/a}/m, a, m > 0$ has period $2m^a a \int_{\mathbb{X}} [H(\cdot) - U(z)]^{1-a} dz$ and is symmetric along $p = 0$ (due to the symmetric form of the kinetic function). The system time, τ, is uniformly sampled from a half-period of the Hamiltonian dynamics.

$$\tau \sim \text{Uniform}\left(-x', -x' + m^a a \int_{\mathbb{X}} [H(\cdot) - U(z)]^{a-1}\right)$$

.

The constant $m^a a$ does not matter because when we transform the problem using inverse CDF methods, this constant would diminish from the formulation due to the normalization. From the inverse CDF transform sampling method, uniformly sampling τ from half of a period and solving for x^* from (3.9), are equivalent to directly sampling x^* from the following density

$$p(x^*|H(\cdot)) \propto [H(\cdot) - U(x^*)]^{a-1} \,, \qquad \text{s.t.,}\ H(\cdot) - U(x^*) \geqslant 0 \,. \tag{A.6}$$

Denote $y_t = e^{-H_t(\cdot)}$, by substituting $H_t(\cdot)$ with y_t in (3.10), the conditional updates for this new sampling procedure can be rewritten as below, yielding the MG-SS with arbitrary momomial parameter $a > 0$, with conditional distributions defined as

$$\text{Sampling a slice:} \quad p(y_t|x_t) = \frac{1}{\Gamma(a) f(x_t)} [\log f(x_t) - \log y_t]^{1-a} \,, \quad \text{s.t. } 0 < y_t < f(x_t) \,, \tag{A.7}$$

$$\text{Conditional sampling:} \quad p(x_{t+1}|y_t) = \frac{1}{Z_2(y_t)} [\log f(x_{t+1}) - \log y_t]^{1-a} \,, \quad \text{s.t. } f(x_t) > y_t \,, \tag{A.8}$$

Note that the mass parameter m in generalized kinetic function will not influence the density (A.7) and (A.8)

A.5 Theoretical properties of MG sampler

A.5.1 Convergence properties of MG-SS

Following (Tierney and Mira, 1999) and (Robert and Casella, 2004), we show in below that the MG-SS is reversible and *Harris ergodic*. As a result, the chain is guaranteed to uniquely and asymptotically converge to the target distribution. Next, following standard slice sampler (Tierney and Mira, 1999), we show that MG-SS is uniformly ergodic under the *Doeblin's conditions* (Isaac, 1963) in Lemma 5[2].

Lemma 5. *(Uniform ergodicity) Suppose $f(\cdot)$ is bounded and has bounded support. If $a \geqslant 1$, the analytic MG-SS is uniformly ergodic.*

Geometric ergodicity is a less restrictive property compared to the uniformly ergodicity. In MG-SS, we hypothesize this requires $yZ_2'(y)$ to be non-increasing. Formal verification of these conditions is beyond the scope of this paper, thus is left as intersting future work.

Invariance

Theorem 6. *(π-invariant) The Hamiltonian dynamics $S : (x, p) \to (x', p')$ with parameter a, is π-invariant.*

Proof. First, the total Hamiltonian H is perserved with Hamiltonian dynamics.

$$\frac{dH}{dt} = \frac{\delta H}{\delta p}\frac{dp}{dt} + \frac{\delta H}{\delta x}\frac{dx}{dt} = 0$$

Thus, for any $f(x, p)$

$$\mathbb{E}_{\pi(x,p)} f(x,p) = \int f(x,p)\frac{e^{-H(x,p)}}{Z_1}dxdp = \int f(x,p)\frac{e^{-H(S(x,p))}}{Z_1}|J_s|dxdp$$

By Liouville's theorem, $|J_s| = 1$. Therefore, $\mathbb{E}_{\pi(x,p)} f(x,p) = \mathbb{E}_{\pi(x,p)} f(S(x,p))$, the transformation S is π-invariant. □

[2] We hypothesize that this proposition holds for $0 < a < 1$, however we leave it for further investigation.

Reversibility

Proposition 7. *(Reversibility, detailed balance) For the transition kernel in eqn.* (A.24). *We have* $\mathbb{E}_{\kappa_h^{-1}(x'|x)}x' = \mathbb{E}_{\kappa_h(x'|x)}x'$.

Proof. From the symmetric form of the joint distribution, we have

$$\kappa_1(x'|x)p(x) = \kappa_1^{-1}(x|x')p(x').$$

Using induction we have $\kappa_h(x'|x)p(x) = \kappa_h^{-1}(x|x')p(x')$. Thus, $\mathbb{E}_{\kappa_h^{-1}(x'|x)}x' = \mathbb{E}_{\kappa_h(x'|x)}x'$ □

Harris ergodicity

Theorem 8. *(Harris ergodicity) The MG sampler with parameter* a*, is Harris ergodic with invariant distribution* $p(x)$*.* $\kappa_h(\cdot, x)$ *is the* h*-th transition kernel.*

$$\|\kappa_h(\cdot, x) - p(x)\|_{TV} \to 0, \text{ as } h \to \infty$$

Further, $\|\kappa_h(\cdot, x) - p(x)\|$ *is monotonically nonincreasing in h. (Meyn and Tweedie (1993), proposition 13.3.2)*

Proof. Following Lemma 1 of Tan and Hobert (2008), it can be shown that MG sample is reversible, aperiodic and π-irreducible. The Harris recurrent property follows directly from Corollary 1 of Tierney (1994), which states that an π-irreducible Markov chain is Harris recurrent if for some h, $\kappa_h x, \cdot$ is absolutely continuous *w.r.t.* $p(x)$ for all $x \in \mathcal{X}$. □

Note that for MG-HMC, the Harris ergodicity cannot be directly extended from above conclusion, because 1) MG-HMC has fixed leap-frog step, 2) the Hamiltonian dynamics would not allow moving between contours with same energy. However, (Cances et al., 2007) showed that HMC is π-irreducible under the assumption that the potential energy has an upper bound. One can use the similar technique to show such conclusion holds for MG-HMC.

Geometric ergodicity

Establishing such ergodicity for general cases requires demonstrating the *drift* and *minorisation* conditions (Johnson, 2009). (Roberts and Rosenthal, 1998) has showed that for any univariate log-concave density $f(\cdot)$, the resulting Markov chain associated with the slice sampler is geometrically ergodic, and the quantitative convergence bounds are available. In fact, a necessary condition for any multivariate density being geometrically ergodic is that $y\mu'(y)$ is non-increasing with respect to y, where $\mu(y)$ is the Lebesgue measure of the slice interval $\{w : f(w) \geqslant y\}$, and the prime symbol denotes derivative *w.r.t.* y. In MG-SS, we hypothesize this requires $yZ_2'(y)$ to be non-increasing. i.e., the geometric ergodicity requires

$$y\frac{d}{dy}\int_{f(x)>y}[\log f(x) - \log y]^{a-1}$$

to be non-increasing with y. For $E(x) = x^{\omega}, \omega > 0$, such condition holds. However we leave the formal verification for future work.

Uniform ergodicity

Proposition 9. *(Uniformly ergodic) If $f(\cdot)$ is bounded and have bounded support, the analytic MG-SS with $a \geqslant 1$ is uniformly ergodic,* i.e. *,*

$$\lim_{h\to\infty}\sup_{x_0\in\mathcal{X}}\|\kappa_h(x_0, x) - p(x)\|_{TV} = 0.$$

Proof. Following (Tierney and Mira, 1999) and (Robert and Casella, 2004), without lossing generality, assume $f(\cdot)\in[0, 1]$ and the support for $f(\cdot)$ is [0,1]. A sufficient and necessary condition to demonstrate uniform ergodicity, is by *Doeblin's condition* (Isaac, 1963). The c.d.f. of transition kernel is given by,

$$\xi(v) \quad = \quad \Pr(f(x_{t+1}) > \eta \,|f(x_t) = v).$$

To establish Doeblin's condition, we will first show that $\xi(v)$ is decreasing in v for $\forall\eta$, when $a > 1$. With some algebra one can obtain,

$$\begin{aligned}\xi(v) &= \frac{1}{v\Gamma(a)}\int_0^{\eta\wedge v}\frac{Z_2(w)-Z_2(\eta)}{Z_2(w)}\cdot(\log v-\log w)^{a-1}dw \\ &= \frac{1}{v\Gamma(a)}\int_0^{v}\max\{K(w;\eta,v),0\}dw,\end{aligned}$$

Where $K(w;\eta,v) \triangleq \left(1-\frac{Z_2(\eta)}{Z_2(w)}\right)\cdot(\log v-\log w)^{a-1}$. $Z_2(w) = \int_{f(x)>w}(\log f(x) - \log w)^{a-1}dx$. When $a > 1$, $Z_2(w)$ is decreasing in w. To see this, suppose $w_1 \geqslant w_2$,

$$\begin{aligned}Z_2(w_1) &\leqslant \int_{f(x)>w_2}(\log x-\log w_1)^{a-1}dx \\ &\leqslant \int_{f(x)>w_2}(\log x-\log w_2)^{a-1}dx = Z_2(w_2).\end{aligned}$$

Denote $\psi(w) = \left(1-\frac{Z_2(\eta)}{Z_2(w)}\right)$. $\psi(w)$ is decreasing in w, and $\psi(w) > 0$ when $w \in (0,\eta)$ When $v \geqslant \eta$, we have,

$$\xi(v) = \frac{1}{v\Gamma(a)}\int_0^{v}K(w;\eta,v)dw.$$

Taking derivatives gives,

$$\begin{aligned}\xi'(v) &= \frac{1}{v\Gamma(a)}\left[\frac{a-1}{v}\int_0^{\eta}\psi(w)\cdot(\log v-\log w)^{a-2}dw\right] \\ &\quad -\frac{1}{v^2\Gamma(a)}\int_0^{\eta}\psi(w)\cdot(\log v-\log w)^{a-1}dw \\ &\triangleq -\frac{1}{v^2\Gamma(a)}\int_0^{\eta}\psi(w)\cdot h(w)dw,\end{aligned}$$

where we denote $h(w) = (\log v - \log w - a + 2)(\log v - \log w)^{a-2}$, it can be validated that $\int h(w)dw = 0$. Meanwhile, one can also validate that $\exists w_0 \in (0,v)$, where $h(w) > 0$

for $\forall w \in (0, w_0)$ and $h(w) < 0$ for $\forall w \in (w_0, v)$. Therefore,

$$\begin{aligned}
\xi'(v) &= -\frac{1}{v^2\Gamma(a)} \int_0^{\eta} [\psi(w) - \psi(w_0)] h(w) dw \\
&= -\frac{1}{v^2\Gamma(a)} \int_0^{w_0} [\psi(w) - \psi(w_0)] h(w) dw \\
&\quad -\frac{1}{v^2\Gamma(a)} \int_{w_0}^{\eta} [\psi(w) - \psi(w_0)] h(w) dw < 0.
\end{aligned}$$

The inequality follows because $\psi(w)$ is increasing in w. Likewise, one can obtain that When $v \leqslant \eta$, $\xi(v)$ can be written as,

$$\xi(v) = \frac{1}{v\Gamma(a)} \int_0^{v} K(w; \eta, v) dw$$

Thereby, similar to the case of $v \geqslant \eta$, we have,

$$\begin{aligned}
\xi'(v) &= -\frac{1}{v^2\Gamma(a)} \int_0^{v} [\psi(w) - \psi(w_0)] h(w) dw \\
&= -\frac{1}{v^2\Gamma(a)} \int_0^{w_0} [\psi(w) - \psi(w_0)] h(w) dw \\
&\quad -\frac{1}{v^2\Gamma(a)} \int_{w_0}^{v} [\psi(w) - \psi(w_0)] h(w) dw < 0.
\end{aligned}$$

Thus, $\xi(v)$ is equally decreasing in v, where $v \in [0, f_0]$, $f_0 = \max(f(x))$. The upper and lower bound of $\xi(v)$ can be achieved by $\lim_{v \to 0} \xi(v)$ and $\lim_{v \to f_0} \xi(v)$, which are non-degenerate cdf for η. Thus one can establish uniform ergodicity via Doeblin's condition. The case $a = 1$ is standard slice sampling, and has been shown to be uniformly ergodic with bounded f and have bounded support (Tierney and Mira, 1999). □

A.5.2 Theoretical result about autocorrelation

Proof of $\rho_x(1) > 0$

We will first prove the Proposition 10, showing that $\rho_x(1) > 0$.

Proposition 10. *The one-step autocorrelation, $\rho_x(1) \triangleq \rho(x_t, x_{t+1})$, is non-negative.*

From (3.11) and (3.12), and provided the conditional density $p(x_t|y_t)$ and $p(x_{t+1}|y_t)$ have the same form, we have

$$\mathbb{E}x_t x_{t+1} = \mathbb{E}_{p(y_t)}\big[\mathbb{E}_{p(x_t|y_t)} x_t \mathbb{E}_{p(x_{t+1}|y_t)} x_{t+1}\big] = \mathbb{E}_{p(y_t)}\big[\mathbb{E}_{p(x_{t+1}|y_t)} x_{t+1}\big]^2 , \tag{A.9}$$

where $p(x|y)$ is the conditional distribution defined in (3.12). From (A.9), when $p(x)$ is *symmetric* at $x = c$ (c being a constant) , $\mathbb{E}_{p(x_{t+1}|y_t)} x_t = \mathbb{E}x$, which gives $\rho_x(1) = 0$. From (A.9) and the Jensen's inequality, assuming the sampler has reached stationary period, we can obtain

$$[\mathbb{E}x]^2 = \big[\mathbb{E}_{p(y)}\mathbb{E}_{p(x|y)} x\big]^2 \leqslant \mathbb{E}x_t x_{t+1} \leqslant \mathbb{E}_{p(y)}\big[\mathbb{E}_{p(x|y)} x^2\big] = \mathbb{E}x^2$$

This indicate $0 \leqslant \rho_x(1) \leqslant 1$

Autocorrelation for $\{y_t\}_{t=1,2,\cdots}$

The analytic MG-SS performs sampling in an iterative manner, *i.e.*, $x_t \to y_t \to x_{t+1} \to y_{t+1} \cdots$. To gain insights about the limiting behavior of $\{x_t\}_{t=1,\ldots}$, when a goes to infinity, we first consider the Markov Chain of $\{y_t\}_{t=1,\ldots}$, which can be analytically calculated regardless of the form of $U(x)$. Particularly, we will show that $\lim_{a\to\infty} \rho(y_t, y_{t+1}) = 0$. Also, that $\rho(y_t, y_{t+h})$ is a non-negative decreasing function of the time lag in discrete steps $h = 1, 2, \ldots$.

We start by finding the autocorrelation $\rho(y_t, y_{t+1})$. First consider compute $\rho(H_t, H_{t+1})$, where $H = -\log y$.

$$
\begin{aligned}
\mathbb{E}H_tH_{t+1} &= \mathbb{E}_{p(x_t)}\mathbb{E}_{p(H_t|x_t)}H_t\mathbb{E}_{p(H_{t+1}|x_t)}H_{t+1} \\
&= \mathbb{E}_{p(x_t)}\left[\frac{\Gamma(a)(a+U(x_t))e^{-U(x_t)}}{\Gamma(a)e^{-U(x_t)}}\right]^2 = \mathbb{E}_{p(x)}[a+U(x)]^2 \\
\mathbb{E}H &= \frac{1}{\Gamma(a)Z_1}\int HB(H)e^{-H}dH \\
&= \frac{1}{\Gamma(a)Z_1}\int \Gamma(a)[a+U(x_t)]e^{-U(x_t)}dx_t = \mathbb{E}_{p(x)}[a+U(x)]
\end{aligned}
$$

$$
\text{Var}(H) = \mathbb{E}H^2 - (\mathbb{E}H)^2 = \mathbb{E}_{p(x)}(a+[a+U(x)]^2) - (\mathbb{E}_{p(x)}[a+U(x)])^2
$$

$$
\rho(H_t, H_{t+1}) = \frac{\mathbb{E}H_tH_{t+1} - (\mathbb{E}H)^2}{\text{Var}(H)} = \frac{\text{Var}_{p(x)}U(x)}{a + \text{Var}_{p(x)}U(x)}
$$

Since the mapping from $y(H) = \exp(-H)$ is bijective, after some algebra we can obtain $\lim_{a\to\infty}\rho(y_t, y_{t+1}) = 0$. Similarly, for two-step autocorrelation one can obtain

$$
\rho(H_t, H_{t+2}) = \frac{\mathbb{E}H_tH_{t+2} - (\mathbb{E}H)^2}{\text{Var}(H)} = \frac{\mathbb{E}_{p(H)}[\mathbb{E}_{p(x|H)}U(x)]^2 - (\mathbb{E}_{p(x)}U(x))^2}{a + \text{Var}_{p(x)}U(x)}
$$

Thereby, one can obtain

$$
0 \leqslant \rho(H_t, H_{t+2}) \leqslant \rho(H_t, H_{t+1}) \tag{A.10}
$$

It not hard to further obtain that $\rho(H_t, H_{t+h})$ is a non-negative decreasing function of step h. After some algebra, the $\rho(y_t, y_{t+h})$ also has this monotonicity w.r.t. h. The property of y as part of the markov chain gives some intuition on the behavior of x.

Proof of Distillation theory

In order to describe the limiting behavior of $\rho(x_t, x_{t+1})$, We first establish following Lemma

Lemma 11. *Define $\mathcal{B}(0,\epsilon)$ to be the $d-$dimensional ball around zero with a radius of ϵ and p_0, k be some positive constants, then for sufficient large a and any $\epsilon = \sqrt{p_0 k}\frac{\log a}{\sqrt{a+1}}$, we have*

$$g(a,d,\epsilon) = \int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt$$

$$\geqslant \frac{d\pi^{d/2}}{\Gamma(d/2+1)} \frac{(d-2)!!p_0^{d/2}}{k(a+1)^{d/2}\log a}\left\{1 - \frac{(d+1)(k\log a)^{\frac{d+1}{2}}}{2a^{k/2}}\right\}.$$

Proof. We will prove the result first for $d = 1$ and $d = 2$ and then extends the result to $d \geqslant 3$ by mathematical induction.

