Robust Monte Carlo Sampling using Riemannian Nosé-Poincaré Hamiltonian Dynamics

Anirban Roychowdhury ROYCHOWDHURY.7@OSU.EDU
Ohio State University, Columbus, OH 43210

Brian Kulis BKULIS@BU.EDU
Boston University, Boston, MA 02215

Srinivasan Parthasarathy SRINI@CSE.OHIO-STATE.EDU
Ohio State University, Columbus, OH 43210

Abstract

We present a Monte Carlo sampler using a modified Nosé-Poincaré Hamiltonian along with Riemannian preconditioning. Hamiltonian Monte Carlo samplers allow better exploration of the state space as opposed to random walk-based methods, but, from a molecular dynamics perspective, may not necessarily provide samples from the canonical ensemble. Nosé-Hoover samplers rectify that shortcoming, but the resultant dynamics are not Hamiltonian. Furthermore, usage of these algorithms on large real-life datasets necessitates the use of stochastic gradients, which acts as another potentially destabilizing source of noise. In this work, we propose dynamics based on a modified Nosé-Poincaré Hamiltonian augmented with Riemannian manifold corrections. The resultant symplectic sampling algorithm samples from the canonical ensemble while using structural cues from the Riemannian preconditioning matrices to efficiently traverse the parameter space. We also propose a stochastic variant using additional terms in the Hamiltonian to correct for the noise from the stochastic gradients. We show strong performance of our algorithms on synthetic datasets and high-dimensional Poisson factor analysis-based topic modeling scenarios.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

1. Introduction

Bayesian inference in high dimensional models often requires one to draw samples from posterior distributions of variables which cannot be computed in closed form. Monte Carlo techniques are the primary tool in one's arsenal for this purpose; they allow one to draw samples from a sequence of probability distributions that form a Markov chain with the target distribution as its stationary distribution. The Hamiltonian Monte Carlo (HMC) technique, first proposed in (Duane et al., 1987) as "Hybrid Monte Carlo", improves on this by using ideas from statistical physics to avoid the random walk behavior that normally arises in these Markov chains. HMC sets the target distribution as the "potential energy" of the simulated system, and uses auxiliary "momentum" variables to augment the potential with a kinetic energy term. Hamiltonian dynamics are then used to create a sampler that conserves this quantity, and the samples generated by this technique are provably less correlated among themselves which leads to faster convergence to the target distribution. The dynamics are usually specified with a set of differential equations which then have to be discretized; however one can derive discrete-time numeric integrators (Neal, 2011; Leimkuhler & Reich, 2004), usually called "leapfrog" methods, that preserve the detailed balance and time reversibility properties of the continuous-time formulations.

In the statistical physics literature, dynamics-based techniques are used to sample from a canonical ensemble, where the possible states of the system remain at a constant temperature. One technique that has been used for this purpose uses the Nosé Hamiltonian (Nosé, 1984), which generates sequence of states from the canonical ensemble under the standard ergodicity assumptions. The Nosé-Hoover system (Hoover, 1985) proposes a change of variables that allows evenly spaced samples from the canonical ensem-

ble, however the resulting system is not Hamiltonian. In particular, the numeric integrator used to solve the system of differential equations for this formulation is not symplectic, in that it does not preserve the symplectic geometry of the original manifold defined by the Hamiltonian system. Symplecticness being in general a stronger property than volume preservation, (Bond et al., 1999) proposed a Hamiltonian which can be used to derive a numeric integrator that samples from a fixed temperature canonical ensemble, and is also symplectic and time-reversible.

Although Monte Carlo techniques constructed from these formulations can be used to sample from Markov chains that converge to the desired target distributions, a difficulty arises when working with very large-scale datasets. Here, due to computational limitations, one often cannot compute the gradient of the target likelihood (potential energy) over the entire dataset in a reasonable amount of time. Instead, at every iteration one computes a stochastic gradient (SG), which is the gradient evaluated over a randomly selected "mini-batch" of data (Robbins & Monro, 1951; Welling & Teh, 2011). This allows the algorithms to scale to massive datasets commonly seen in machine learning, while preserving the desired theoretical properties. Recent work in this vein includes SG Langevin dynamics (Welling & Teh, 2011), SG Hamiltonian Monte Carlo (Chen et al., 2014), and other variants and extensions (Ding et al., 2014; Patterson & Teh, 2013). First order Langevin dynamics methods are inherently random-walk based, whereas the HMC methods exploit the exploration efficiencies of Hamiltonian systems to derive more robust samplers in a stochastic setting. However, the samples are not automatically drawn from the canonical ensemble, an issue that was addressed in (Ding et al., 2014), where the SGNHT algorithm was proposed. The authors do show the efficacy of the sampler in the face of stochastic noise; but as mentioned above, the Nosé-Hoover system is not Hamiltonian, which can have adverse effects on the efficiency and convergence speed of any MCMC algorithm derived using its dynamics.