For $d = 1$, it is hard to directly evaluate $g(a, 1, \epsilon)$, but we notice that

$$g(a,1,\epsilon) \geqslant \frac{2}{\epsilon}\int_0^\epsilon t\left(1 - \frac{t^2}{2p_0}\right)^a dt = \frac{2p_0}{\epsilon(a+1)} - \frac{2p_0}{\epsilon(a+1)}\left(1 - \frac{\epsilon^2}{2p_0}\right)^a.$$

Taking $\epsilon = \sqrt{p_0 k}\frac{\log(a+1)}{\sqrt{a+1}}$ for any $k > 0$, we have

$$g(a,1,\epsilon) \geqslant \frac{2p_0^{1/2}}{k(a+1)^{1/2}\log a}\left\{1 - \left(1 - \frac{\epsilon^2}{2p_0}\right)^a\right\} \geqslant \frac{\pi^{1/2}}{\Gamma(\frac{3}{2})}\frac{p_0^{1/2}}{k(a+1)^{1/2}\log a}\left\{1 - \frac{1}{a^{k/2}}\right\}.$$

For $d = 2$, we transform the integral to polar coordinate transformation and obtain that

$$g(a,2,\epsilon) = \pi\int_0^\epsilon r\left(1 - \frac{r^2}{2p_0}\right)^a dr = \frac{\pi p_0}{a+1} - \frac{\pi p_0}{a+1}\left(1 - \frac{\epsilon^2}{2p_0}\right)^a.$$

where r is an unknown positive constant. With the same ϵ, we have

$$g(a,2,\epsilon) \geqslant \frac{2\pi}{\Gamma(2)}\frac{p_0}{k(a+1)\log a}\left\{1 - \frac{1}{a^{k/2}}\right\}.$$

Consequently, we would generally guess that for any $d \geqslant 3$, we have

$$g(a,d,\epsilon) \geqslant \frac{d\pi^{d/2}}{\Gamma(d/2+1)}\frac{(d-2)!!p_0^{d/2}}{k(a+1)^{d/2}\log a}\left\{1 - \frac{(d+1)(k\log a)^{\frac{d+1}{2}}}{2a^{k/2}}\right\}. \tag{A.11}$$

We use mathematical induction to prove the this inequality. It has already been verified for $d = 1$ and $d = 2$. For $d \geqslant 3$, using polar coordinate, the general $g(a, d, \epsilon)$ can be written as

$$g(a,d,\epsilon) = \frac{d\pi^{d/2}}{\Gamma(d/2+1)}\int_0^\epsilon r^{d-1}\left(1 - \frac{r^2}{2p_0}\right)^a dr,$$

Using integration by parts, we have

$$\int_0^\epsilon r^{d-1}\left(1-\frac{r^2}{2p_0}\right)^a dr = \frac{1}{d}r^d\left(1-\frac{r^2}{2p_0}\right)^a\Big|_0^\epsilon + \frac{a}{dp_0}\int_0^\epsilon r^{d+1}\left(1-\frac{r^2}{2p_0}\right)^{a-1} dr$$

$$= \frac{\epsilon^d}{d}\left(1-\frac{\epsilon^2}{2p_0}\right)^a + \frac{a}{dp_0}\int_0^\epsilon r^{d+1}\left(1-\frac{r^2}{2p_0}\right)^{a-1} dr.$$

This implies that

$$g(a,d,\epsilon) = \frac{(d-2)p_0}{a+1}g(d-2,a+1,\epsilon) - \frac{p_0\epsilon^{d-2}}{a+1}\left(1-\frac{\epsilon^2}{2p_0}\right)^{a+1}.$$

Using the induction, we have

$$g(a,d,\epsilon) \geqslant \frac{d\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\frac{(d-2)!!p_0^{d/2}}{k(a+1)^{d/2}\log a}\left\{1-\frac{(d-1)(k\log(a+1))^{\frac{d-1}{2}}}{2(a+1)^{k/2}}\right\}$$

$$-\frac{d\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\frac{p_0\epsilon^{d-2}}{a+1}\left(1-\frac{\epsilon^2}{2p_0}\right)^{a+1}$$

Since $\epsilon = \sqrt{p_0 k}\frac{\log a}{\sqrt{a+1}}$, the second term can be upper bounded as

$$\frac{p_0\epsilon^{d-2}}{a+1}\left(1-\frac{\epsilon^2}{2p_0}\right)^{a+1} \leqslant \frac{p_0^{d/2}}{k(a+1)^{d/2}\log a}\cdot\frac{(k\log a)^{d/2}}{(a+1)^{k/2}}$$

Thus, we have

$$g(a,d,\epsilon)$$

$$\geqslant \frac{d\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\frac{(d-2)!!p_0^{d/2}}{k(a+1)^{d/2}\log a}\left\{1-\frac{(d-1)(k\log(a+1))^{\frac{d-1}{2}}+(k\log(a+1))^{\frac{d}{2}}}{2(a+1)^{k/2}}\right\}$$

$$\geqslant \frac{d\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\frac{(d-2)!!p_0^{d/2}}{k(a+1)^{d/2}\log a}\left\{1-\frac{(d+1)(k\log(a+1))^{\frac{d+1}{2}}}{2(a+1)^{k/2}}\right\},$$

which completes the proof. □

From Lemma 11, we give the Lemma 12 as below, in order to describe the limiting behavior of conditional density $p(x|H)$

Lemma 12. *(Distillation) Let $p(x)$ be a non-negative integrable function defined on $x \in \mathcal{D}$, where $\mathcal{D} \subseteq \mathcal{R}^d$. Assume $p(x)$ is thrice differentiable with the third-order derivative being bounded. Define $\mathcal{M} = \{x : x = argmax_x(p(x))\}$ to be the collection of all maximum point(s) of $p(x)$. We assume $p(x)$ is locally concave on $\mathcal{M}$, i.e., $\nabla^2 p(x)$ is negative definite for any $x \in \mathcal{M}$. Define a measure on $\mathcal{M}$ as*

$$\mu(x) \propto -\nabla^2 p(x), \quad \forall x \in \mathcal{M},$$

then for any $r > 0$ and sufficiently large a, we have the following result

$$\left\| \frac{\int x p(x)^a dx}{\int p(x)^a dx} - \mathbb{E}_\mu X \right\| = \mathcal{O}\left\{ p_0^{1/2} \frac{\log a}{\sqrt{a+1}} + \left(\frac{\int p(x)}{p_0^{d/2+1}} + \frac{\int \|x\|_1 p(x)}{p_0^{d/2+1}} \right) \frac{\log a}{(a+1)^r} \right\},$$

where $p_0 = \max p(x)$.

Intuitively, Lemma 12 states that when $a \to \infty$, the limiting expectation of distribution $p(x; a)$ will be distilled to the expectation over the domain that maximizes $g(x)$. Assume that our target density function contains no singular points, and $U(x) = -\log f(x)$ has minimum value. From Lemma 12, for any feasible H, when $a \to \infty$, we can obtain $\mathbb{E}_{p(x|H)} x = \mathbb{E}_{u(x)} x, x \in \mathcal{M}$, which are the expectation of the maximum point(s) of $g(x) \triangleq H - U(x)$ (or, minimum point(s) of $U(x)$) that do not depend on H. Based on this result, one can establish a Theorem 1 describing the limiting behavior of $\rho_x(1)$,

Proof. We consider the case where $\mathcal{M}$ is a finite set, which is the most common scenario. The proof for $\mathcal{M}$ being positive measure follows similarly and will be ignored here.

Let $\mathcal{E}(x, \epsilon, \nu, \rho) = \left\{ y : (y-x)^\top (\Sigma_x + 2\nu\rho I)(y-x) \leqslant \epsilon^2 \right\}$ denote the elliptical ball around point x and $\$\mathcal{E}(\mathcal{M}, \epsilon, \nu) = \bigcup_{x_i \in \mathcal{M}} \mathcal{E}(x_i, \epsilon, \nu, \rho_i)$, where the positive definite matrix $\Sigma_x = -\nabla^2 p(x)$ if x is the maximum point and any constants $\nu, \rho, \epsilon > 0$. The maximum radius of the elliptical ball $\rho_{\max}(\nu) = (\tilde{\lambda}_{\min} + 2\nu\rho)^{-1/2}\epsilon$, where $\tilde{\lambda}$ is the eigen value of Σ_x. By solving the equation $2\nu\rho^3 + \tilde{\lambda}_{\min}\rho^2 - \epsilon^2 = 0$ w.r.t. $\rho > 0$, we can obtain a stable or determined $\rho_{\max}$ and we will use this stable value later. Similarly, we can compute $\rho_{\min}(\nu) = (\tilde{\lambda}_{\max} + 2\nu\rho)^{-1/2}\epsilon$.

By the maximality of $\mathcal{M}$ and the smoothness of $A(H)/B(H)$ ($A(H)$ and $B(H)$ are defined in Section A.6), there must exist some ϵ_0 such that for any $\epsilon_1, \epsilon_2 \leqslant \epsilon_0$, we have

$$\mathcal{E}(x_1, \epsilon_1, \nu, \rho_1) \cap \mathcal{E}(x_2, \epsilon_2, \nu, \rho_2) = \varnothing, \quad \forall x_1, x_2 \in \mathcal{M},$$

where $A^c = \mathcal{D} \backslash A$. In addition, since we assume $p(x)$ is locally concave at all $x_i \in \mathcal{M}$ and the second-order derivative of $p(x)$ is continuous, we can quantify the local behavior of $p(x)$ at each point in $\mathcal{M}$ by using the Taylor expansion,

$$p(x) - p(x_i) = \nabla p(x_i)(x - x_i) + \frac{1}{2}(x - x_i)^T \nabla^2 p(x_i)(x - x_i) + \mathcal{O}(\|x - x_i\|^3),$$

where the tail form is due to the existence of the third-order derivative. By definition we have $\nabla p(x_i) = 0$. The local concavity and smoothness ensures that $\nabla^2 p(x_i)$ is negative definite and the largest and smallest eigenvalue can be controlled by some constants, i.e., there exists some constants $L \geqslant l > 0$, such that

$$-L \leqslant \lambda_{\min}(\nabla^2 p(x_i)) \leqslant \lambda_{\max}(\nabla^2 p(x_i)) \leqslant -l, \quad \forall x_i \in \mathcal{M}.$$

Therefore, when ϵ is sufficiently small, we can obtain $\rho_{\max}$ is small as well, and expect $p(x)$ can be well approximated by some local quadratic function. More precisely, defining $p_0 = p(x_i) = \max_{x \in \mathcal{D}} p(x)$, there exists some $\nu > 0$, $\epsilon_0' > 0$, $\epsilon_0' < \epsilon_0$ such that,

$$p(x) - p_0 \geqslant \frac{1}{2}(x - x_i)^T \nabla^2 p(x_i)(x - x_i) - \nu \|x - x_i\|^3 \geqslant -\frac{L}{2}\|x - x_i\|^2 - \nu \|x - x_i\|^3 \tag{A.12}$$

and

$$p_0 - p(x) \geqslant -\frac{1}{2}(x - x_i)^T \nabla^2 p(x_i)(x - x_i) - \nu \|x - x_i\|^3 \geqslant \frac{l}{2}\|x - x_i\|^2 - \nu \|x - x_i\|^3, \tag{A.13}$$

when $x \in \mathcal{E}(x_i, \epsilon_0', \nu, \rho_i)$ for any $x_i \in \mathcal{M}$. The constants chosen are not the tightest, but adequate for the proof.

Now for any $\epsilon > 0$, we will partition the space into $\mathcal{E}(\mathcal{M}, \epsilon, \nu)$ and $\mathcal{D}\backslash\mathcal{E}(\mathcal{M}, \epsilon, \nu)$ (short noted as $\mathcal{E}(\mathcal{M}, \epsilon, \nu)^c$), and the target can then be written as

$$\frac{\int xp(x)^a dx}{\int p(x)^a dx} = \frac{\int x\left(\frac{p(x)}{p_0}\right)^a dx}{\int \left(\frac{p(x)}{p_0}\right)^a dx} = \frac{\int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)} x\left(\frac{p(x)}{p_0}\right)^a dx + \int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} x\left(\frac{p(x)}{p_0}\right)^a dx}{\int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)} \left(\frac{p(x)}{p_0}\right)^a dx + \int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} \left(\frac{p(x)}{p_0}\right)^a dx}$$

$$= \frac{F_1(a) + F_2(a)}{G_1(a) + G_2(a)}.$$

The proof will be done by bounding the four terms and will be divided into two parts.

1. Bounding F_2 and G_2 We first look at F_2 and G_2. When ϵ is small compared to ϵ_0', $\max_{\mathcal{B}(\mathcal{M},\epsilon)^c} p(x)$ will be achieved at some point inside $\mathcal{E}(\mathcal{M}, \epsilon_0', \nu)$. This is easy to show. Since $p_0 > \max_{\mathcal{E}(\mathcal{M},\epsilon_0',\nu)^c} p(x)$, we can always find an $\epsilon_0'' < \epsilon_0'$, such that $\min_{\mathcal{E}(\mathcal{M},\epsilon_0'',\nu)} p(x) > \max_{\mathcal{E}(\mathcal{M},\epsilon_0,\nu)^c} p(x)$. Now for any $\epsilon < \epsilon_0''$, $\max_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} p(x)$ must be achieved within $\mathcal{E}(\mathcal{M}, \epsilon_0', \nu)$. We assume $\epsilon_0' = \epsilon_0''$ in the proof just for simplicity. The above argument along with (A.13) suggests that for sufficiently small ϵ, it holds that

$$p_0 - \max_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} p(x) \geqslant \frac{l}{3} \min_{\mathcal{M}} \rho_{\min}^2.$$

Therefore, for $a \geqslant 1$, we can obtain an explicit decaying rate on $F_2(a)$ and $G(a)$ as

$$|F_2(a)| = \left| \int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} x \left(\frac{p(x)}{p_0}\right)^{a-1} \frac{p(x)}{p_0} dx \right| \leqslant C_1 \left(1 - \frac{l \min_{\mathcal{M}} \rho_{\min}^2}{3p_0}\right)^{a-1},$$

where $C_1 \leqslant \frac{\int \|x\|_1 p(x)}{p_0}$ is a absolute constant only depending on $p(x)$. Similarly, we have for $G_2(a)$ that

$$G_2(a) = \int_{\mathcal{E}(\mathcal{M},\epsilon,\nu)^c} \left(\frac{p(x)}{p_0}\right)^{a-1} \frac{p(x)}{p_0} dx \leqslant C_2 \left(1 - \frac{l \min_{\mathcal{M}} \rho_{\min}^2}{3p_0}\right)^{a-1},$$

where $C_2 \leqslant \int p(x)/p_0$ is an absolute constant depending on $p(x)$.

2. Bounding F_1 and G_1 Next, we consider $F_1(a)$ and $G_1(a)$. Notice that we have

$$F_1(a) = \sum_{x_i \in \mathcal{M}} \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} x \left(\frac{p(x)}{p_0}\right)^a dx = \sum_{x_i \in \mathcal{M}} x_i \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx$$

$$+ \sum_{x_i \in \mathcal{M}} \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} (x - x_i) \left(\frac{p(x)}{p_0}\right)^a dx.$$

In addition, we notice by Cauchy-Schwarz inequality that

$$\left| \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} (x - x_i) \left(\frac{p(x)}{p_0}\right)^a dx \right| \leqslant \rho_{\max} \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx,$$

Therefore, the key to bound F_1 and G_1 is to bound the integral $\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx$, for which we have

$$\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx$$

$$\geqslant \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(1 - \frac{-(x - x_i)^T \nabla^2 p(x_i)(x - x_i) + 2\nu \|x - x_i\|^3}{2p_0}\right)^a dx.$$

If we transform $x - x_i$ to a new variable t, we can obtain a simpler form as

$$\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx \geqslant \int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(1 - \frac{t^T(\Sigma_i + 2\nu\epsilon)t}{2p_0}\right)^a dt.$$

This form can be further simplified by doing transformation $t' = (\Sigma_i + 2\nu\rho_i I)^{1/2} t$ on the right, the key point is to set ρ_i as the solution of equation $2\nu\rho^3 + \tilde{\lambda}_{\min}(\Sigma_i)\rho^2 - \epsilon^2 = 0$, i.e. fix point $\rho_{\max}$. We have

$$\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx \geqslant |\Sigma_i + 2\nu\rho_{\max}|^{-1/2} \int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt. \tag{A.14}$$

(A.14) indicates that $\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx$ will be lower bounded $\int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt$. Thus, the remaining task is to compare this quantity with F_2 and G_2. For a brief summary on this part, we define

$$w_i = \frac{\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx}{\int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt} \quad \text{and} \quad g(a,d,\epsilon) = \int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt$$

we have $w_i \geqslant |\Sigma_i + 2\nu\rho_{\max}|^{-1/2}$.