Therefore one would want a stochastic technique that samples from the canonical ensemble, without sacrificing the advantages of Hamiltonian trajectories. To that end, we propose the stochastic gradient Nosé-Poincaré Hamiltonian Monte Carlo sampler, which uses a variant of the Hamiltonian proposed in (Bond et al., 1999) that leverages Riemannian preconditioning and corrects for the random noise from the stochastic gradients while preserving the desired properties mentioned above. The basic idea of Riemannian adaptations, first proposed in (Girolami & Calderhead, 2011), is to define a Riemannian metric tensor on the parameter space and use structural cues from the resulting manifold while traversing the Hamiltonian trajectories. This technique was exploited in the context of sampling from a high dimensional probability simplex (Patterson & Teh, 2013), where the geometric information allows the sampler to improve upon the slow mixing behavior exhibited by the first order Langevin dynamics on parameter spaces with a high degree of correlation. Another advantage of locally-adaptive preconditioning is that one does not have to worry about selecting optimal values for the "mass" matrices in Hamiltonian samplers, an aspect such samplers tend to be highly sensitive to (Girolami & Calderhead, 2011; Bond et al., 1999). In our algorithm, we use Riemann tensors on the original parameter space to precondition the momenta of both the real and extended position variables in the Nosé Hamiltonian, and show that the resulting system samples from the canonical ensemble. We then add correction terms to account for noise when the full gradient is replaced by the stochastic gradient, and use the Fokker-Planck equation to prove that the dynamics conserve the desired energy. Finally, we apply our algorithm to parameter estimation in synthetic settings and a high dimensional topic modeling scenario using the Poisson factor analysis framework (Zhou & Carin, 2015) and the exact Gamma process construction of (Roychowdhury & Kulis, 2014).

2. Preliminaries

2.1. Monte Carlo using Hamiltonian Dynamics

Let us denote the model parameters by $\boldsymbol{\theta}$. Suppose we want to generate samples from the posterior distribution of $\boldsymbol{\theta}$ given data $\mathbf{X}$, $p(\boldsymbol{\theta}|\mathbf{X})$. In a Hamiltonian setting, we take the joint log likelihood of the data and the parameters, $\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$, and add to it a term involving auxiliary "momentum" variables $\mathbf{p}$, to get the Hamiltonian

$$H(\boldsymbol{\theta}, \mathbf{p}) = -\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T \mathrm{M}^{-1}\mathbf{p}. \tag{1}$$

The quantity can be interpreted in a physical sense as the sum of the potential energy $\mathcal{L}(\boldsymbol{\theta})$ and the kinetic energy $\frac{1}{2}\mathbf{p}^T\mathrm{M}^{-1}\mathbf{p}$, where M^{-1} acts as the canonical mass matrix. The joint distribution of $\boldsymbol{\theta}$ and $\mathbf{p}$ is then defined as $p(\boldsymbol{\theta}, \mathbf{p}) \propto \exp\left(-H(\boldsymbol{\theta}, \mathbf{p})\right)$. It is easily seen that we can integrate out $\mathbf{p}$ from $p(\boldsymbol{\theta}, \mathbf{p})$ to get the desired posterior distribution on $\boldsymbol{\theta}$.