Similarly, consider the elliptical ball $\mathcal{E}(x_i, \epsilon, -\nu, \rho_i)$ and fixed point $\rho_{\min}$ for equation $\rho = (\tilde{\lambda}_{\max}(\Sigma_i) - 2\nu\rho)^{-1/2}\epsilon$, we can obtain

$$\int_{\mathcal{E}(x_i,\epsilon,-\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx \leqslant |\Sigma_i - 2\nu\rho_{\min}|^{-1/2} \int_{\mathcal{B}(0,\epsilon)} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt. \tag{A.15}$$

Notice we can pick up a very small ϵ' (and the resulted ν) for (A.15) s.t. $\rho'_{\min}$ is bigger than the $\rho_{\max}$ in (A.14). This is possible since the equation (A.15) is actually derived by defining new elliptical ball from the beginning of this proof, i.e.

$$\int_{\mathcal{E}(x_i,\epsilon,\nu,\rho_i)} \left(\frac{p(x)}{p_0}\right)^a dx \tag{A.16}$$

$$\leqslant \int_{\mathcal{E}(x_i,\epsilon',-\nu',\rho'_i)} \left(\frac{p(x)}{p_0}\right)^a dx \leqslant |\Sigma_i - 2\nu\rho_{\min}|^{-1/2} \int_{\mathcal{B}(0,\epsilon')} \left(1 - \frac{\|t\|^2}{2p_0}\right)^a dt. \tag{A.17}$$

and then we have $w_i \leqslant |\Sigma_i - 2\nu'\rho'_{\min}|^{-1/2}$, i.e.

$$|\Sigma_i + 2\nu\rho_{\max}|^{-1/2} \leqslant w_i \leqslant |\Sigma_i - 2\nu'\rho'_{\min}|^{-1/2}. \tag{A.18}$$

Additionally,

$$\frac{F_1(a)}{g(a,d,\epsilon)} = \sum_{x_i \in \mathcal{M}} w_i x_i + \mathcal{O}(\epsilon|\mathcal{M}|), \qquad \frac{G_1(a)}{g(a,d,\epsilon)} = \sum_{x_i \in \mathcal{M}} w_i$$

3. Synthesizing the results The value of $g(a, d, \epsilon)$ has been evaluated in Lemma 11. When ϵ is chosen to be $\epsilon = \sqrt{p_0 k}\frac{\log a}{\sqrt{a+1}}$, we have

$$g(a,d,\epsilon) \geqslant \frac{C_3 p_0^{d/2}}{k(a+1)^{\frac{d}{2}}\log a},$$

where C_3 is an absolute constant depends only on d. On the other hand, we have

$$\frac{G_2(a)}{g(a,d,\epsilon)} \leqslant \frac{C_2 k(a+1)^{\frac{d}{2}}\log a}{C_3 p_0^{d/2}}\left(1 - \frac{l\epsilon^2}{3p_0}\right)^{a-1} \leqslant \frac{C_2 k(a+1)^{\frac{d}{2}}\log a}{C_3 p_0^{d/2}} \frac{1}{2(a+1)^{\frac{kl}{3}}}$$

$$= \frac{C_2 k \log a}{2C_3 p_0^{d/2}}(a+1)^{\frac{d}{2}-\frac{kl}{3}} = \mathcal{O}\left(C_2 \frac{\log a}{(a+1)^r}\right),$$

as long as we choose $k \geqslant \frac{3(d+2r)}{2l}$ for some $r > 0$. Similar result holds for $F_2(a)$ as well. Therefore, we have

$$\frac{F_1(a)+F_2(a)}{F_2(a)+G_2(a)} = \frac{\frac{F_1(a)}{g(a,d,\epsilon)}+\frac{F_2(a)}{g(a,d,\epsilon)}}{\frac{G_1(a)}{g(a,d,\epsilon)}+\frac{G_2(a)}{g(a,d,\epsilon)}} = \frac{\sum_{x_i\in\mathcal{M}} w_i x_i + \mathcal{O}\big(\frac{\int \|x\|_1 p(x)}{p_0^{d/2+1}}\frac{\log a}{(a+1)^r}\big)}{\sum_{x_i\in\mathcal{M}} w_i + \mathcal{O}(\epsilon|\mathcal{M}|) + \mathcal{O}\big(\frac{\int p(x)}{p_0^{d/2+1}}\frac{\log a}{(a+1)^r}\big)}.$$

The relationship (A.18) entails that

$$w_i = |\Sigma_i|^{-1} + \mathcal{O}(\epsilon),$$

and placing the value of ϵ into the equation, we finally have

$$\frac{F_1(a)+F_2(a)}{F_2(a)+G_2(a)} = \frac{\sum_{x_i\in\mathcal{M}} |\Sigma_i|^{-1} x_i + \mathcal{O}\big(\frac{\int \|x\|_1 p(x)}{p_0^{d/2+1}}\frac{\log a}{(a+1)^r}\big)}{\sum_{x_i\in\mathcal{M}} |\Sigma_i|^{-1} + \mathcal{O}\big(p_0^{1/2}\frac{\log a}{\sqrt{a+1}}\big) + \mathcal{O}\big(\frac{\int p(x)}{p_0^{d/2+1}}\frac{\log a}{(a+1)^r}\big)},$$

given the points in $\mathcal{M}$ are bounded, which is true in this case. It then follows naturally that

$$\left\| \frac{\int x p(x)^a dx}{\int p(x)^a dx} - \frac{\sum_{x_i\in\mathcal{M}} |\Sigma_i|^{-1} x_i}{\sum_{x_i\in\mathcal{M}} |\Sigma_i|^{-1}} \right\| = \mathcal{O}\left\{ p_0^{1/2}\frac{\log a}{\sqrt{a+1}} + \left(\frac{\int p(x)}{p_0^{d/2+1}} + \frac{\int \|x\|_1 p(x)}{p_0^{d/2+1}}\right)\frac{\log a}{(a+1)^r} \right\}$$

This completes the whole proof. □

Proof of Theorem 1

Now it's ready to prove the Theorem 1, i.e. $\lim_{a\to\infty} \rho_x(1) = 0$. First, to gain some intuitions, we see from Section A.6 that for the exponential family class of model (with potential function $U(x) = x^\omega$), the $\mathbb{E}_{p(x|H)}x = \frac{A(H)}{B(H)}$ is at order $\mathcal{O}(H^{1/\omega}a^{-1/\omega})$ (using $A(H)$ and $B(H)$ as defined in Section A.6, i.e. $A(H) = \int_{U(x)\leqslant H} x \cdot g(x,H)^{a-1}dx$, $B(H) = \int_{U(x)\leqslant H} g(x,H)^{a-1}dx$, see Section A.6 for more details). In fact, for all of our verified cases that can be analytically derived in Section A.6, $\mathbb{E}_{p(x|H)}x$ can be expressed in $\mathcal{O}(H^r a^{-r})$, where r is a positive constant. Generally, if $\frac{A(H)}{B(H)}$ can be expressed in given form, one can verify below proposition, which leads to $\lim_{a\to\infty} \rho_x(1) = 0$.

Proposition 13. *If* $\mathbb{E}_{p(x|H)}x = \frac{A(H)}{B(H)}$ *can be written as* $\mathbb{E}_{p(x)}x\left(\sum_{r=0}^{\infty}\frac{s_r f_r(H)}{g_r(a)}\right)$*, where* $\lim_{H\to\infty} f_r(H)/H^r = 1$*,* $\lim_{H\to\infty} g_r(a)/a^r = 1$ *and* $\sum_{r=0}^{\infty} s_r = 1$*, one can obtain ,* $\lim_{a\to\infty}\rho_x(1) = 0$*.*

Proof. It can be shown that

$$\lim_{a\to\infty}\mathbb{E}x_t x_{t+1} = \lim_{a\to\infty}\frac{1}{\Gamma(a)Z_1}\int\frac{A(H)^2}{B(H)}e^{-H}dH$$

$$= \frac{\mathbb{E}_p x}{Z_1}\lim_{a\to\infty}\frac{1}{\Gamma(a)}\int\left(\sum_{r=0}^{\infty}\frac{s_r f_r(H)}{g_r(a)}\right)A(H)e^{-H}dH \tag{A.19}$$

$$= \frac{\mathbb{E}_p x}{Z_1}\sum_{r=0}^{\infty}s_r\lim_{a\to\infty}\frac{1}{\Gamma(a)}\int\frac{f_r(H)}{g_r(a)}A(H)e^{-H}dH \tag{A.20}$$

The (A.20) follows by Fubini's theorem. Let $K = H - U(x)$, for any $r \geqslant 0$, we have,

$$\lim_{a\to\infty}\frac{1}{\Gamma(a)}\int\frac{f_r(H)}{g_r(a)}A(H)e^{-H}dH$$

$$= \lim_{a\to\infty}\int\frac{xe^{-U(x)}}{\Gamma(a)}\int\frac{K^r + \mathcal{O}(K^{r-1}U(x)^r)}{a^r + \mathcal{O}(a^{r-1})}K^{a-1}e^{-K}dKdx$$

$$= Z_1\mathbb{E}_p x\cdot\lim_{a\to\infty}\frac{\Gamma(a+r) + \mathcal{O}(U(x)^r)\cdot\Gamma(a+r-1)}{\Gamma(a)(a^r + \mathcal{O}(a^{r-1}))} = Z_1\mathbb{E}_p x$$

Taking together with (A.20), we have

$$\lim_{a\to\infty}\mathbb{E}x_t x_{t+1} = (\mathbb{E}_{p(x)}x)^2$$

Thus, $\lim_{a\to\infty}\rho_x(1) = 0$ □

This establish a sufficient condition for limitation of $\rho_x(1)$ goes to zero. This characterizes the order of $\mathbb{E}_{p(x|H)}x$.

We further provide a proof of Theorem 1 for more general cases in univariate setup, based on Lemma 12 (distillation)

Theorem 14. *For a univariate target distribution,* i.e. $\exp[-U(x)]$ *has finite integral over* $\mathbb{R}$*, if* $U(x)$ *is thrice differentiable with bounded third-order derivative, the one-step autocorrelation of the MG-SS parameterized by* a*, asymptotically approaches zero as* $a \to \infty$*,* i.e.*,* $\lim_{a\to 0} \rho_x(1) = 0$*.*

Proof. Let $p(H) \triangleq \frac{1}{\Gamma(a)Z_1} B(H)e^{-H}$. From Lemma 12, one can obtain that $\lim_{a\to\infty} \frac{A(H)}{B(H)} = C_0$ where C_0 is a constant that is independent with H, where the convergence ratio is characterized by $\mathcal{O}\left\{ \frac{\log a}{\sqrt{a+1}} + \frac{\log a}{(a+1)^r} \right\}$ where r is the unknown constant in Lemma 11. As a result,

$$\lim_{a\to\infty} \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right) = 0, \text{ or, } \lim_{a\to\infty} \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right)^2 = 0 \qquad \text{(A.21)}$$

for any distribution $p(H)$. As a result, we have

$$\begin{aligned}
&\lim_{a\to\infty} \text{Var}_{p(H)} \left(\frac{A(H)}{B(H)} \right) = \lim_{a\to\infty} \int \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right)^2 p(H) dH \\
&\leqslant \lim_{a\to\infty} \sqrt{\int \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right)^2 dH \int p(H) dH} \\
&= \sqrt{\lim_{a\to\infty} \int \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right)^2 dH} \, .
\end{aligned}$$

Because $\frac{A(H)}{B(H)}$ is bounded, it satisfies the conditions of Dominate Convergence Theorem, *i.e.*, the integration and limit operations are exchangeable, thus

$$\lim_{a\to\infty} \text{Var}_{p(H)} \left(\frac{A(H)}{B(H)} \right) \leqslant \int \lim_{a\to\infty} \left(\frac{A(H)}{B(H)} - \mathbb{E}_{p(H)} \frac{A(H)}{B(H)} \right)^2 dH = 0 \, .$$

On the other hand, we have $\lim_{a\to\infty} \text{Var}_{p(H)} \left(\frac{A(H)}{B(H)} \right) \geqslant 0$. As a result, we have

$$\lim_{a\to\infty} \text{Var}_{p(H)} \left(\frac{A(H)}{B(H)} \right) = 0 \, .$$

Substitute this into (A.10), one can obtain,

$$\lim_{a\to\infty} \mathbb{E}x_t x_{t+1} = \lim_{a\to\infty} \int \left[\frac{A(H)}{B(H)}\right]^2 p(H)dH$$

$$= \lim_{a\to\infty} \left(\int \frac{A(H)}{B(H)} p(H)dH\right)^2 + \lim_{a\to\infty} \text{Var}_{p(H)} \left(\frac{A(H)}{B(H)}\right)$$

$$= \lim_{a\to\infty} \left(\int \frac{A(H)}{B(H)} p(H)dH\right)^2 + 0 = \lim_{a\to\infty} \frac{1}{\Gamma(a)Z_1} \times \frac{[\int \frac{A(H)}{B(H)} \cdot B(H)e^{-H}dH]^2}{\int B(H)e^{-H}dH},$$

By changing the integration order, one can obtain, $\int A(H)e^{-H}dH = \Gamma(a)\int xe^{-U(x)}dx$. Similarly, $\int B(H)e^{-H}dH = \Gamma(a)\int e^{-U(x)}dx$. Notice that $Z_1 = \int e^{-U(x)}dx$,

$$\lim_{a\to\infty} \mathbb{E}x_t x_{t+1} = (\mathbb{E}x)^2,$$

Thereby, $\lim_{a\to\infty} \rho_x(1) = 0$ □

A.5.3 Discussions for effective sample size

Effective sample size is associated with the variance of estimator based on MCMC sample (Brooks et al., 2011), and can be used to measure the mixing performance of certain sampler. We hope to show that the ESS will become full sample size, indicating that the limiting behavior of Monte Carlo samples from analytic MG-SS becomes decorrelated, as a approaches infinity. ESS is defined as ESS $= N/(1 + 2 \times \sum_{h=1}^{\infty} \rho_x(h))$, where N is the total number of samples, $\rho_x(h)$ is the h-step autocorrelation function. In this section, we first prove that $\rho_x(h)$ is non-negative. Then, assume the MG sampler is uniformly ergodic, *i.e.*, the total variance distance between the h-th transition kernel and $p(x)$ is bounded by $M(x)t^h$, where $M(x)$ is a bounded function and $0 < t < 1$ (Rosenthal, 1995), under the condition that $\text{Var}_{\kappa_h(x_0|x)}x$ is bounded, where $\kappa_h(x_{t+h}|x_t)$ represents the *h-order transition kernel*, we can show that $\rho_x(h)$ is bounded by $Ct^{h/2}$, with C a positive constant. If we further assume that $\rho_x(h)$ is monotonically decreasing, it can be shown that $\lim_{a\to\infty}$ ESS $= N$. When ESS approaches full sample size, N, the resulting sampler delivers excellent mixing efficiency (Girolami and Calderhead, 2011).

Proof of $\rho_x(h) \geqslant 0$

The h-time-lag autocorrelation function $\rho_x(h)$ can be formulated as

$$\rho_x(h) = \frac{\mathbb{E}_{p(x)}\big[\mathbb{E}_{\kappa_h(x_{t+h}|x)} x_{t+h} x\big] - (\mathbb{E}x)^2}{\text{Var}(x)} \,, \tag{A.22}$$

where, $\kappa_h(x_{t+h}|x_t)$ represents the *h-order transition kernel*, which can be calculated in a recursive manner as:

$$\kappa_1(x_{t+1}|x_t) = \int p(x_{t+1}|y_t) p(y_t|x_t) dy_t \,, \kappa_h(x_{t+h}|x_t) \tag{A.23}$$

$$= \int \kappa_{h-1}(x_{t+1}|x_t) \kappa_1(x_{t+h}|x_{t+1}) dx_{t+1} \,. \tag{A.24}$$

Proposition 15. *The h-order transition kernel, $\rho_x(h)$, is non-negative.*

Proof. From the reversibility shown above, we have,

$$\mathbb{E}x_{2h}x_0 = \mathbb{E}_{p(x_h)}\big[\mathbb{E}_{\kappa_h(x_{2h}|x_h)} x_{2h} \mathbb{E}_{\kappa_h^{-1}(x_0|x_h)} x_0\big] \geqslant \big[\mathbb{E}_{p(x_h)} \mathbb{E}_{\kappa_h(x'|x_h)} x'\big]^2 = (\mathbb{E}x)^2$$

$$\begin{aligned}\mathbb{E}x_{2h+1}x_0 &= \mathbb{E}_{p(x_h, x_{h+1})}\big[\mathbb{E}_{\kappa_h(x_{2h+1}|x_{h+1})} x_{2h+1} \mathbb{E}_{\kappa_h^{-1}(x_0|x_h)} x_0\big] \\ &\geqslant \big[\mathbb{E}_{p(x_h, x_{h+1})} \mathbb{E}_{\kappa_h(x'|x_h)} x'\big]^2 = (\mathbb{E}x)^2\end{aligned}$$

Thus, by definition, $\rho_x(h) \geqslant 0$ □

Discussions for effective sample size

Proposition 16. *(Convergence of moments) Suppose a MCMC sampler is Harris ergodic with invariant distribution $p(x)$. Let $\kappa_h(\cdot, x)$ denote the h-th transition kernel. Define $\hat{x}_h(x_0) \triangleq \mathbb{E}_{\kappa_h(x_0,\cdot)} x_h$ as the expected value of h-time lag sample. If the variance of transition kernel Var$_{\kappa_h(\cdot,x)}(x)$ is bounded, when $h \to \infty$, we have,*

$$\hat{x}_h(x_0) \triangleq \mathbb{E}_{\kappa_h(x_0,\cdot)} x_h \to \mathbb{E}x,$$

Proof. From Harris ergodicity, there exists h' so that for $\forall \epsilon > 0$ and $h \geqslant h'$

$$\int_{\mathcal{X}} |\kappa_h(\cdot, x) - p(x)| dx \quad < \quad \epsilon$$

From Cauchy's inequality, considering the Harris ergodicity, one can obtain the convergence of the first moment as,

$$
\begin{aligned}
&|\mathbb{E}_{\kappa_h(x_0,\cdot)}x_h - \mathbb{E}x| \leqslant \int_{\mathcal{X}} |x| \cdot |\kappa_h(x_0,x) - p(x)|dx \\
&= \int_{\mathcal{X}} |x| \cdot |\kappa_h(x_0,x) - p(x)|^{\frac{1}{2}} |\kappa_h(x_0,x) - p(x)|^{\frac{1}{2}} dx \\
&\leqslant \sqrt{\int_{\mathcal{X}} x^2 |\kappa_h(x_0,x) - p(x)|dx} \sqrt{\int_{\mathcal{X}} |\kappa_h(x_0,x) - p(x)|dx} \leqslant S \times M(x_0)^{\frac{1}{2}} t^{\frac{h}{2}}
\end{aligned}
$$

Where $S \leqslant \text{Var}_{\kappa_h(x_0,x)}x + \text{Var}_{p(x)}x + 2[\mathbb{E}_{p(x)}x]^2$. Thereby,

$$
\lim_{h\to\infty} |\mathbb{E}_{\kappa_h(x_0,\cdot)}x_h - \mathbb{E}x| = 0
$$

□

We propose an assumption as below,

Assumption 1. *(Expected 1-lag sample) The expected 1-lag sample $\hat{x}_1(x)$ lies in between the interval defined by x_0 and $\mathbb{E}x$*

$$
\frac{\hat{x}_1(x) - \mathbb{E}x}{x - \mathbb{E}x} \in [0, 1) \tag{A.25}
$$

If such assumption holds for 1-time lag, the conclusion can be extended to h-time lag using below Lemma

Lemma 17. *(Transitivity) Assume eqn. (1) holds when $h = 1$, it holds for any $h \in \{2, \cdots\}$.*

Proof. We consider using induction, if eqn. (1) holds for $h = 1$. Without losing generality, we assume $x_0 \geqslant \mathbb{E}x$, thus we have $0 \leqslant \hat{x}_{h-1}(x_t) - \mathbb{E}x \leqslant x_0 - \mathbb{E}x$, holds for any x_0,

$$
\begin{aligned}
\hat{x}_h(x_0) - \mathbb{E}x &= \mathbb{E}_{k_1(x_1|x_0)}\hat{x}_h(x_1) - \mathbb{E}x = \mathbb{E}_{k_1(x_1|x_0)}[\hat{x}_h(x_1) - \mathbb{E}x] \\
&\leqslant \mathbb{E}_{k_1(x_1|x_0)}[x_1 - \mathbb{E}x] = \hat{x}_1(x_0) - \mathbb{E}x
\end{aligned}
$$

□

Note that $\mathbb{E}x_h x_0 = \mathbb{E}_{p(x_0)}\hat{x}_h(x_0)x_0$ and $\mathbb{E}_{p(x_0)}\hat{x}_h(x_0) = \mathbb{E}x$, one can validate that

$$\rho_x(h) = \frac{\mathbb{E}_{p(x_0)}[\hat{x}_h(x_0) - \mathbb{E}x](x_0 - \mathbb{E}x)}{\text{Var}(x)} = \frac{1}{\text{Var}(x)} \int [\hat{x}_h(x_0) - \mathbb{E}x] C(x_0) dx_0$$

Where $C(x_0) \leqslant (x_0 - \mathbb{E}x)p(x_0)$. assumption 1 guarantees that $x_0 - \mathbb{E}x$ and $\hat{x}_h(x_0) - \mathbb{E}x$, have same sign. Without loss of generality, we assume $x_0 \geqslant \mathbb{E}x$, thereby,

$$\rho_x(h) \quad \leqslant \quad \frac{S \times t^{\frac{h}{2}}}{\text{Var}(x)} \int M(x_0)^{\frac{1}{2}} C(x_0) dx_0$$

This indicates that the $\rho_x(h)$ is bounded by an exponentially fast decreasing function, at a speed of $\mathcal{O}\left(t^{\frac{h}{2}}\right)$, where $t \in (0, 1)$ is the decay rate of total variance distance between $\kappa_h(\cdot, x)$ and $p(x)$. As a result

$$\text{ESS} = \frac{N}{1 + 2 \times \sum_{h=1}^{\infty} \rho_x(h)} \geqslant \frac{N\text{Var}(x) \times (1-t)^{1/2}}{\text{Var}(x)(1-t)^{1/2} + 2St^{1/2} \int M(x_0)^{\frac{1}{2}} C(x_0) dx_0}$$

The monotonicity of $\rho_x(1)$ could possible be shown by using below Lemmas and assumption 1.

Lemma 18. *(Relative distance) Assume $p(x)$ is a well-defined probability density function with expectation $\mathbb{E}x$, $f_h(x)$ are a family of function of x, parameterized by h. if,*

$$\frac{f_h(x) - \mathbb{E}x}{x - \mathbb{E}x} \in [0, 1)\,,\ \mathbb{E}f_h(x) = \mathbb{E}x$$

We have,

$$\mathbb{E}f_h(x) f_{h'}(x) < \mathbb{E}x f_{h'}(x) \tag{A.26}$$

Proof.

$$\begin{aligned}
& \mathbb{E}f_{h'}(x) f_h(x) - \mathbb{E}x f_h(x) = \mathbb{E}[f_{h'}(x) - \mathbb{E}x][f_h(x)] \\
= \ & \mathbb{E}[f_{h'}(x) - x][f_h(x) - \mathbb{E}x] + \mathbb{E}x[\mathbb{E}f_{h'}(x) - \mathbb{E}x] \\
= \ & \mathbb{E}[f_{h'}(x) - x][f_h(x) - \mathbb{E}x] < 0
\end{aligned}$$

The last inequality holds because the $[f_{h'}(x) - x][f_h(x) - \mathbb{E}x] < 0$ for any x. $\square$

Given above results, note from stationary assumption that $\mathbb{E}_x \hat{x}_h(x) = \mathbb{E}_x \mathbb{E}_{\kappa_h(x'|x)} x' = \mathbb{E}x' = \mathbb{E}x$. From Lemma 18, letting $f_{h'}(x_0) = \hat{x}_1(x_0)$ and $f_h(x_0) = \hat{x}_h(x_0)$, one can obtain $\mathbb{E}x_h x_0 \geqslant \mathbb{E}x_{h-1} x_0$ for $t > 1$. Thus, $\rho_x(h) \geqslant \rho(h-1)$

Proposition 19. *(Monotonicity) Monotonicity for autocorrelation function can be established if assumption 1 holds.* $\rho_x(h) \geqslant \rho(h-1)$

As shown in previous sections, when $a \to \infty$, the $\rho_x(1) \to 0$. Suppose the $\rho_x(h)$ is monotonically decreasing, Together with above result that $\rho_x(h)$ decrease in exponential speed, one can conclude that ESS would converge to the full sample size N.

Theorem 20. *(Limiting ESS) If 1) assumption 1 holds, 2) the variance of transition kernel* $Var_{\kappa_h(\cdot,x)}(x)$ *is bounded, 3) uniform ergodicity can be established. When* $a \to \infty$*, we have,* $ESS \to N$

Proof. From Equation (A.26), for any given a, there exists a H, such that, $Mt^{-H/2} < \rho_x(1)$ for $\forall h > H$. (M denote the constant of the bound function). Thus,

$$\lim_{a\to\infty} \sum_{h=1}^{\infty} \rho_x(h) = \lim_{a\to\infty} \sum_{h=1}^{H} \rho_x(h) + \lim_{a\to\infty} \sum_{h=H+1}^{\infty} \rho_x(h) \leqslant \sum_{h=1}^{H} 0 + \lim_{a\to\infty} \rho_x(1) \frac{Mt^{-H/2}}{1 - t^{-1/2}} = 0$$

Note that $\sum_{h=1}^{\infty} \rho_x(h) > 0$. This indicates, $\lim_{a\to\infty} \sum_{h=1}^{\infty} \rho_x(h) = 0$. Thus, ESS $\to N$. $\square$

A.5.4 MG-HMC mixing performance

The *analytical MG-HMC* (without integration error, with adequate evolving time) is expected to have the same theoretical property as the analytical MG-SS since they are derived from the same problem setup. However, the mixing performance of the two methods could differ significantly, especially when sampling from a multimodal distribution.

Suppose we are sampling from a bimodal distribution. There must exist a critical value y_T, such that when the slicing variable y exceeds y_T, the slice interval $\mathbb{X}$ will have two

disjoint components. The corresponding Hamiltonian, H, will also have a critical value H_T, below which there would be two closed Hamiltonian contours associated with the same energy. The nature of Hamiltonian dynamics only allows moving along a single contour, whereas the analytic MG-SS is able to sample from distributions with disjoint domain, *i.e.*, $\mathbb{X}$ having several disjoint components. As a consequence, the analytical MG-HMC is expected to be less efficient than its analytic MG-SS counterpart. In order to move across different modes, the sampler has to have a large Hamiltonian, $H \geqslant H_T$.

To characterize the performance gap between the analytic MG-SS and MG-HMC, we note that the marginal distribution of H can be obtained as $p(H; a) = \frac{[H-U(x)]^{a-1}e^{-H}}{\Gamma(a)Z_1}$. Therefore,

$$\begin{aligned} P(H \leqslant H_T) &= 1 - \int_{H \geqslant H_T} \frac{\int_{U(x) \leqslant H} [H - U(x)]^{a-1} dx}{\Gamma(a) Z_1} \times e^{-H} dH \\ &= 1 - \frac{1}{Z_1} \int \int_{H \geqslant \max(U(x), H_T)} [H - U(x)]^{a-1} \times e^{-H} / \Gamma(a) dH dx \\ &= \frac{1}{Z_1} \int_{U(x) \leqslant H_T} \frac{\gamma(a, H_T - U(x))}{\Gamma(a)} \times e^{-U(x)} dx \ , \end{aligned}$$

where $\gamma(\cdot, \cdot)$ denotes the lower incomplete Gamma function. Note that $F(a, x) = \frac{\gamma(a,x)}{\Gamma(a)}$ is the cumulative distribution function of Gamma$(a, 1)$, thus is monotonically decreasing with a, and as $a \to \infty$, $F(a, x) \to 0$. This implies that when a is large enough, the chances of reaching an energy level that restricts the traversing across modes can be arbitrarily small. Note that as in Section A.4, the mass parameter m have no impact on the analysis. As a result, in theory the analytical MG-HMC with large value of a is particularly advantageous for sampling multimodal distributions.

A.6 Theoretical autocorrelations and ESS for 1D cases

For derivation conveniency we first introduce several additional denotations. Note that $H = -\log y$, denoting $g(x, H) = H - U(x), s.t. U(x) \leqslant H$ to be the kinetic energy function *w.r.t.*

x conditioning on Hamiltonian H. We further denoting $A(H) = \int_{U(x)\leqslant H} x \cdot g(x,H)^{a-1}dx$, $B(H) = \int_{U(x)\leqslant H} g(x,H)^{a-1}dx$. $\mathbb{E}x_t x_{t+1}$ can be rewritten as,

$$\mathbb{E}x_t x_{t+1} \quad = \quad \int \frac{e^{-H}}{\Gamma(a)Z_1} \cdot \frac{A(H)^2}{B(H)} dH$$

A.6.1 Theoretical autocorrelation for sampling exponential distribution

For $U(x) = x/\theta, x > 0$, using the definition of $A(H)$ and $B(H)$ from above, one can derive from algebra that,

$$B(H) = \frac{\theta H^a}{a} \quad , \quad A(H) = \frac{H^{a+1}\theta^2}{a+a^2} \quad , \quad \rho_x(1) = \frac{Ex_t x_{t+1} - \theta^2}{\theta^2} = \frac{1}{a+1}$$

Follow similar derivation, one could validate that,

$$\rho_x(h) = \frac{1}{(a+1)^h}\,, \text{ESS} = \frac{Na}{a+2} \tag{A.27}$$

A.6.2 Theoretical autocorrelation for sampling positive-truncated Gaussian

In positive-truncated Gaussian case, $U(x) = x^2, x > 0$, we have,

$$B(H) = \frac{H^{a-\frac{1}{2}}\sqrt{\pi}\Gamma(a)}{2\Gamma\left(a+\frac{1}{2}\right)} \quad , \quad A(H) = \frac{H^a}{2a}$$

$$\rho_x(1) = \frac{Ex_t x_{t+1} - 1/\pi}{1/2 - 1/\pi} = \frac{1}{(\pi/2-1)}\left[\frac{\Gamma\left(a+\frac{1}{2}\right)\Gamma\left(a+\frac{3}{2}\right)}{\Gamma(a+1)^2} - 1\right]$$

Thereby

$$\rho_x(1) = \tfrac{1}{(\pi/2-1)}\left[\tfrac{\Gamma\left(a+\frac{1}{2}\right)\Gamma\left(a+\frac{3}{2}\right)}{\Gamma(a+1)^2} - 1\right],\ \rho_x(h) = \rho_x(1)^h\,,$$

$$\text{ESS} = \tfrac{N}{1+2\rho_x(1)/(1-\rho_x(1))}$$

A.6.3 *Theoretical autocorrelation for $U(x) = x^\omega$*

The exponential family class of model introduced in (Roberts and Tweedie, 1996) have the potential energy with form $U(x) = x^\omega, x \geqslant 0, \omega > 0$. For these model we have,

$$
\begin{aligned}
A(H) &= \frac{H^{a-1+2/\omega}\Gamma(2/\omega)\Gamma(a)}{\omega\Gamma(a+2/\omega)}, \; B(H) = \frac{H^{a-1+1/\omega}\Gamma(1/\omega)\Gamma(a)}{\omega\Gamma(a+1/\omega)} \\
\rho_x(1) &= \frac{\left[\frac{\Gamma(a+3/\omega)\Gamma(a+1/\omega)}{\Gamma(a+2/\omega)\Gamma(a+2/\omega)} - 1\right] \times \frac{\Gamma(2/\omega)^2}{\Gamma(1/\omega)^2}}{\frac{\Gamma(3/\omega)}{\Gamma(1/\omega)} - \frac{\Gamma(2/\omega)^2}{\Gamma(1/\omega)^2}}
\end{aligned}
$$

A rough estimation for above $\rho_x(1)$ using Stirling's formula shows that $\rho_x(1) = \mathcal{O}(1/(a+1))$. Note that this results holds for $U(x) = x^\omega, x \geqslant 0, \omega > 0$, where θ is a scale parameter. Interestingly, (Livingstone et al., 2016) showed that if the integration time (leap-frog steps) is fixed, the geometric ergodicity holds only when $\omega \in [1, 2]$. However, with a random integration time the geometric ergodicity can be established for any $\omega > 0$. For this reason, we use a random integration time in our experiments.

A.6.4 *Theoretical autocorrelation for Gamma*

We conducted numerical theoretical analysis on Gamma$(r, 1)$, where $r = 2, 3$. For each a, one can apply numerical methods for calculating the integrals $A(H)$ and $B(H)$. The $\rho_x(1)$ can then be calculated from Equation (A.9). The continuous function is plotted by interpolating from functional evaluation at $\{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$ using quadrature.

A.7 Remedy strategies for numerical issue and convergence issue

A.7.1 *Remedy strategies for numerical issue*

To ameliorate numerical difficulties associated with large a (Figure A.4 (left)), we propose two remedies, i.e. **Reflection** and **Softened kinetics**.

Reflection: Reflection performs well in low-dimensional phase space, but may suffer from sticky behavior in high-dimensional cases. This is because the probability of a sign

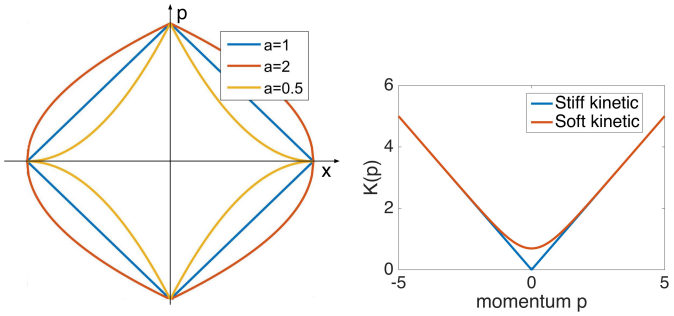


FIGURE A.4: Left: large a gives "stiff" Hamiltonian trajectories. Right: Soft kinetic vs stiff kinetic

change occurrence is high; recall there are at least 2^D turnovers. Besides, the transformation $(x, p) \mapsto (x, -p)$ depends on current (x, p) and the determinant of the corresponding Jacobian is not one. This may lead to discrepancy in the sampled distribution, though the empirical discrepancy is not very large.

Softened kinetics: We define softened kinetics as

$$K(p) = -g(p) + 2/c \log(1 + e^{cg(p)}), g(p) = \text{sign}(p)|p|^{1/a}/m. \tag{A.28}$$

Where c is a softening parameter. The comparison between standard kinetics ("stiff" kinetics) and softened kinetics are shown in Figure A.4 (right). This kinetic share same tail behavior with stiff kinetic $K(p) = p^{1/a}/m$, and is differentiable, rendering much less numerical error. It asymptotically approaches standard kinetics when $c \to \infty$. One can use dimensional-wise importance sampling to correct the monomial gamma distribution to get momentum from the distribution defined by softened kinetics. However, when dealing with higher dimensional problems, the rejection rate of importance sampling step is high ($\mathcal{O}(D)$), which brings additional computational concerns.

In our tested scenarios, softened kinetics usually performs better than reflection in low dimensional problems, but fails to outperform reflection in high dimensional problems. For specific application, we advocate to try both to get the maximum of performance. In addition, note that for softened kinetics 1) the kinetics is modified, thus the theory can not be directly applied. 2) Softened kinetics requires heavier computation than reflection. 3)

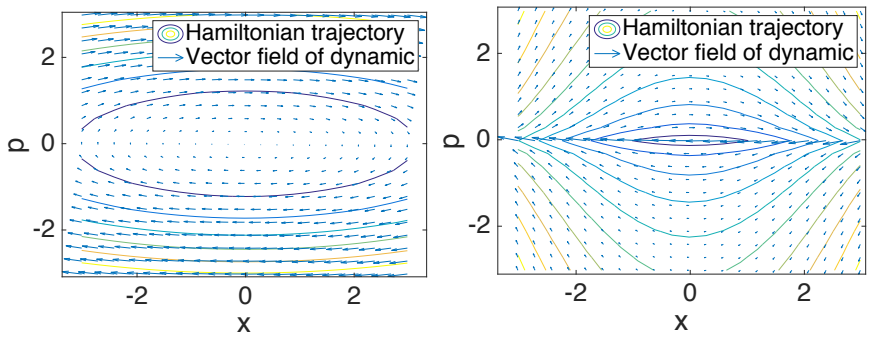


FIGURE A.5: The Hamiltonian trajectory when $a = 0.5$ (upper) and $a = 2$ (lower).

The parameter c requires tuning.

Extensions that account for geometric information (Girolami and Calderhead, 2011) or using more accurate numerical integrator (Chao et al., 2015) may also help alleviating the numerical problems.

A.7.2 Remedy strategies for convergence issue

If the sampler is initialized in the tail region of a light-tailed target distribution, MG-HMC with $a > 1$ may converge arbitrarily slow to the true target distribution, i.e., the burn-in period could take arbitrarily long time. In Figure A.5 we show this issue. When $a > 1$, the numerical difficulty increases, and samples of x may move slowly in the light-tailed region (far from $x = 0$). To avoid being arbitrarily slow convergence with random initialization scheme, we suggest two strategies. First, we suggest using a step-size decay scheme, *e.g.*, $\epsilon = \max(\epsilon_1 \rho^t, \epsilon_0)$. In our experiments we use $(\epsilon_1, \rho) = (10^6, 0.9)$, where ϵ_0 is problem-specific. This allows the sampler to move larger to avoid slow convergence within burnin-steps, then gradually decreases to a normal step-size to perform stationary sampling. This approach empirically alleviates the slow convergence problem in our tested scenarios. Second, we suggest to initialize the sampler from a local maximum of posterior estimated from optimization methods, such as gradient descent method. This strategy ensure the sample is not initialized in light-tailed region. Third, with fixed computational budget, we encourage using reducing the leapfrog step in each iteration and increase the total number

of iterations, which is essentially increasing the *resampling rate* of momentum variables. However, we note that the MG-HMC sampler with large a may still be less sensitive when sampling from the light-tail, where a more sophisticated methods adaptively selecting a during the sampling procedure are presumably useful. It can be proved that this adaptively selecting a will still leave invariante target distribution. Nevertheless, we left this for further investigation.

A.8 Analytical MG-SS

A.8.1 Analytical MG-SS for exponential distribution

For sampling an exponential distribution Exp(θ), *i.e.* $U(x) = \theta x, x > 0$, analytic MG-SS is available for all a. The procedure is given by

Algorithm 3: Analytical MG-SS for exponential distribution

Input: Total sample size S
Output: Sample results
Initialization: Choose initial sample point x_0
for $t = 1$to$S$ do **do**
  Sample $K_t \sim$ Gamma$(a, 1)$, find $H_t = x_t/\theta + K_t$
  Sample $\tau \sim$Uniform(0, 1)
  Sample $x_{t+1} = (1 - \tau^{1/a})\theta H_t$
end for

A.8.2 Analytical MG-SS for positive-truncated Gaussian distribution

Here we provide the algorithms for MG-SS in sampling positive-truncated Gaussian, *i.e.* $U(x) = x^2, x > 0$, for $a = 0.5, 1, 2$

Algorithm 4: Analytical MG-SS for half-Gaussian, $a = 0.5$

Input: Total sample size S
Output: Sample results
Initialization: Choose initial sample point x_0
for $t = 1$to$S$ do **do**
  Sample $p_t \sim N\left(0, 1/\sqrt{2}\right)$, find $H_t = x_t^2 + p_t^2$
  Sample $\tau \sim$Uniform(0, π)
  Sample $x_{t+1} =$ abs $\left(\sqrt{H_t}\cos(\tau)\right)$
end for

For $a = 1$, this is standard slice sampler.