Denoting time derivatives with the dot accent, i.e. $\dot{\boldsymbol{\theta}} = d\boldsymbol{\theta}/dt$, the Hamiltonian equations of motion governing the dynamics of this system can be written as $\dot{\boldsymbol{\theta}} = \frac{\partial}{\partial \mathbf{p}} H(\boldsymbol{\theta}, \mathbf{p})$, $\dot{\mathbf{p}} = -\frac{\partial}{\partial \boldsymbol{\theta}} H(\boldsymbol{\theta}, \mathbf{p})$. In our formulation the dynamics are $\dot{\boldsymbol{\theta}} = \mathrm{M}^{-1}\mathbf{p}$, $\dot{\mathbf{p}} = \nabla\mathcal{L}(\boldsymbol{\theta})$. These equations are time-reversible, and the dynamics conserve the total energy and are symplectic as well. These continuous-time equations are discretized to give "leapfrog" algorithms which are used for Monte Carlo simulations along with Metropolis-Hastings correction steps. For details see (Neal, 2011; Leimkuhler & Reich, 2004).

2.2. Riemann Adjusted Hamiltonian Monte Carlo

The scaling issues associated with standard Hamiltonian Monte Carlo algorithms can be partially alleviated using Riemannian preconditioning. We first define the Hamiltonian on a Riemann manifold defined by a positive definite metric $\mathbf{G}(\boldsymbol{\theta})$. Trajectories incorporating information from this manifold can be simulated by simply defining the kinetic energy in terms of the metric tensor (Girolami & Calderhead, 2011), which leads to the following Hamiltonian:

$$H_{gc}(\boldsymbol{\theta}, \mathbf{p}) = -\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p} + \frac{1}{2}\log\left\{(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|\right\} \tag{2}$$

where D is the dimensionality of the parameter space. The log term ensures that the momentum variable $\mathbf{p}$ can be integrated out to recover the desired marginal density of $\boldsymbol{\theta}$. The equations of motion in this system therefore are

$$\begin{aligned}\dot{\boldsymbol{\theta}} &= \mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p} \\ \dot{\mathbf{p}} &= \nabla\mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2}\mathrm{tr}\left(\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\right) \\ &\quad + \frac{1}{2}\mathbf{p}^T\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p}.\end{aligned}$$

To discretize this system of equations, we use the generalized leapfrog algorithm, where a first order symplectic integrator is composed with its adjoint; the resultant second order integrator can be shown to be both time-reversible and symplectic (Leimkuhler & Reich, 2004). We use this integrator to derive discretized samplers from our Nosé-Poincaré Hamiltonians in §3.

2.3. Stochastic Gradient Dynamics

For moderate datasets, the gradient of the log-likelihood in the dynamics above can be evaluated over the entire dataset. However, for large datasets, doing so in every iteration becomes prohibitively expensive. The most common way around this problem is to replace the full gradient by one evaluated over a random "mini-batch" of the dataset, a technique inspired by (Robbins & Monro, 1951). The approximate gradient of the log-likelihood can be written as

$$\nabla\tilde{\mathcal{L}}\left(\boldsymbol{\theta}\right) = \frac{|N|}{|\tilde{N}|}\sum_{x\in\tilde{N}}\nabla\log p(x|\boldsymbol{\theta}) + \nabla\log p(\boldsymbol{\theta}),$$

where N denotes the entire dataset, and $\tilde{N}$ denotes a random mini-batch. Monte Carlo samplers using stochastic gradients have been proposed for first order Langevin dynamics (Welling & Teh, 2011; Patterson & Teh, 2013), as well as for Hamiltonian systems using second order Langevin dynamics (Chen et al., 2014).

Another feature of these algorithms is the removal of a Metropolis-Hastings correction step, as that would require very expensive computations over the entire dataset. Instead, a decaying sequence of stepsizes $\{\epsilon_t\}$ satisfying $\sum_t \epsilon_t = \infty$ and $\sum_t \epsilon_t^2 < \infty$ is used, for which the Markov chain of distributions can be proved to have the desired target as its equilibrium distribution.

3. Riemannian Nosé-Poincaré Dynamics

3.1. The Deterministic Case

The Nosé-Poincaré Hamiltonian proposed in (Bond et al., 1999) can be written as

$$H(\boldsymbol{\theta}, \mathbf{p}, s, q) = s\Bigg(-\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\left(\frac{\mathbf{p}}{s}\right)^T\mathbf{M}^{-1}\frac{\mathbf{p}}{s} + \frac{q^2}{2Q} + gkT\log s - H_0\Bigg). \tag{3}$$

Here k is Boltzmann's constant, T is the system temperature, g is equal to the number of degrees of freedom of the system. s is an extended position variable with momentum q and associated mass Q, as introduced by (Nosé, 1984). s acts as the time-scaling function in a Poincaré transformation, which allows us to preserve the dynamics of the original Hamiltonian upto the time transformation.