Algorithm 5: Analytical MG-SS for half-Gaussian, $a = 1$

Input: Total sample size S
Output: Sample results
Initialization: Choose initial sample point x_0
for $t = 1$to$S$ do **do**
  Sample $p_t \sim$ Gamma$(1, 1)$, find $H_t = x_t^2 + p_t$
  Sample $\tau \sim$Uniform$(0, 2\sqrt{H_t})$
  Sample $x_{t+1} = \left(\tau - \sqrt{H_t}\right)$
end for

For $a = 2$, we solve a cubic function to estimate $\mathbb{X}$, the resulting procedure is:

Algorithm 6: Analytical MG-SS for half-Gaussian, $a = 2$

Input: Total sample size S
Output: Sample results
Initialization: Choose initial sample point x_0
for $t = 1$to$S$ do **do**
  Sample $K_t \sim$ Gamma$(2, 1)$, find $H_t = x_t^2 + K_t$
  Sample $\tau \sim$Uniform$(0, 1)$
  Compute $C = \left(2\sqrt{\tau(\tau - 1)} + 1 - 2\tau\right) H_t^{1/2}$
  Sample $x_{t+1} = \text{abs}\left(-\frac{1+\sqrt{3}i}{2}C - \frac{1-\sqrt{3}i}{2C}\beta\right)$
end for

The results are shown in table A.1. "AR" denotes acceptance rate.

	Th. $\rho_x(1)$	Th. ESS	SS $\rho_x(1)$	SS ESS	HMC $\rho_x(1)$	HMC ESS	HMC AR	HMC time(s)
$a = 0.5$	0.67	6,000	0.6620	6,204	0.6711	6,069	0.99	30
$a = 1$	0.50	10,000	0.4868	10,227	0.5218	9,773	0.99	32
$a = 2$	0.33	15,000	0.3265	15,547	0.3777	14,028	0.98	31
$a = 3$	0.25	18,000	0.2494	17,507	0.2741	17,488	0.95	31
$a = 4$	0.20	20,000	0.2108	19,229	0.2555	17,775	0.92	30

Table A.1: 1D exponential distribution.

	Th. $\rho_x(1)$	Th. ESS	SS $\rho_x(1)$	SS ESS	HMC $\rho_x(1)$	HMC ESS	HMC AR	HMC time(s)
$a = 0.5$	0.4787	10,576	0.4736	10,705	0.4802	10,510	0.99	42
$a = 1$	0.3120	15,731	0.3040	15,457	0.3061	15,595	0.99	41
$a = 2$	0.1830	20,718	0.1770	21,468	0.1937	20,498	0.99	43
$a = 3$	0.1293	23,132	-	-	0.1665	21,303	0.96	65
$a = 4$	0.0999	24,552	-	-	0.1508	22,115	0.94	120

Table A.2: 1D positive-truncated Gaussian.

$r=2$	Th. $\rho_x(1)$	$\rho_x(1)$	ESS	$r=3$	Th. $\rho_x(1)$	$\rho_x(1)$	ESS
$a=0.5$	0.4600	0.3523	10457	$a=0.5$	0.3729	0.2182	15507
$a=1$	0.3023	0.3008	15248	$a=1$	0.2030	0.1979	18416
$a=2$	0.1891	0.1838	20979	$a=2$	0.1290	0.1223	23486
$a=3$	0.1372	0.1684	21703	$a=3$	0.0931	0.1572	22106
$a=4$	0.1077	0.2430	19062	$a=4$	0.0728	0.2116	19541

Table A.3: MG-HMC results of Gamma distribution

A.9 Complimentary experimental results

A.9.1 Simulation results for 1D toy cases

The 1D simulation results are summarized in Table A.1(Exp) ,Table A.2(half-Gaussian) and Table A.3(Gamma).

The comparison of $\rho_x(1)$ is provided in Figure A.7. The skewness for Exp(θ), Gamma$(2,\theta)$, $\mathcal{N}_+(0,\theta)$ and Gamma$(3,\theta)$ are $\{2, 1.41, 0.99, 1.15\}$, respectively. We observed that a large value of shape parameter of Gamma distribution would lead to a lower $\rho_x(1)$, as a fixed. Informally, this seems suggest that the skewness of the target distribution would influence the behavior of MG samplers. A more skewed distribution tends to have higher autocorrelation $\rho_x(h)$, and lower ESS.

We compare the empirical $\rho_x(1)$ and ESS of the analytic MG-SS and MG-HMC with their theoretical values of each cases in Figure A.6. In the Gamma distribution case, analytic derivations of the autocorrelations and ESS are difficult, thus we resort to a numerical approach to compute $\rho_x(1)$ and ESS.

In principle, as a becomes larger, it would be desirable to choose a smaller ϵ to compensate for the numerical hardness. Empirically, the choice of m and ϵ is dependent. A small value of m would compensate the demand for choosing a small step-size to certain extent. However, optimal performances were achieved by tuning both of them. Presumably, m will influence the relative scale of the contour along x-axis and p-axis, thus tuning m will influence the general shape of the contour, which may be beneficial in some cases. For the exponential and positive-truncated Gaussian cases, as a becomes larger, the autocorrelation

decreases from 1 to a small value close to zero, meanwhile the ESS increases to approach the total sample size. However, the acceptance rates also decrease.

The results for analytic MG-SS match well with the theoretical results, however MG-HMC seems to suffer from practical difficulties when a is large, evidenced by results gradually deviating from the theoretical values. This issue is more prominent in the Gamma case, where the autocorrelation first decreases then becomes larger.

Figure A.6 demonstrate results for exponential distribution (a,b) and $\mathcal{N}_+$ (c,d). $\rho_x(1)$ for Gamma distribution with parameters $r = 2$ (e) and $r = 3$ (f)

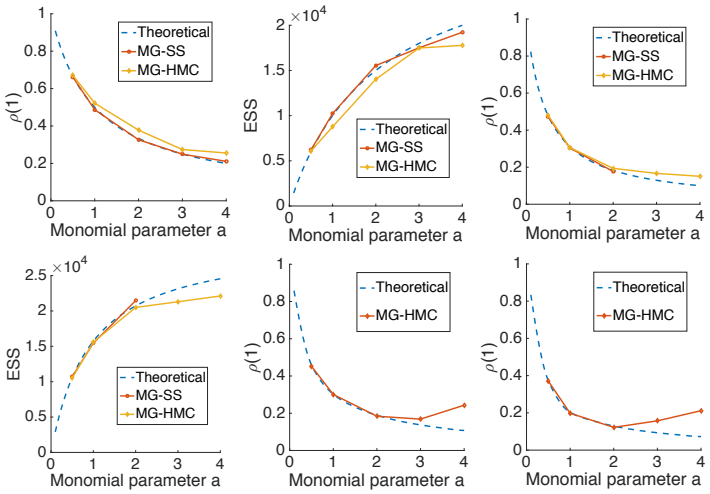


FIGURE A.6: Theoretical and empirical $\rho_x(1)$ and ESS of toy distributions

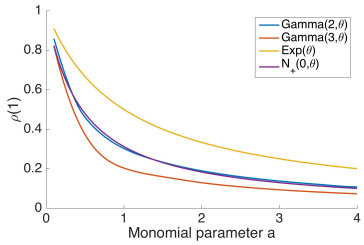


FIGURE A.7: Comparison of $\rho_x(1)$ for 4 simulated univariate cases

A.9.2 Comparison between MG-HMC $a = 1$ with standard SS

1D unimodal toy cases To validate that when $a = 1$, the resulting sampler can be understood as standard slice sampling, we also compared with standard slice sampling using

Dataset(dim)	Australian(15)	German(25)	Heart(14)	Pima(8)	Ripley(7)	Cavaran (87)
ϵ	0.1	0.05	0.14	0.1	0.14	0.03
m when $a = 0.5$	10	10	10	10	10	10
m when $a = 1$	2	2	2	2	2	2
m when $a = 2$	1	1	1	1	1	1

Table A.4: BLR setup (dimensionality in parenthesis)

Exponential	m	ϵ	AR	Gaussian(+)	m	ϵ	AR
$a = 0.5$	1	0.05	0.991	$a = 0.5$	1	0.05	0.996
$a = 1$	1	0.05	0.982	$a = 1$	1	0.1	0.981
$a = 2$	0.15	0.05	0.980	$a = 2$	0.15	5×10^{-3}	0.983
$a = 3$	0.02	1×10^{-3}	0.94	$a = 3$	0.02	5×10^{-5}	0.962
$a = 4$	3×10^{-3}	5×10^{-8}	0.90	$a = 4$	3×10^{-3}	2.5×10^{-8}	0.946
Gamma, $r = 2$	m	ϵ	AR	Gamma, $r = 3$	m	ϵ	AR
$a = 0.5$	1	0.05	0.986	$a = 0.5$	1	0.05	0.986
$a = 1$	1	0.1	0.982	$a = 1$	1	0.05	0.983
$a = 2$	0.15	5×10^{-3}	0.965	$a = 2$	0.15	0.01	0.972
$a = 3$	0.02	2.5×10^{-5}	0.902	$a = 3$	0.02	1×10^{-5}	0.885
$a = 4$	3×10^{-3}	2.5×10^{-8}	0.783	$a = 4$	3×10^{-3}	2.5×10^{-8}	0.766

Table A.5: Experiment setups for 1D simulated study

1D	ESS	$\rho_x(1)$	AR	2D	ESS	$\rho_x(1)$	AR
$a = 0.5$	5175	0.60	0.98	$a = 0.5$	4691	0.67	0.96
$a = 1$	10157	0.43	0.97	$a = 1$	16349	0.60	0.87
$a = 2$	24298	0.11	0.92	$a = 2$	18007	0.53	0.78

Table A.6: Effective sample size of MG-HMC for 1D and 2D bimodal distribution.

MG-HMC($a = 1$)	Exponential	half-Gaussian	Gamma($r = 2$)	Gamma($r = 3$)
$\rho_x(1)$	0.5218	0.3061	0.3008	0.1979
ESS	9,773	15,595	15,248	18416
Standard SS	Exponential	half-Gaussian	Gamma($r = 2$)	Gamma($r = 3$)
$\rho_x(1)$	0.5198	0.3039	0.3011	0.1954
ESS	9,622	16,051	15,092	18874

Table A.7: Comparison between MG-HMC $a = 1$ with standard SS in toy cases

MG-HMC($a = 1$)	1D	2D	Standard SS	1D	2D
$\rho_x(1)$	0.43	0.60	$\rho_x(1)$	0.056	0.697
ESS	10157	16349	ESS	27469	9566

Table A.8: Comparison between MG-HMC with standard SS in 1D and 2D cases

Min ESS	A	G	H	P	R	C	ICA
MG-HMC($a = 1$)	4308	4353	4591	4664	4226	36	3029
Standard SS	8.1	9.7	5.7	5.2	3.7	42.9	3.29

Table A.9: Comparison between MG-HMC with standard SS in BLR and ICA experiments

doubling and shrinking scheme (Neal, 2003). The resulting ESS and $\rho(1)$ is almost identical to analytical MG-SS and MG-HMC with $a = 1$. The results are reported in Table A.7.

Bimodal 1D and 2D cases We also applied standard slice sampler with dimensional-wise doubling and shrinking scheme (Neal, 2003) for these bimodal tasks. In 1D case, the standard slice sampler yields ESS close to full sample size, while in 2D cases, the resulting ESS is 9533. This is reasonable because when sampling from these bimodal symmetric distribution, in theory, analytical MG-SS gives full ESS. However the slice sampler is usually less efficient when dealing with more than one dimension. We note that there are more sophisticated methods for performing slice sampling in multidimensional (Murray et al., 2009), however we leave the comparison for future investigation. The results are reported in Table A.8.

BLR and ICA We reported the result for standard slice sampler (Neal, 2003) in Table A.9. In general, standard slice sampler with adaptive search fails to achieve a comparable results with other compared methods when applying to multi-dimensional scenarios.

A.9.3 100 dimensional multivariate Gaussian

We assessed the performance of MG-HMC for sampling 100 dimensional Gaussian distribution. The target Gaussian distribution has zero-mean and diagonal covariance matrix, where the diagonal elements are uniformly drawn from $[0, 10]$. We collected 5000 MC samples after applying 2500 burn-in rounds. We compared the efficiency and accuracy of MG-HMC with $a = \{0.5, 1, 2\}$. For each scheme, we use 5 different leapfrog step-sizes ϵ_t, $t = \{1 \cdots 5\}$, where $\epsilon_{t+1} = 0.8\epsilon_t$, so as to make the acceptance rates ranges from 0.4 to

0.9. MG-HMC with $a = 1$ achieved highest median effective sample size, as well as lowest mean square error between empirically estimated parameters and truth (Figure A.8).In figure A.8 we show the scatter plot of mean squared error of estimated covariance and median ESS for simulated 100D Gaussian distribution. Number labels denote the stepsize index. The maximum acceptance rates for $a = \{0.5, 1, 2\}$ is $\{0.98, 0.96, 0.63\}$, respectively. MG-HMC with $a = 2$ failed to outcompete other two tested schemes, probably due to the increasing numerical hardness.

A.10 Experimental setup

1D toy synthetic problems we use a random integration time (leap-frog steps) uniformly drawn from $(20, 180)$, which has better convergence guarantee as suggested by (Livingstone et al., 2016). Step sizes and m are selected such that the acceptance rates fall within $[0.6, 0.9]$, as suggested by (Betancourt et al., 2014). The parameters for MG-HMC in our simulation study is selected by grid search. Specifically, we tried stepsize $\epsilon \in (0.05, 0.025, 1 * 10^{-2}, \cdots, 10^{-8})$, and mass parameter $m \in 2, 1, 0.5, 0.25, 0.15, 0.05, 0.02, 0.003$. For univariate distributions the optimal setup is provided in the Table A.5.

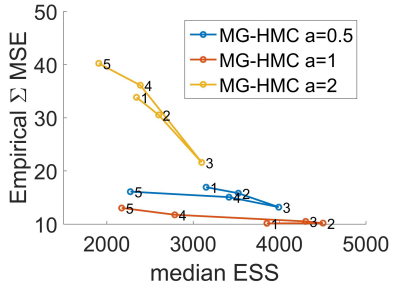


FIGURE A.8: 100D Gaussian

Simulated bimodal experiments Each leap-frog update has $(50 - l, 50 + l), l = 20$ steps, the step-size is set as $\epsilon = 0.05$. The mass parameter for 1D case is chosen to be $m = \{5, 1.2, 0.4\}$ for $a = \{0.5, 1, 2\}$, respectively. For 2D case, the mass matrix is obtained by a mass parameter m times the identity matrix, where $m = \{1, 0.1, 0.35\}$ for $a = \{0.5, 1, 2\}$.

Bayesian logistic regression We follow the setup in (Girolami and Calderhead, 2011) and (Chao et al., 2015) for BLR experiment. For data $\mathbf{X} \in \mathbb{R}^{d \times N}$, response variable

Dataset (D)	A (15)	G (25)	H (14)	P (8)	R (7)	C (87)
$a = 0.5$	0.92	0.78	0.92	0.90	0.89	0.82
$a = 1$	0.93	0.79	0.93	0.88	0.86	0.84
$a = 2$	0.92	0.79	0.96	0.94	0.87	0.69

Table A.10: The avg AUROC for each method. Dimensionality in parenthesis.

$\mathbf{t} \in \{0,1\}^N$ and target parameters $\boldsymbol{\beta} \in \mathbb{R}^d$, if we impose a Gaussian prior $\mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$ (where $\alpha > 0$) on $\boldsymbol{\beta}$, the log posterior is given by (Girolami and Calderhead, 2011),

$$\mathcal{L}(\boldsymbol{\beta}) = \boldsymbol{\beta}^T\mathbf{X}\mathbf{t} - \sum_{n=1}^{N} log(1 + \exp(\boldsymbol{\beta}^T\mathbf{X}_{n,\cdot}^T)) - \frac{\boldsymbol{\beta}^T\boldsymbol{\beta}}{2\alpha}$$

We collect 5000 MC samples, with 1000 burn-in samples. For each dataset, we use a random integration time (leap-frog steps) uniformly drawn from $(20, 180)$, which has better convergence guarantee as suggested by (Livingstone et al., 2016). Step sizes and m are selected such that the acceptance rates fall within $[0.6, 0.9]$, as suggested by (Betancourt et al., 2014). The stepsize and mass parameter varies from dataset to dataset (Table A.4). To deal with numerical problems, in Table A.4, for $a = 1$ we use reflection, for $a = 2$ we use softened kinetics. The softening parameter c is set as $[0.3, 0.2, 0.2, 0.2, 0.3, 0.2]$ for the 6 datasets, respectively.

The average AUROC based on 10 folds cross-validation for each method is reported in Table A.10

Independent component analysis For data $\mathbf{X} \in \mathbb{R}^{d\times N}$ and target parameters $\mathbf{W} \in \mathbb{R}^{d\times d}$, the joint likelihood is given by (Hyvärinen and Oja, 2000; Chao et al., 2015),

$$p(\mathbf{X}, \mathbf{W}) = |\det(\mathbf{W})|^N \prod_{i=1}^{N}\prod_{j=1}^{d} p_j(w_j^T x_i) \prod_{k,l} \mathcal{N}(W_{kl}; 0, \sigma)$$

In our experiments, we set the variance of the Gaussian prior to 100. The $p_j(w_j^T x_i) = \{4cosh^2(1/2y_{ij})\}^{-1}$, where $\mathbf{y}_i = \mathbf{W}\mathbf{x}_i$ (Chao et al., 2015; Korattikara et al., 2013).

Appendix B

Additional theories and results for chapter 4 (SGMGT)

B.1 The Main Theorem

We provide the following theorem to characterize the stationary distribution of the stochastic process with SDEs in (4.9).

Theorem 2. *The stochastic process generated from SDEs* (4.9) *converges to a stationary distribution* $p(\Gamma) \propto \exp(-H(\Gamma))$*, where* $H(\Gamma)$ *is defined as in* (4.6)*.*

Proof. We first show that the Fokker-Planck equation holds for the proposed SDE and probability density $p(\Gamma)$,

$$\nabla_\Gamma \cdot p(\Gamma) V(\Gamma) = \nabla_\Gamma \nabla_\Gamma^T : [p(\Gamma) D(\Gamma)]$$

Here the $\nabla \triangleq (\partial/\partial\theta, \partial/\partial p, \partial/\partial\xi)$. $\cdot$ represents a vector inner product and : denotes the matrix double dot product, *i.e.*, $X : Y = \text{Tr}(X^T Y)$. In order to show FP equation holds, we look at both side of the equation.

The left hand side can be written as

$$\begin{aligned}
& \nabla_\Gamma \cdot p(\Gamma)V(\Gamma) \\
= \; & [\frac{\partial V(\Gamma)}{\partial \Gamma} - \frac{\partial H(\Gamma)}{\partial \Gamma} V(\Gamma)]p(\Gamma) \\
= \; & \{\sigma_\theta[(\nabla U(\theta))^2 - \nabla^2 U(\theta)] \\
& + \sigma_p[(\nabla K(p))^2 - \nabla^2 K(p)] \\
& + \sigma_\xi[(\nabla F(\xi))^2 - \nabla^2 F(\xi)]\} p(\Gamma)
\end{aligned}$$

For the right hand side,

$$\begin{aligned}
& \nabla_\Gamma \nabla_\Gamma^T : [p(\Gamma)D(\Gamma)] \\
= \; & \sigma_\theta \nabla\nabla^T : p(\Gamma) + \sigma_p \nabla\nabla^T : p(\Gamma) + \sigma_\xi \nabla\nabla^T : p(\Gamma) \\
= \; & \{\sigma_\theta[(\nabla U(\theta))^2 - \nabla^2 U(\theta)] \\
& + \sigma_p[(\nabla K(p))^2 - \nabla^2 K(p)] \\
& + \sigma_\xi[(\nabla F(\xi))^2 - \nabla^2 F(\xi)]\} p(\Gamma)
\end{aligned}$$

For stationary distribution,

$$\frac{\partial p(\Gamma, t)}{\partial t} = 0$$

As a result, the equality in (B.1) holds. The stochastic process defined by (4.9) is preserved by the dynamic. Alternatively, one can leverage the recipe from Ma et al. (2015) to recover the same conclusion, by setting semi-definite matrix $D = \text{Diag}([\sigma_\theta, \sigma_p, \sigma_\xi])$ and skew-symmetric Q to be

$$\begin{pmatrix} 0 & -I & 0 \\ I & 0 & \gamma\nabla K(p) \\ 0 & -\gamma\nabla K(p) & 0 \end{pmatrix}$$

Note that under the softened kinetics, the $K_c(p)$ is twice differentiable, and $\nabla K_c(p)$ is Lipschitz continuous. Thus the Fokker-Planck equation holds, leading to a stationary

Algorithm 7: SGMGT/SGMGT-D with Euler integrator.

Input: Monomial parameter a, variances $\{\sigma_\theta, \sigma_p, \sigma_\xi\}$, thermostat variance γ, resampling frequency $\{T_p, T_\xi\}$, and learning rate h.
Initialized θ_0, p_0 and ξ_0 (all to zero).
for $t = 1, 2, \ldots$ **do**
  Evaluate stochastic gradient $\nabla\tilde{U}(\theta)$ on mini-batch.
  After each T_p iterations, resample momentum, p_t, from $\propto \exp(-K_c(p))$ using rejection sampling.
  For each T_ξ iterations, resample thermostats, ξ_t, from $\mathcal{N}(\sigma_\xi, \sqrt{\gamma})$.
  Generate random variables $\epsilon_\theta, \epsilon_p, \epsilon_\xi \sim \mathcal{N}^D(0, 1)$.
  Compute first and second order derivatives for kinetics $K_c(p)$ as in (4.8).
  $\theta_t = \theta_{t-1} - \sigma_\theta \nabla\tilde{U}(\theta)h + \nabla K_c(p)h + \sqrt{2\sigma_\theta h}\epsilon_\xi$.
  $p_t = p_{t-1} - \nabla\tilde{U}(\theta)h - \xi \odot (\nabla K_c(p))h + \sqrt{2\sigma_p h}\epsilon_p$.
  $\xi_t = \xi_{t-1} + \gamma[\nabla K_c(p) \odot \nabla K_c(p) - (\nabla^2 K_c(p))]h - \frac{\sigma_\xi}{\gamma}(\xi - \sigma_p)h + \sqrt{2\sigma_\xi h}\epsilon_\xi$.
end for

distribution invariant to target distribution. Another remark is that the resampling process for p and ξ will still lead to the same invariante distribution $p(\Gamma)$, since the resampling process is directly drawing sample from the marginal distribution. Finally, it can be proved that the corresponding Itô diffusion of our algorithm in (4.9) is non-reversible. This speed up the convergence speed to equilibrium, because it is known that a reversible process convergences slower than its non-reversible counter part Hwang et al. (2005). □

B.2 SGMGT-D/SGMGT-D with Euler integrator

B.3 Details for softened kinetics

We provide the details for the derivation of softened kinetics. Note that in the SDE (4.9), only $\nabla K_c(p)$ and $\nabla^2 K_c(p)$ is involved. For $a = 1$, we consider

$$K_c(p) = -g(p) + 2/c \log(1 + e^{cg(p)}), g(p) = p/m.$$

which gives

$$\nabla K_c(p) = \tfrac{1}{m}\psi(g(p)), \nabla^2 K_c(p) = \tfrac{1}{m^2}\psi'(g(p)).$$

Where, $\psi(x) = \frac{e^{cx}-1}{e^{cx}+1}$ is the hyperbolic tangent function (tanh) with the softening parameter c, $\psi'(x) = \frac{2ce^{cx}}{(e^{cx}+1)^2}$.

For $a = 2$, we consider

$$K_c(p) = g(p) + \frac{4}{c(1 + e^{cg(p)})}, g(p) = |p|^{1/2}/m.$$

which gives

$$\begin{aligned} \nabla K_c(p) &= \tfrac{1}{2m}\text{sign}(p)\psi(g(p))^2|p|^{-1/2}, \\ \nabla^2 K_c(p) &= \tfrac{1}{2m^2}\psi(g(p))\psi'(g(p))|p|^{-1} - \tfrac{1}{4m}\psi^2(g(p))|p|^{-3/2}. \end{aligned}$$

In general, for arbitrary a, we consider setting the

$$\nabla K_c(p) = \frac{a}{m}\psi(g(p))^a|p|^{-1/a},$$

Such specification will yield a differentiable softened kinetics function by computing the integral, which is tractable for positive value of a. However, in practice, as suggested by Zhang et al. (2016d) the optimal a would usually between $[0.5, 2]$. We would suggest consider using $a = 1$ or $a = 2$ for general inference tasks.

B.4 Synthetic multi-well potential problem

The five-well potential is defined as:

$$U(\theta) \triangleq e^{\frac{3}{4}\theta^2 - \frac{3}{2}\sum_{i=1}^{10} c_i \sin\left(\frac{1}{4}\pi i(\theta+4)\right)}, \tag{B.1}$$

where $c = (-0.47, -0.83, -0.71, -0.02, 0.24, 0.01, 0.27,$
$-0.37, 0.87, -0.37)$ is a vector, c_i is the i-th element of c.

B.5 Symmetric Splitting Integrators for SGMGT

The first-ordered Euler integration results in high discretization error in Hamiltonian dynamic updating of HMC. In Chen et al. (2016a), a symmetric splitting scheme is leveraged

to reduce the numerical error. We applied the softened kinetics $K_c(p)$, and set $F(\xi)$ as $\frac{(\xi-\sigma_p)^2}{2\gamma}$. In this symmetric splitting scheme, the Hamiltonian is split into sub-componenents, and for each sub-componenents an individual SDE is applied on. The resulting discretization is symplectic and second-ordered:

$$
\begin{aligned}
A : d\Gamma &= \begin{pmatrix} -\sigma_\theta \nabla \tilde{U}(\theta) + \nabla K_c(p) \\ 0 \\ f(\Gamma) \end{pmatrix} dt/2 \\
B : d\Gamma &= \begin{pmatrix} 0 \\ -\xi \cdot \nabla K_c(p) \\ 0 \end{pmatrix} dt/2 \\
O : d\Gamma &= \begin{pmatrix} 0 \\ -\nabla \tilde{U}(\theta) \\ 0 \end{pmatrix} dt + D(\Gamma) dW
\end{aligned}
$$

Here we denote $f(\Gamma) \triangleq \gamma[(\nabla K_c(p))^2 - (\nabla^2 K_c(p))] - \frac{\sigma_\xi}{\gamma}(\xi - \sigma_p)$ for clarity. The sub-SDE under sub-SDE B is analytically solvable. Following Chen et al. (2015), for $a \neq 1/2$, the updating procedure follows an ABOBA scheme, given by

$$
\begin{aligned}
A &: \theta_{t+1/3} = \theta_t + \nabla K_c(p) h/2, \xi_{t+1/3} = \xi_t + f(\Gamma) h/2 \\
B &: p_{t+1/3} = [p_t^{(2a-1)/a} - \frac{2a-1}{a^2} \xi_{t+1/2} h/2]^{a/(2a-1)} \\
O &: \theta_{t+2/3} = \theta_{t+1/3} + \sqrt{2\sigma_\theta} \epsilon_\theta \\
&\quad p_{t+2/3} = p_{t+1/3} - \nabla \tilde{U}(\theta)) h/2 + \sqrt{2\sigma_p} \epsilon_p, \\
&\quad \xi_{t+2/3} = \xi_{t+1/3} + \sqrt{2\sigma_\xi} \epsilon_\xi \\
B &: p_{t+1} = [p_{t+2/3}^{(2a-1)/a} - \frac{2a-1}{a^2} \xi_{t+2/3} h/2]^{a/(2a-1)} \\
A &: \theta_{t+1} = \theta_{t+2/3} + \nabla K_c(p) h/2, \xi_{t+1} = \xi_{t+2/3} + f(\Gamma) h/2
\end{aligned}
$$

When $a = 1/2$, it follows the splitting scheme with standard SGNHT (Chen et al., 2015).

Algorithms	σ_p	σ_θ	σ_ξ	γ	c	h
SGNHT	10	-	-	1	-	2e-4
SGNHT-D	10	0.1	0.1	1	-	2e-4
SGMGT-D (a=1)	10	0.1	0.1	1	3	1e-5
SGMGT-D (a=2)	10	0.1	0.1	1	5	5e-5

Table B.1: Experimental setup for discriminative RBM

B.6 Experimental setups for DRBM

The hyper-parameter setups for the DRBM experiments are provided as below. We select the hyperparameters based on the performance on validation dataset. The algorithm will be early stopped if the validation error start to increase. The selection is based on a grid search. For σ_p, σ_ξ and σ_θ we select from $\{0.001, 0.01, 0.1, 1, 10\}$. For the softening parameter c we select from $\{3, 5, 8\}$. We fixed the $m = 1$ and $\gamma = 1$. The stepsize is chosen from $\{1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4\}$. The T_p and T_ξ are set as 100 and 100, respectively.

For SGLD, we use a stepsize of $1e-5$

B.7 Experimental setups for RNNs

The hyper-parameter setups for the RNNs experiments are similar to the DRBM experiments. For σ_p, σ_ξ and σ_θ we select from $\{0.01, 0.1, 1, 10\}$. For the softening parameter c we select from $\{3, 5, 8\}$. We fixed the $m = 1$ and $\gamma = 1$. The stepsize of SGMGT-D/SGMGT is chosen from $\{1e-3, 1.5e-3, 2e-3, 2.5e-3, 3e-3\}$. The T_p and T_ξ are set as 100 and 100, respectively. We also incorporate a decay scheme for stepsize, *i.e.* the stepsize is divided by a decaying factor $\alpha = 1.1$ for each scan of dataset (*i.e.* each epoch). The gradient estimated on a subset of data is clipped to have a maximum value of 5 as in Chen et al. (2016a) for each dimension to prevent updates from a large gradient value to blow up the objective loss. For JSB we use a stepsize of $2e-3$ for SGMGT, for other three datasets (Piano, Muse, Nott) we use a stepsize of $3e-3$. For SGLD, we use a stepsize of $1e-3$,

Algorithms	m	σ_p	σ_θ	σ_ξ	γ	c
SGNHT	1	10	-	-	1	-
SGNHT-D	1	10	0.01	0.01	1	-
SGMGT/SGMGT-D (a=1)	1	10	0.1	0.01	1	5
SGMGT/SGMGT-D (a=2)	1	10	0.1	0.01	1	3

Table B.2: Experimental setup for discriminative RNNs

for SGNHT the stepsize is set as $5e-5$. The other hyperparameters are provided in B.2

B.8 Additional figure for RNNs experiment

We provide the traceplot of one parameter in RNN experiment of JSB dataset. We choose this parameter at random. Generally, the SGMGT with $a = 2$ seems to demonstrate more random walk behavior than SGMGT with $a = 1$

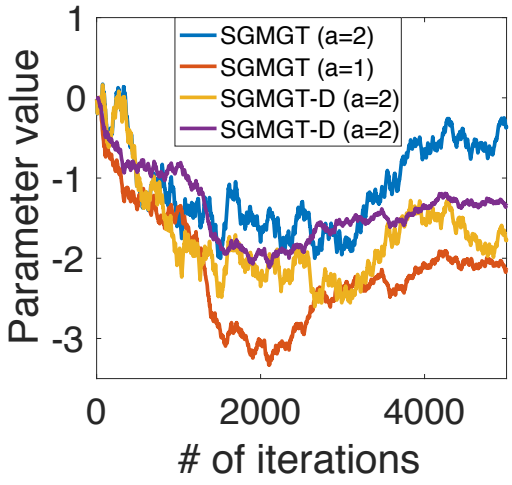


FIGURE B.1: Traceplot for RNN experiments

B.9 Additional results for RNN experiments

Here we provide the results of several optimization methods, the results are taken from Chen et al. (2016a).

Algorithms	Piano.	Nott.	Muse.	JSB.
Adam	8.00	3.70	7.56	8.51
RMSprop	7.70	3.48	7.22	8.52
SGD-M	8.32	3.60	7.69	8.59
SGD	11.13	5.26	10.08	10.81
HF	7.66	3.89	7.19	8.58
SGD-M	8.37	4.46	8.13	8.71

Table B.3: Test negative log-likelihood results on polyphonic music datasets using RNN.

B.10 Convergence property

Proof. This follows the proof for general SG-MCMC algorithms. Specifically, in SGMGT, the generator of the corresponding SDE is defined as:

$$\mathcal{L}f(x) \triangleq \left(F(x) \cdot \nabla + \frac{1}{2} \left(\Sigma\Sigma^T \right) : \nabla\nabla^T \right) f(x) \, ,$$

where

$$x = (\theta, p, \xi),$$

$$F(x) = \begin{pmatrix} -\sigma_\theta \nabla U(\theta) + \nabla K_c(p) \\ -\nabla \tilde{U}(\theta) - (\sigma_p + \gamma \nabla F(\xi)) \nabla K_c(p) \\ \gamma \left[(\nabla K_c(p))^2 - \nabla^2 K_c(p) \right] - \sigma_\xi \nabla F(\xi) \end{pmatrix},$$

$$\Sigma = \begin{pmatrix} \sqrt{2\sigma_\theta} & 0 & 0 \\ 0 & \sqrt{2\sigma_p} & 0 \\ 0 & 0 & \sqrt{2\sigma_\xi} \end{pmatrix}.$$

After introducing stochastic gradients, in each iteration t, the generator is perturbed by:

$$\Delta V_t = \left(\nabla \tilde{U}(\theta) - \nabla U(\theta) \right) \cdot \left(\nabla - \sigma_\theta \nabla \right) \, ,$$

such that $\tilde{\mathcal{L}}_t = \mathcal{L} + \Delta V_t$, where $\tilde{\mathcal{L}}_t$ is the local generator for the SDE in iterator t.

After defining these notation, we follows the proofs of Theorem 2 and Theorem 3 in (Chen et al., 2015).

The proof for the bias: Following Theorem 2 in Chen et al. (2015), in the decreasing

step size setting, the split flow can be written as:

$$\mathbb{E}\left(\psi(\mathbf{X}_{lh})\right) = \left(\mathbb{I} + h_l\tilde{\mathcal{L}}_l\right)\psi(\mathbf{X}_{(l-1)h}) + \sum_{k=2}^{K}\frac{h_l^k}{k!}\tilde{\mathcal{L}}_l^2\psi(\mathbf{X}_{(l-1)h}) + O(h_l^{K+1})\ .$$

Similarly, the expected difference between $\tilde{\phi}$ and $\bar{\phi}$ can be simplified using the step size sequence (h_l) as:

$$\mathbb{E}\left(\tilde{\phi} - \bar{\phi}\right) \tag{B.2}$$

$$= \frac{1}{S_L}\left(\mathbb{E}\left(\psi(\mathbf{X}_{Lh})\right) - \psi(\mathbf{X}_0)\right) - \sum_{k=2}^{K}\sum_{l=1}^{L}\frac{h_l^k}{k!S_L}\tilde{\mathcal{L}}_l^k\psi(\mathbf{X}_{(l-1)h}) + O(\frac{\sum_{l=1}^{L}h_l^{K+1}}{S_L}) \tag{B.3}$$

Similar to the derivation in Chen et al. (2015), we can derive the following bounds $k = (2, \cdots, K)$:

$$\sum_{l=1}^{L} h_l^k\mathbb{E}\tilde{\mathcal{L}}_l^k\psi(\mathbf{X}_{(l-1)h}) \tag{B.4}$$

$$= O\left(\sum_{l=1}^{L}\left((h_l^{k-1} - h_{l-1}^{k-1})\tilde{\mathcal{L}}_l^{k-1}\psi(\mathbf{X}_{(l-1)h}) + h_l^{K+1}\right)\right) = O\left(1 + \sum_{l=1}^{L}h_l^{K+1}\right)\ . \tag{B.5}$$

Substitute (B.4) into (B.2) and collect low order terms, we have:

$$\mathbb{E}\left(\tilde{\phi} - \bar{\phi}\right) = \frac{1}{S_L}\left(\mathbb{E}\left(\psi(\mathbf{X}_{Lh})\right) - \psi(\mathbf{X}_0)\right) + O(\frac{\sum_{l=1}^{L}h_l^{K+1}}{S_L})\ .$$

As a result, the bias can be expressed as:

$$\begin{aligned}\left|\mathbb{E}\tilde{\phi} - \bar{\phi}\right| &\leqslant \left|\frac{1}{S_L}\left(\mathbb{E}\left[\psi(\mathbf{X}_{Lh})\right] - \psi(\mathbf{X}_0)\right) + O(\frac{\sum_{l=1}^{L}h_l^{K+1}}{S_L})\right| \\ &\lesssim \left|\frac{1}{S_L}\right| + \left|\frac{\sum_{l=1}^{L}h_l^{K+1}}{S_L})\right| = O\left(\frac{1}{S_L} + \frac{\sum_{l=1}^{L}h_l^{K+1}}{S_L}\right)\ .\end{aligned}$$

Taking $L \to \infty$, both terms go to zero by assumption.