3.1.1. The Riemann augmentation

As mentioned previously, the dynamics of Hamiltonian systems are highly sensitive to the values of the mass matrices, in this case M and Q. To take advantage of locally-adaptive walks on the Riemann manifold, we replace the mass matrices in (3) by a metric tensor $\mathbf{G}(\boldsymbol{\theta})$ as follows:

$$H(\boldsymbol{\theta}, \mathbf{p}, s, q) = s\Bigg(-\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\left(\frac{\mathbf{p}}{s}\right)^T\mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\mathbf{p}}{s} + \frac{1}{2}|\mathbf{G}(\boldsymbol{\theta})|^{-1}q^2 + \frac{1+kT}{2}\log\left\{(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|\right\} + gkT\log s - H_0\Bigg). \tag{4}$$

The log term ensures that we can integrate out the extended momentum term q to get back the Hamiltonian (2), or for that matter the one in (1), if one treats the curvature of the metric-defined manifold as a constant. Note that the only constant one needs to choose in this system is the value for T. In Bayesian inference we usually take $kT = 1$, though use of this Hamiltonian is not restricted to that specific choice.

Theorem 1. *The dynamical system derived from the Riemannian Nosé-Poincaré Hamiltonian (4) generates samples from the canonical ensemble.*

Proof. We will show that we can integrate out s, q from $p(\boldsymbol{\theta}, \mathbf{p}, s, q) \propto \exp\left(-H(\boldsymbol{\theta}, \mathbf{p}, s, q)\right)$ to get $p(\boldsymbol{\theta}, \mathbf{p}) \propto \exp\left(-H_{gc}(\boldsymbol{\theta}, \mathbf{p})/kT\right)$. The integration of s essentially follows (Bond et al., 1999), so we detail that in §A of the supplementary. With s integrated out, we are left with $p(\boldsymbol{\theta}, \mathbf{p}, q) \propto \exp\left(-H_{gc}(\boldsymbol{\theta}, \mathbf{p})/kT\right) \exp\Big[-\frac{1}{2kT}|\mathbf{G}(\boldsymbol{\theta})|^{-1}q^2 - \frac{1}{2}\log\left\{(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|\right\}\Big]$. We can easily integrate out q from the second exponential term on the right to get the desired form for $p(\boldsymbol{\theta}, \mathbf{p})$. □

The dynamics for this system are given by

$$\begin{aligned}
\dot{\boldsymbol{\theta}} &= \left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1} \\
\dot{\mathbf{p}} &= s\frac{1}{2}\left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\mathbf{G}(\boldsymbol{\theta})^{-1}\left(\frac{\mathbf{p}}{s}\right) \\
&\quad + s\frac{1}{2}q^2\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\mathbf{G}(\boldsymbol{\theta})^{-1} \\
&\quad - \frac{s}{2}(1+kT)\text{tr}\left(\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\right) + s\nabla\mathcal{L}(\boldsymbol{\theta}) \\
\dot{s} &= sq\mathbf{G}(\boldsymbol{\theta})^{-1} \\
\dot{q} &= -gkT + \left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\mathbf{p}}{s} - H_{\text{inner}}
\end{aligned} \tag{5}$$

where $H_{\text{inner}} = \Big(-\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\mathbf{p}}{s}$ $+ \frac{1}{2}|\mathbf{G}(\boldsymbol{\theta})|^{-1}q^2 + \frac{1+kT}{2}\log\left\{(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|\right\}$ $+ gkT\log s - H_0\Big)$.

Since we derived these equations following Hamiltonian's laws of motion, the dynamics are time-reversible and symplectic. Moreover, as shown above, the samples are drawn from the fixed temperature canonical ensemble, and the use of Riemann metric tensors allows us to exploit the geometry of the manifold. We can see that if the manifold is assumed to have a constant curvature. i.e. $\nabla\mathbf{G}(\boldsymbol{\theta}) = 0$, then the dynamics above reduce to those of the standard Nosé-Poincaré system in (Bond et al., 1999).

We can also see that the Hamiltonian (4) is not a simple generalization of (2); the only way to recover (2) is to set the extended position variable $s = 1$, $q = 0$ (effectively assuming that the particles are not changing position in the extended phase space), as well as setting $T = 0$. This is problematic in a physical sense, since particles do not move at absolute zero.

3.1.2. The Discretized Dynamics

For Monte Carlo sampling we need to discretize this system, and to do so we use the generalized leapfrog algorithm of (Leimkuhler & Reich, 2004). The generalized leapfrog algorithm can be shown to be both time-reversible and symplectic. However, since our Hamiltonian is not separable due to the coupling of the momenta terms with the position variable $\boldsymbol{\theta}$, the leapfrog equations are implicitly defined, necessitating the use of fixed point techniques or Newton-like iterations to solve them. The algorithm applied to the dynamics (5) yields the following discrete update equations:

$$\begin{aligned}
p_i^{(t+\frac{\epsilon}{2})} &= p_i^{(t)} - \frac{\epsilon}{2}\nabla_{\theta_i}H\left(\boldsymbol{\theta}^{(t)}, s^{(t)}, \mathbf{p}^{(t+\epsilon/2)}, q^{(t+\epsilon/2)}\right) \\
q^{(t+\frac{\epsilon}{2})} &= q^{(t)} - \frac{\epsilon}{2}\Bigg[H_{\text{inner}} + gkT \\
&\quad - \left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t)}}\right)^T G(\boldsymbol{\theta}^{(t)})^{-1}\left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t)}}\right)\Bigg] \\
\theta_i^{(t+\epsilon)} &= \theta_i^{(t)} + \frac{\epsilon}{2}\Bigg[\left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t)}}\right)^T G(\boldsymbol{\theta}^{(t)})^{-1} \\
&\quad + \left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t+\epsilon)}}\right)^T G(\boldsymbol{\theta}^{(t+\epsilon)})^{-1}\Bigg]_i \\
s^{(t+\epsilon)} &= s^{(t)} + \frac{\epsilon}{2}\Bigg[s^{(t)}q^{(t+\epsilon/2)}|G(\boldsymbol{\theta}^{(t)})|^{-1} \\
&\quad + s^{(t+\epsilon)}q^{(t+\epsilon/2)}|G(\boldsymbol{\theta}^{(t+\epsilon)})|^{-1}\Bigg] \\
p_i^{(t+\epsilon)} &= p_i^{(t+\epsilon/2)} - \frac{\epsilon}{2}\nabla_{\theta_i}H\left(\boldsymbol{\theta}^{(t+\epsilon)}, s^{(t+\epsilon)}, \mathbf{p}^{(t+\frac{\epsilon}{2})}, q^{(t+\frac{\epsilon}{2})}\right) \\
q^{(t+\epsilon)} &= q^{(t+\epsilon/2)} - \frac{\epsilon}{2}\Bigg[H_{\text{inner}} + gkT \\
&\quad - \left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t+\epsilon)}}\right)^T G(\boldsymbol{\theta}^{(t+\epsilon)})^{-1}\left(\frac{\mathbf{p}^{(t+\epsilon/2)}}{s^{(t+\epsilon)}}\right)\Bigg]
\end{aligned} \tag{6}$$

where $\nabla_{\theta_i}H(\boldsymbol{\theta}, s, \mathbf{p}, q)$

$$\begin{aligned}
&= \frac{\epsilon}{2}\Bigg[-\frac{1}{2}s\left(\frac{\mathbf{p}}{s}\right)^T G(\boldsymbol{\theta})^{-1}\left(\frac{\partial}{\partial\theta_i}G(\boldsymbol{\theta})\right)G(\boldsymbol{\theta})^{-1}\left(\frac{\mathbf{p}}{s}\right) \\
&+ \frac{s}{2}(1+kT)\text{tr}\left\{G(\boldsymbol{\theta})^{-1}\frac{\partial}{\partial\theta_i}G(\boldsymbol{\theta})\right\} - s\frac{\partial}{\partial\theta_i}\mathcal{L}(\boldsymbol{\theta}) \\
&- s\frac{q^2}{2}|G(\boldsymbol{\theta})|^{-1}\text{tr}\left\{G(\boldsymbol{\theta})^{-1}\frac{\partial}{\partial\theta_i}G(\boldsymbol{\theta})\right\}\Bigg].
\end{aligned}$$

The half-step $(t + \epsilon/2)$ updates equations for the momenta $\mathbf{p}$ and q are implicitly defined, as are those for the $\boldsymbol{\theta}, s$ pair. The full step $(t + \epsilon)$ updates for $\mathbf{p}$ and q are explicit, since they depend only on the half-step values of $\mathbf{p}$ and q and the full step ones for $\boldsymbol{\theta}, s$. The overall procedure is outlined in Algorithm 1.