The proof for the MSE: Following similar derivations as in Theorem 2 in Chen et al. (2015), we have that

$$\sum_{l=1}^{L}\mathbb{E}\left(\psi(\mathbf{X}_{lh})\right)=\sum_{l=1}^{L}\psi(\mathbf{X}_{(l-1)h})+\sum_{l=1}^{L}h_l\mathcal{L}\psi(\mathbf{X}_{(l-1)h})$$
$$+\sum_{l=1}^{L}h_l\Delta V_l\psi(\mathbf{X}_{(l-1)h})+\sum_{k=2}^{K}\sum_{l=1}^{L}\frac{h_l^k}{k!}\tilde{\mathcal{L}}_l^k\psi(\mathbf{X}_{(l-1)h})+C\sum_{l=1}^{L}h_l^{K+1}\,.$$

Substitute the Poisson equation into the above equation and divided both sides by S_L, we have

$$\hat{\phi}-\bar{\phi}=\frac{\mathbb{E}\psi(\mathbf{X}_{Lh})-\psi(x_0)}{S_L}+\frac{1}{S_L}\sum_{l=1}^{L-1}\left(\mathbb{E}\psi(\mathbf{X}_{(l-1)h})+\psi(\mathbf{X}_{(l-1)h})\right)$$
$$+\sum_{l=1}^{L}\frac{h_l}{S_L}\Delta V_l\psi(\mathbf{X}_{(l-1)h})+\sum_{k=2}^{K}\sum_{l=1}^{L}\frac{h_l^k}{k!S_L}\tilde{\mathcal{L}}_l^k\psi(\mathbf{X}_{(l-1)h})+C\frac{\sum_{l=1}^{L}h_l^3}{S_L}\,.$$

As a result, there exists some positive constant C, such that:

$$\mathbb{E}\left(\hat{\phi}-\bar{\phi}\right)^2\leqslant C\mathbb{E}\left(\underbrace{\frac{1}{S_L^2}\left(\psi(\mathbf{X}_0)-\mathbb{E}\psi(\mathbf{X}_{Lh})\right)^2}_{A_1}\right. \tag{B.6}$$
$$\underbrace{+\frac{1}{S_L^2}\sum_{l=1}^{L}\left(\mathbb{E}\psi(\mathbf{X}_{(l-1)h})-\psi(\mathbf{X}_{(l-1)h})\right)^2}_{A_2}$$
$$\left.+\sum_{l=1}^{L}\frac{h_l^2}{S_L^2}\left\|\Delta V_l\right\|^2+\underbrace{\sum_{k=2}^{K}\left(\sum_{l=1}^{L}\frac{h_l^k}{k!S_L}\tilde{\mathcal{L}}_l^k\psi(\mathbf{X}_{(l-1)h})\right)^2}_{A_3}+\left(\frac{\sum_{l=1}^{L}h_l^3}{S_L}\right)^2\right) \tag{B.7}$$

A_1 can be bounded by assumptions, and A_2 is shown to be bounded by using the fact that $\mathbb{E}\psi(\mathbf{X}_{(l-1)h})-\psi(\mathbf{X}_{(l-1)h})=O(\sqrt{h_l})$ from Theorem 2 in Chen et al. (2015). Furthermore, similar to the proof of Theorem 2 in Chen et al. (2015), the expectation of A_3 can also be bounded by using the formula $\mathbb{E}[\mathbf{X}^2]=(\mathbb{E}\mathbf{X})^2+\mathbb{E}[(\mathbf{X}-\mathbb{E}\mathbf{X})^2]$ and (B.4). It turns

out that the resulting terms have order higher than those from the other terms, thus can be ignored in the expression below. After some simplifications, (B.6) is bounded by:

$$\mathbb{E}\left(\hat{\phi}-\bar{\phi}\right)^2 \lesssim \sum_l \frac{h_l^2}{S_L^2}\mathbb{E}\left\|\Delta V_l\right\|^2 + \frac{1}{S_L} + \frac{1}{S_L^2} + \left(\frac{\sum_{l=1}^{L} h_l^{K+1}}{S_L}\right)^2$$

$$= C\left(\sum_l \frac{h_l^2}{S_L^2}\mathbb{E}\left\|\Delta V_l\right\|^2 + \frac{1}{S_L} + \frac{(\sum_{l=1}^{L} h_l^{K+1})^2}{S_L^2}\right) \tag{B.8}$$

for some $C > 0$, this completes the first part of the theorem. We can see that according to the assumption, the last two terms in (B.8) approach to 0 when $L \to \infty$. If we further assume $\frac{\sum_{l=1}^{\infty} h_l^2}{S_L^2} = 0$, then the first term in (B.8) approaches to 0 because:

$$\sum_l \frac{h_l^2}{S_L^2}\mathbb{E}\left\|\Delta V_l\right\|^2 \leqslant \left(\sup_l \mathbb{E}\left\|\Delta V_l\right\|^2\right)\frac{\sum_l h_l^2}{S_L^2} \to 0\,. \tag{B.9}$$

As a result, we have $\lim_{L\to\infty}\mathbb{E}\left(\hat{\phi}-\bar{\phi}\right)^2 = 0$.

□

B.11 Proof for Lemma 1

To prove Lemma 1, we first introduce the following lemma from Geyer (2005).

Lemma 21 (Geyer (2005))**.** *Suppose μ is a probability distribution and for each z in the domain of domain of μ there is a Markov kernel P_z satisfying $\pi = \pi P_z$, and suppose that the map $(z, x) \rightarrowtail P_z(x, A)$ is jointly measurable for each A. Then*

$$Q(x, A) = \int \mu(\mathrm{d}z) P_z(x, A) \tag{B.10}$$

defines a kernel Q that is Markov and satisfies $\pi = \pi Q$.

Proof. Detailed proof can be found in Chapter 3 of Geyer (2005). □

Now it is ready to prove Lemma 1.

Proof of Lemma 1. First, we note that the momentum (or other auxiliary variables) is resampled from the stationary distribution of the Itô diffusion. As a result, for each model parameter θ, it corresponds to a Markov kernel P_θ with the stationary Gaussian density. According to Lemma 21, the composition of the numerical integrator in SGMGT and the resampling forms a Markov kernel $Q(\theta, A)$, such that

$$\pi_h = \pi_h Q \ . \tag{B.11}$$

The above equation means that π_h is also the stationary distribution of the Markov kernel Q, which completes the proof. □

B.12 Proof for Lemma 2

Proof. First, the optimal bias and MSE bounds in Proposition 2 are given by:

$$\text{Bias: } \left|\mathbb{E}\hat{\phi}_T - \bar{\phi}\right| = O\left(T^{-1/2}\right) \ , \tag{B.12}$$

$$\text{MSE: } \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 = O\left(T^{-2/3}\right) \ . \tag{B.13}$$

Let the number of samples in each resampling period to be $(T_l)_{l=1}^L$, and denote $T \triangleq \sum_{l=1}^L T_l$. Further denote the sample average in the l-th resampling period to be:

$$\hat{\phi}_{T_l} \triangleq \frac{1}{T_l} \sum_{l=1}^{T_l} \phi(x_l^{(T_l)}) \ , \tag{B.14}$$

where $\{x_l^{(T_l)}\}$ denotes samples in the l-th resampling period. The final sample average is defined as:

$$\hat{\phi}_T \triangleq \sum_{l=1}^{L} \frac{T_l}{\sum_{l'=1}^{T_{l'}}} \hat{\phi}_{T_l} \ . \tag{B.15}$$

As a result, the bias can be bounded as:

$$\left|\mathbb{E}\hat{\phi}_T - \bar{\phi}\right| = \left|\mathbb{E}\sum_{l=1}^{L}\frac{T_l}{\sum_{l'=1}^{T_{l'}}}\hat{\phi}_{T_l} - \bar{\phi}\right| \tag{B.16}$$

$$=\frac{1}{\sum_l T_l}\left|\sum_{l=1} T_l\left(\mathbb{E}\hat{\phi}_{T_l} - \bar{\phi}\right)\right| \leqslant \sum_l \frac{T_l}{\sum_{l'} T_{l'}}\left|\mathbb{E}\hat{\phi}_{T_l} - \bar{\phi}\right| \tag{B.17}$$

$$=\sum_l \frac{T_l}{\sum_{l'}} T_{l'} O\left(\frac{1}{T_l h} + h\right) = \sum_l \frac{1}{\sum_{l'}} T_{l'} O\left(\frac{1}{h} + T_l h\right) \tag{B.18}$$

Optimizing over h, we have

$$\left|\mathbb{E}\hat{\phi}_T - \bar{\phi}\right| = \sum_l \frac{1}{\sum_{l'}} T_{l'} O\left(T_l^{1/2}\right) \leqslant O\left(\frac{(\sum_l T_l)^{1/2}}{\sum_l T_l}\right) = O\left(T^{-1/2}\right) \ , \tag{B.19}$$

which is the same as the optimal bias bound for SGMGT.

The proof for the optimal MSE bound follows similarly. □

Appendix C

Additional derivations and results for chapter 5

C.1 Augmented MCMC inference

Latent counts:

$$x_{mknt} \sim \text{Multinomial}\left(x_{mnt}, [\hat{\lambda}_{m1nt}, \ldots, \hat{\lambda}_{mKnt}]\right) , \tag{C.1}$$

where $\hat{\lambda}_{m\cdot nt} = \lambda_{m\cdot nt} / \sum_k \lambda_{mknt}$, $\lambda_{mknt} = \psi_{mk}\theta_{knt}h_{knt}$.

Factor loadings columns:

$$\boldsymbol{\psi}_k \sim \text{Dirichlet}\left(\eta_\psi + x_{1k\cdot\cdot}, \ldots, \eta_\psi + x_{Mk\cdot\cdot}\right) ,$$

where $x_{mk\cdot\cdot} = \sum_{n,t} x_{mknt}$.

Factor intensities:

$$\theta_{knt} \sim \text{Gamma}\left(r_k h_{knt} + x_{\cdot knt}, b_\theta\right) , \tag{C.2}$$

$$r_k \sim \text{Gamma}\left(1 + \sum_{n,t} \ell_{knt}, 1 - \sum_{n,t} h_{knt}\log(1 - b_\theta)\right) ,$$

where $x_{\cdot knt} = \sum_m x_{mknt}$ and $\ell_{knt} \sim \mathrm{CRT}(x_{\cdot ntk}, r_k)$ is the Chinese Restaurant Table (CRT) distribution (Zhou and Carin, 2015). Note that the conditional update for r_k is obtained by leveraging the gamma-Poisson conjugacy and the data augmentation scheme described in (Zhou and Carin, 2015). Briefly, the gamma-Poisson construction of $x \sim \text{Poisson}(\lambda), \lambda \sim \text{Gamma}(r, b/(1-b))$ can be represented as $m = \sum_{t=1}^{L} \ell_t$, $\ell_t \sim \text{Log}(b), L \sim \text{Poisson}(-r\ln(1-b))$, where $\text{Log}(\cdot)$ denote the logarithmic distribution (Johnson et al., 2005).

Factor activations:

$$h_{knt} \sim \delta\left(x_{\cdot knt} = 0 \wedge z_{\cdot kt} = 0\right) \text{Bernoulli}(\hat{\pi}_{knt}) + \delta\left(x_{\cdot knt} > 0 \vee z_{knt} > 0\right) ,$$

where

$$\hat{\pi}_{knt} = \frac{\tilde{\pi}_{knt}}{\tilde{\pi}_{knt} + (1 - \pi_{knt})} ,$$

$$\tilde{\pi}_{knt} = \pi_{knt}(1 - b_\theta)^{r_k}(1 - b_w)^{s_k} .$$

Factor intensities for transition model:

$$w_{knt} \sim \text{Gamma}\left(s_k h_{knt} + z_{knt}, \tilde{b}_w(\tau_{nt})\right) ,$$

$$s_k \sim \text{Gamma}\left(1 + \sum_{n,t} \ell'_{knt}, 1 - \sum_{n,t} h_{knt} \log(1 - \tilde{b}_w(\tau_{nt}))\right) ,$$

$$\tilde{b}_w(\tau_{nt}) = \frac{b_w}{b_w + (1 - b_w)\tau_{nt}}$$

Similar to (C.2), $\ell'_{knt} \sim \mathrm{CRT}(z_{knt}, r_k)$ is drawn from the CRT distribution.

Latent counts for transition model:

$$z_{knt} \sim \delta(h_{knt} = 1)\text{Poisson}_+(\tilde{\lambda}_{knt}) ,$$

where $\text{Poisson}_+(\tilde{\lambda})$ is the truncated Poisson distribution with rate $\tilde{\lambda}$. Note that from (5.5), if $h_{knt} = 1$ when $z_{knt} > 0$, then conditioned on $h_{knt} = 1$, latent count variable $z_{knt}|h_{knt} = 1$ is truncated Poisson with rate $\tilde{\lambda}_{knt}$.

In all the expressions above, summations over n, t and m run with $n = 1, \ldots, N$, $t = 1, \ldots, T_n$ and $m = 1, \ldots, M$, respectively.

The conditional posteriors for $\mathbf{\Phi}$, $\boldsymbol{\pi}_0$ are similar to the ones presented above, thus omitted here for conciseness. In practice, we initialize model parameters at random from their corresponding prior distributions.

C.2 Additional experiments

Here we provide 3 additional experiments that we used to evaluate our proposed method in chapter 5. The first consists of artificially generated data with different levels of sparsity, selected to show the computational efficiency of our learning procedure, implemented using GPUs. The next two datasets are publicly available, and consist of collections of text documents over time. Specifically, we consider the US presidential State of the Union addresses (Han et al., 2014), and NIPS abstracts (Globerson et al., 2007). We also consider one public dataset of binary time-series data, consisting of multiple polyphonic music transcriptions (Boulanger-Lewandowski et al., 2012).

C.2.1 Artificial data

We wish to evaluate quantitatively the parallelized implementation of our model using GPUs, to sample from the multinomial distribution, in terms of runtime. We compare this *efficient* implementation with our own Matlab and C++ implementation, which only differs from the GPU version in the multinomial sampler routine, where most of the runtime is spent. There are two routines implemented in C++ in both versions, the CRT and the truncated Poisson samplers. For the experiment we used a standard desktop machine with 4 cores at 3.2GHz and 24Mb RAM. The GPU is a Geforce GTX 750i with 640 cores

and 2Mb RAM. We consider datasets of size $M = T = 2000$, $N = 1$ and different observed sparsity levels, namely fractional levels of sparsity equal to members of the set $\{0.7, 0.8, 0.9\}$. Figure C.1 shows average runtime in seconds for a full Gibbs iteration (a cycle through the updates in Section 5.3), as a function of the number of binary latent variables, K, in the dynamic Poisson factor model. In Figure C.1 we use $\log_2$-transformed runtime per full Gibbs iteration (in seconds) vs. number of binary latent variables, for datasets of fixed size, but different sparsity levels. In all cases we observe speedups ranging from 85 to 250x with an average of about 120x, which constitutes a substantial acceleration, considering the relatively outdated GPU we used for the experiments. It is worth noting that sampling from multinomial distributions represents about 90% of the total runtime.

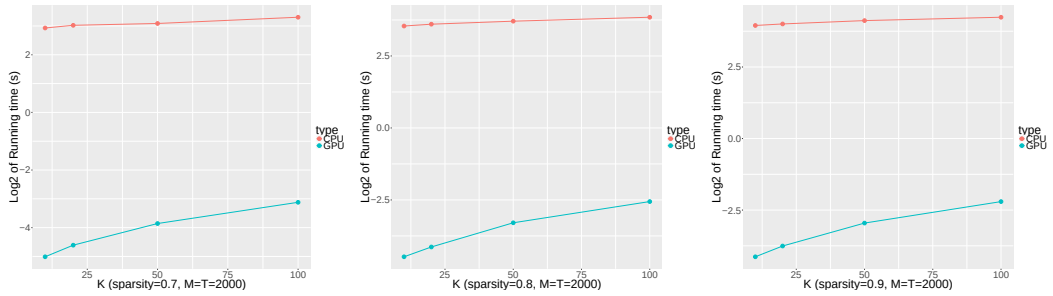


FIGURE C.1: Computational complexity of dynamic Poisson factor analysis on artificial data.

C.2.2 NIPS Abstracts

This dataset contains distributions of words (including authors) in all NIPS papers from years 1988 to 2003. Again, we consider the dataset as one time-series ($N = 1$) with 17 time-points, where each year is a document, thus $\tau_{nt} = 1, \forall t$ and $n = 1$. We preprocess the dataset using the same criteria used for the State of the Union data, which results in $M = 14,036$ distinct words.

In this experiment, we focus on a quantitative evaluation using the performance metrics previously defined. Provided that TSBN consistently outperforms DRFM, GP-DPFA and TRBM (Gan et al., 2015a), we only consider comparisons against TSBN here. In

Model	Mean Precision	Predictive precision
Dynamic PFA	0.876	0.664
TSBN	0.810	0.538

Table C.1: Average predictive precision for NIPS abstracts.

both models, the number of latent variables is set to $K = 50$. Predictive performance is summarized in Table C.1, where mean precision was computed on held-out subsets of each year. Predictive precision was computed on the last year. From Table C.1 we can see that our model outperforms TSBN by marked margins, using a 80/20% split for mean precisions and the last year, 2003, for predictive precisions.

Bibliography

Acharya, A., Ghosh, J., and Zhou, M. (2015), "Nonparametric Bayesian factor analysis for dynamic count matrices," in *Artificial Intelligence and Statistics Conference*.

Andrieu, C. and Thoms, J. (2008), "A tutorial on adaptive MCMC," *Statistics and Computing*, 18.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016), "Deep learning for computational biology," *Molecular systems biology*, 12, 878.

Arnol'd, V. I. (2013), *Mathematical methods of classical mechanics*, vol. 60, Springer Science & Business Media.

Bache, K. and Lichman, M. (2013), "UCI machine learning repository," .

Betancourt, M., Byrne, S., and Girolami, M. (2014), "Optimizing the integrator step size for Hamiltonian Monte Carlo," *ArXiv*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003), "Latent Dirichlet Allocation," *JMLR*, 3.

Bottou, L. (2010), "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012), "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *ICML*.

Bris, C. L. and Lions, P.-L. (2008), "Existence and uniqueness of solutions to Fokker–Planck type equations with irregular coefficients," *Communications in Partial Differential Equations*, 33, 1272–1317.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011), *Handbook of Markov Chain Monte Carlo*, CRC press.

Brunick, G., Shreve, S., et al. (2013), "Mimicking an Itô process by a solution of a stochastic differential equation," *The Annals of Applied Probability*, 23, 1584–1628.

Bubeck, S., Eldan, R., and Lehec, J. (2015), "Finite-time analysis of projected Langevin Monte Carlo," in *NIPS*.

Cances, E., Legoll, F., and Stoltz, G. (2007), "Theoretical and numerical comparison of some sampling methods for molecular dynamics," *ESAIM: Mathematical Modelling and Numerical Analysis*, 41.

Chao, W.-L., Solomon, J., Michels, D., and Sha, F. (2015), "Exponential Integration for Hamiltonian Monte Carlo," in *ICML*.

Chen, C., Rao, V., Buntine, W., and Whye Teh, Y. (2013), "Dependent normalized random measures," in *ICML*.

Chen, C., Ding, N., and Carin, L. (2015), "On the convergence of stochastic gradient MCMC algorithms with high-order integrators," in *NIPS*, pp. 2278–2286.

Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. (2016a), "Bridging the gap between stochastic gradient mcmc and stochastic optimization," in *AISTATS*.

Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. (2016b), "Bridging the gap between stochastic gradient MCMC and stochastic optimization," in *Artificial Intelligence and Statistics*, pp. 1051–1060.

Chen, C., Li, C., Chen, L., Wang, W., Pu, Y., and Carin, L. (2018a), "Continuous-Time Flows for Deep Generative Models," in *arXiv*.

Chen, L., Dai, S., Pu, Y., Li, C., and Carin, Q. S. L. (2018b), "Symmetric Variational Autoencoder and Connections to Adversarial Learning," in *AISTATS*.

Chen, T., Fox, E. B., and Guestrin, C. (2014), "Stochastic Gradient Hamiltonian Monte Carlo," *ArXiv*.

Collett, D. (2002), *Modelling binary data*, CRC Press.

Dalalyan, A. S. (2016), "Theoretical guarantees for approximate sampling from smooth and log-concave densities," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

David, L. A., Materna, A. C., Friedman, J., Campos-Baptista, M. I., Blackburn, M. C., Perrotta, A., Erdman, S. E., and Alm, E. J. (2014), "Host lifestyle affects human microbiota on daily timescales," *Genome Biology*, 15.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014a), "Bayesian sampling using stochastic gradient thermostats," in *NIPS*.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014b), "Bayesian sampling using stochastic gradient thermostats," in *Neural Information Processing Systems*.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987), "Hybrid Monte Carlo," *Physics letters B*, 195.

Ekeland, I. and Lasry, J.-M. (1980), "On the number of periodic trajectories for a Hamiltonian flow on a convex energy surface," *Annals of Mathematics*.

Escobar, M. D. and West, M. (1995), "Bayesian density estimation and inference using mixtures," *JASA*, 90, 577–588.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "A note on the group lasso and a sparse group lasso," *arXiv preprint arXiv:1001.0736*.

Gan, C., Gan, Z., He, X., Gao, J., and Deng, L. (2017a), "StyleNet: Generating Attractive Visual Captions With Styles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3137–3146.

Gan, Z., Li, C., Henao, R., Carlson, D. E., and Carin, L. (2015a), "Deep temporal Sigmoid belief networks for sequence modeling," in *Neural Information Processing Systems*.

Gan, Z., Li, C., Henao, R., Carlson, D. E., and Carin, L. (2015b), "Deep temporal sigmoid belief networks for sequence modeling," in *Advances in Neural Information Processing Systems*, pp. 2467–2475.

Gan, Z., Henao, R., Carlson, D., and Carin, L. (2015c), "Learning Deep Sigmoid Belief Networks with Data Augmentation," in *Artificial Intelligence and Statistics Conference*.

Gan, Z., Chen, C., Henao, R., Carlson, D., and Carin, L. (2015d), "Scalable Deep Poisson Factor Analysis for Topic Modeling," in *International Conference of Machine Learning*.

Gan, Z., Chen, C., Henao, R., Carlson, D., and Carin, L. (2015e), "Scalable Deep Poisson Factor Analysis for Topic Modeling," in *International Conference of Machine Learning*.

Gan, Z., Pu, Y., Henao, R., Li, C., He, X., and Carin, L. (2017b), "Learning generic sentence representations using convolutional neural networks," in *EMNLP*.

Gan, Z., Li, C., Chen, C., Pu, Y., Su, Q., and Carin, L. (2017c), "Scalable bayesian learning of recurrent neural networks for language modeling," in *ACL*.

Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., Carin, L., and Deng, L. (2017d), "Semantic compositional networks for visual captioning," in *CVPR*.

Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., Carin, L., and Deng, L. (2017e), "Semantic Compositional Networks for Visual Captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5630–5639.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C., and Carin, L. (2017f), "Triangle Generative Adversarial Networks," in *NIPS*.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C., and Carin, L. (2017g), "Triangle generative adversarial networks," in *Advances in Neural Information Processing Systems*, pp. 5253–5262.

Geyer, C. J. (2005), "Markov Chain Monte Carlo Lecture Notes," .

Girolami, M. and Calderhead, B. (2011), "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. (2007), "Euclidean Embedding of Co-occurrence Data," *Journal of Machine Learning Research*, 8, 2265–2295.

Goldstein, H. (1965), *Classical mechanics*, Pearson Education India.

Haario, H., Saksman, E., and Tamminen, J. (2001), "An adaptive Metropolis algorithm," *Bernoulli*.

Han, S., Du, L., Salazar, E., and Carin, L. (2014), "Dynamic Rank Factor Model for Text Streams," in *Neural Information Processing Systems*.

Henao, R., Gan, Z., Lu, J., and Carin, L. (2015a), "Deep Poisson Factor Modeling," in *Neural Information Processing Systems*.

Henao, R., Gan, Z., Lu, J., and Carin, L. (2015b), "Deep Poisson factor modeling," in *Advances in Neural Information Processing Systems*, pp. 2800–2808.

Hermans, M. and Schrauwen, B. (2013), "Training and analysing deep recurrent neural networks," in *Neural Information Processing Systems*.

Hochreiter, S. and Schmidhuber, J. (1997), "Long short-term memory," *Neural computation*.

Hoff, P. D. (2009), *A first course in Bayesian statistical methods*, Springer Science & Business Media.

Homan, M. D. and Gelman, A. (2014), "The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *The Journal of Machine Learning Research*, 15.

Hwang, C.-R., Hwang-Ma, S.-Y., Sheu, S.-J., et al. (2005), "Accelerating diffusions," *The Annals of Applied Probability*.

Hyvärinen, A. and Oja, E. (2000), "Independent component analysis: algorithms and applications," *Neural networks*, 13.

Isaac, R. (1963), "A General Version of Doeblin's Condition," *The Annals of Mathematical Statistics*.

Jiang, C. and Cong, Y. (2015), "A sixth order diagonally implicit symmetric and symplectic Runge-Kutta method for solving Hamiltonian systems," *Journal of Applied Analysis and Computation*, 5.

Johnson, A. A. (2009), "Geometric ergodicity of Gibbs samplers," Ph.D. thesis, university of Minnesota.

Johnson, N. L., Kemp, A. W., and Kotz, S. (2005), *Univariate discrete distributions*, vol. 444, John Wiley & Sons.

Kalman, R. (1963), "Mathematical description of linear dynamical systems," in *J. the Society for Industrial & Applied Mathematics, Series A: Control*.

Kingma, D. and Ba, J. (2014), "ADAM: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

Korattikara, A., Chen, Y., and Welling, M. (2013), "Austerity in MCMC land: Cutting the Metropolis-Hastings budget," *ArXiv*.

Landau, L. and Lifshitz, E. (1976), *Mechanics, 1st edition*, Pergamon Press, Oxford.

Larochelle, H. and Bengio, Y. (2008), "Classification using Discriminative Restricted Boltzmann Machines," in *ICML*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015), "Deep learning," *Nature*, 521, 436–444.

Li, C., Chen, C., Fan, K., and Carin, L. (2016a), "High-order stochastic gradient thermostats for Bayesian learning of deep models," in *AAAI*.

Li, C., Stevens, A., Chen, C., Pu, Y., Gan, Z., and Carin, L. (2016b), "Learning weight uncertainty with stochastic gradient mcmc for shape classification," in *CVPR*.

Li, C., Stevens, A., Chen, C., Pu, Y., Gan, Z., and Carin, L. (2016c), "Learning weight uncertainty with stochastic gradient mcmc for shape classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5666–5675.

Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R., and Carin, L. (2017), "ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching," in *NIPS*.

Livingstone, S., Betancourt, M., Byrne, S., and Girolami, M. (2016), "On the Geometric Ergodicity of Hamiltonian Monte Carlo," *ArXiv*.

Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., and Vollmer, S. J. (2016), "Relativistic Monte Carlo," *arXiv*.

Ma, Y.-A., Chen, T., and Fox, E. (2015), "A complete recipe for stochastic gradient MCMC," in *NIPS*, pp. 2917–2925.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993), "Building a large annotated corpus of English: The Penn Treebank," *Computational linguistics*.

Martens, J. and Sutskever, I. (2011), "Learning recurrent neural networks with Hessian-free optimization," in *International Conference of Machine Learning*.

Mattingly, J. C., Stuart, A. M., and Tretyakov, M. V. (2010), "Construction of numerical time-average and stationary measures via Poisson equations," *SIAM J. NUMER. ANAL.*, 48, 552–577.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of state calculations by fast computing machines," *The journal of chemical physics*, 21.

Mittelman, R., Kuipers, B., Savarese, S., and Lee, H. (2014), "Structured Recurrent Temporal Restricted Boltzmann Machines," in *International Conference of Machine Learning*.

Mnih, A. and Gregor, K. (2014), "Neural variational inference and learning in belief networks," in *International Conference of Machine Learning*.

Mohamed, S., de Freitas, N., et al. (2013), "Adaptive Hamiltonian and Riemann Manifold Monte Carlo Samplers," *Arxiv*.

Murray, I., Adams, R. P., and MacKay, D. J. (2009), "Elliptical slice sampling," *ArXiv*.

Nadarajah, S. (2005), "A generalized normal distribution," *Journal of Applied Statistics*, 32.

Neal, R. (1992), "Connectionist learning of belief networks," in *Artificial intelligence*.

Neal, R. M. (1993), "Probabilistic inference using Markov chain Monte Carlo methods," *Technical Report CRG-TR-93-1*.

Neal, R. M. (2003), "Slice sampling," *Annals of statistics*.

Neal, R. M. (2011), "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, 2.

Nishimura, A. and Dunson, D. (2016a), "Geometrically Tempered Hamiltonian Monte Carlo," *arXiv preprint arXiv:1604.00872*.

Nishimura, A. and Dunson, D. (2016b), "Variable length trajectory compressible hybrid Monte Carlo," *arXiv preprint arXiv:1604.00889*.

Nishimura, A., Dunson, D., and Lu, J. (2017), "Discontinuous Hamiltonian Monte Carlo for sampling discrete parameters," *arXiv preprint arXiv:1705.08510*.

Pakman, A. and Paninski, L. (2013), "Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions," in *NIPS*.

Park, Y. and Kellis, M. (2015), "Deep learning for regulatory genomics," *Nature biotechnology*, 33, 825–826.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013), "On the difficulty of training recurrent neural networks," in *International Conference of Machine Learning*.

Piegorsch, W. W. (1992), "Complementary log regression for generalized linear models," *The American Statistician*, 46, 94–99.

Pu, Y., Yuan, X., and Carin, L. (2015), "Generative Deep Deconvolutional Learning," in *ICLR workshop*.

Pu, Y., Yuan, X., Stevens, A., Li, C., and Carin, L. (2016a), "A Deep Generative Deconvolutional Image Model," in *AISTATS*.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L. (2016b), "Variational Autoencoder for Deep Learning of Images, Labels and Captions," in *NIPS*.

Pu, Y., Gan, Z., Henao, R., L, C., Han, S., and Carin, L. (2017), "VAE learning via stein variational gradient descent," in *NIPS*.

Pu, Y., Min, R., Gan, Z., and Carin, L. (2018), "Adaptive Feature Abstraction for Translating Video to Text," in *AAAI*.

Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., et al. (2010), "A human gut microbial gene catalogue established by metagenomic sequencing," *nature*, 464, 59–65.

Rabiner, L. and Juang, B. (1986), "An introduction to hidden Markov models," in *ASSP Magazine, IEEE*.

Risken, H. (1984), "Fokker-planck equation," in *The Fokker-Planck Equation*, pp. 63–95, Springer.

Robert, C. and Casella, G. (2004), *Monte Carlo statistical methods*, Springer Science & Business Media.

Roberts, G. O. and Rosenthal, J. S. (1998), "Markov-chain Monte Carlo: Some practical implications of theoretical results," *Canadian Journal of Statistics*, 26.

Roberts, G. O. and Tweedie, R. L. (1996), "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*.

Rosenthal, J. S. (1995), "Minorization conditions and convergence rates for Markov chain Monte Carlo," *Journal of the American Statistical Association*, 90.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988), "Learning representations by back-propagating errors," *Cognitive modeling*, 5, 1.

Salimans, T., Kingma, D. P., and Welling, M. (2014), "Markov chain Monte Carlo and variational inference: Bridging the gap," *ArXiv*.

Sohl-Dickstein, J., Mudigonda, M., and DeWeese, M. R. (2014), "Hamiltonian Monte Carlo without detailed balance," *ArXiv*.

Song, J., Gan, Z., and Carin, L. (2016), "Factored temporal sigmoid belief networks for sequence learning," in *International Conference on Machine Learning*, pp. 1272–1281.

Stevens, A., Pu, Y., Sun, Y., Spell, G., and Carin, L. (2017), "Tensor-Dictionary Learning with Deep Kruskal-Factor Analysis," in *AISTATS*.

Striebel, M., Günther, M., Knechtli, F., and Wandelt, M. (2011), "Accuracy of symmetric partitioned Runge-Kutta methods for differential equations on Lie-groups," *arXiv:1112.4336*.

Su, Q., Liao, X., Li, C., Gan, Z., and Carin, L. (2017), "Unsupervised Learning with Truncated Gaussian Graphical Models." in *AAAI*.

Sutskever, I. and Hinton, G. (2007), "Learning multilevel distributed representations for high-dimensional sequences," in *Artificial Intelligence and Statistics Conference*.

Sutskever, I., Hinton, G., and Taylor, G. (2009), "The recurrent temporal restricted Boltzmann machine," in *Neural Information Processing Systems*.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013), "On the importance of initialization and momentum in deep learning," in *International Conference of Machine Learning*.

Taylor, G., Hinton, G., and Roweis, S. (2006), "Modeling human motion using binary latent variables," in *Neural Information Processing Systems*.

Taylor, J. R. (2005), *Classical mechanics*, University Science Books.

Teh, Y. W., Thiéry, A., and Vollmer, S. (2014), "Consistency and fluctuations for stochastic gradient Langevin dynamics," *ArXiv*.

Tieleman, T. and Hinton, G. (2012), "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*.

Tierney, L. and Mira, A. (1999), "Some adaptive Monte Carlo methods for Bayesian inference," *Statistics in Medicine*, 18.

Tuckerman, M. (2010), *Statistical mechanics: theory and molecular simulation*, Oxford University Press.

Van Der Putten, P. and van Someren, M. (2000), "COIL challenge 2000: The insurance company case," *Sentient Machine Research*, 9.

Vigário, R., Jousmäki, V., Hämäläninen, M., Haft, R., and Oja, E. (1998), "Independent component analysis for identification of artifacts in magnetoencephalographic recordings," in *NIPS*.

Vollmer, S. J., Zygalakis, K. C., and Teh, Y. W. (2016), "Exploration of the (non-) asymptotic bias and variance of stochastic gradient Langevin dynamics," *Journal of Machine Learning Research*, 17, 1–48.

Wang, W., Pu, Y., Verma, V., Fan, K., Zhang, Y., Chen, C., and P. Rai, L. C. (2018), "Zero-Shot Learning via Class-Conditioned Deep Generative Mode," in *AAAI*.

Wang, Y. P. W., Henao, R., Chen, L., Gan, Z., Li, C., and Carin, L. (2017), "Adversarial Symmetric Variational Autoencoder," in *NIPS*.

Welling, M. and Teh, Y. W. (2011a), "Bayesian learning via stochastic gradient Langevin dynamics," in *ICML*.

Welling, M. and Teh, Y. W. (2011b), "Bayesian learning via stochastic gradient Langevin dynamics," in *International Conference of Machine Learning*.

Xian, Y., Pu, Y., Gan, Z., Lu, L., and Thompson, A. (2017), "Adaptive DCTNet for audio signal classification," in *ICASSP*.

Yuan, X., Pu, Y., and Carin, L. (2017), "Compressive Sensing via Convolutional Factor Analysis," in *arXiv*.

Yuan, X., Pu, Y., and Carin, L. (2018), "Parallel lensless compressive imaging via deep convolutional neural networks," *Optics Express*.

Zhang, Y., Ghahramani, Z., Storkey, A. J., and Sutton, C. A. (2012), "Continuous relaxations for discrete Hamiltonian Monte Carlo," in *NIPS*.

Zhang, Y., Zhao, Y., David, L., Henao, R., and Carin, L. (2016a), "Dynamic Poisson Factor Analysis," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 1359–1364, IEEE.

Zhang, Y., Gan, Z., and Carin, L. (2016b), "Generating text via adversarial training," in *NIPS workshop on Adversarial Training*.

Zhang, Y., Chen, C., Henao, R., and Carin, L. (2016c), "Laplacian Hamiltonian Monte Carlo," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 98–114, Springer.

Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K., and Carin, L. (2016d), "Towards Unifying Hamiltonian Monte Carlo and Slice Sampling," in *NIPS*.

Zhang, Y., Shen, D., Wang, G., Gan, Z., Henao, R., and Carin, L. (2017a), "Deconvolutional paragraph representation learning," in *Advances in Neural Information Processing Systems*, pp. 4172–4182.

Zhang, Y., Chen, C., Gan, Z., Henao, R., and Carin, L. (2017b), "Stochastic Gradient Monomial Gamma Sampler," in *ICML*.

Zhou, M. (2015), "Infinite Edge Partition Models for Overlapping Community Detection and Link Prediction," in *Artificial Intelligence and Statistics Conference*.

Zhou, M. and Carin, L. (2015), "Negative binomial process count and mixture modeling," *Pattern Analysis and Machine Intelligence*, 37, 307–320.

Zhou, M., Hannah, L., Dunson, D., and Carin, L. (2012), "Beta-negative binomial process and Poisson factor analysis," in *Artificial Intelligence and Statistics Conference*.

Biography

Yizhe Zhang was born in January 31, 1989 in Taiyuan, China. He received a B.S. in Physics from Nanjing University in July 2011, an M.S. in statistic science from Duke University in 2018, and a Ph.D. in computational biology and bioinformatics from Duke University in 2018.

He was the winner of student travel awards for NIPS, ICML and other conferences. His published papers include Zhang et al. (2016d, 2017b, 2016b); Gan et al. (2017g); Zhang et al. (2017a, 2016c) and Zhang et al. (2016a). He graduated with his Ph.D. in computational biology and bioinformatics under the supervision of Professor Lawrence Carin.