In (Girolami & Calderhead, 2011) the authors mention using fixed point iterations for solving a similar set of equations. However, in our experiments with real datasets, fixed point updates led to unstable mixing at even moderate learning rates. One reason for this could be the fact that the Jacobians of the implicit equations are large (implying "stiff" domains) for these datasets

Algorithm 1 Riemann Nosé-Poincaré HMC

Input: $\boldsymbol{\theta}, \epsilon, kT$
Initialize $\boldsymbol{\theta}, s$
repeat
· Sample $\mathbf{p}^{(t)} \sim N(0, G(\boldsymbol{\theta})), q^{(t)} \sim N(0, |G(\boldsymbol{\theta})|)$
·Perform leapfrog dynamics (6)
to get $((\boldsymbol{\theta}^{(t)}, s^{(t)}, \mathbf{p}^{(t)}, q^{(t)})$:
for $i = 1$ **to** $leapfrog\_iterations$ **do**
· Perform implicit Newton updates to get $\mathbf{p}^{(t+\epsilon/2)}, q^{(t+\epsilon/2)}, \boldsymbol{\theta}^{(t+\epsilon)}, q^{(t+\epsilon)}$
· Perform explicit updates to get $\mathbf{p}^{(t+\epsilon)}, q^{(t+\epsilon)}$
end for
$\cdot(\boldsymbol{\theta}^{'}, s^{'}, \mathbf{p}^{'}, q^{'}) \leftarrow ((\boldsymbol{\theta}^{(t+\epsilon)}, s^{(t+\epsilon)}, \mathbf{p}^{(t+\epsilon)}, q^{(t+\epsilon)})$
· Set $((\boldsymbol{\theta}^{(t+1)}, s^{(t+1)}, \mathbf{p}^{(t+1)}, q^{(t+1)})$
using Metropolis-Hastings
until forever

and this specific formulation. Therefore, for strictly positive parameters we resort to using diagonal metric tensors $\big(G(\boldsymbol{\theta}) = \text{diag}(\boldsymbol{\theta})^{-1}$, assuming $G(\boldsymbol{\theta}) \succ 0\big)$, and using Newton's method for solving the implicit systems. We provide additional details in §B.1 of the supplementary.

3.2. The Stochastic Case

Typically when working with large datasets, computing the gradient of the log-likelihood over the entire dataset is very expensive. Therefore we resort to evaluating the gradients on a mini-batch of the data. The stochastic gradient of the log-likelihood can be written as

$$\nabla \tilde{\mathcal{L}}\left(\boldsymbol{\theta}\right) = \frac{|N|}{|\tilde{N}|} \sum_{x \in \tilde{N}} \nabla \log p(x|\boldsymbol{\theta}) + \nabla \log p(\boldsymbol{\theta}),$$

where N denotes the entire dataset, and $\tilde{N}$ denotes a random mini-batch.

From the dynamics in the deterministic case (5), we can see that there are two sources of stochastic noise in the minibatch setting from the two momenta terms: one in the equation for $\dot{\mathbf{p}}$, where we have the extended position variable s multiplied by the gradient of the log-likelihood, and the second in the equation for $\dot{q}$, where we have the full log-likelihood term in H_{inner}. Note that the additive noise in the update for q is purely a function of $\boldsymbol{\theta}$, whereas the noise arising from the stochastic gradient in $\dot{\mathbf{p}}$ is multiplied by the extended position variable s. Therefore, following convention, if we write the stochastic terms (likelihood as well as gradient) as the corresponding full terms plus random noise, then we have the following expressions for the momenta dynamics in the stochastic setting:

$$\tilde{\dot{q}} = \dot{q} + N(0, 2A(\boldsymbol{\theta}))$$
$$\tilde{\dot{\mathbf{p}}} = \dot{\mathbf{p}} + N(0, 2\sqrt{s}B(\boldsymbol{\theta})).$$

This therefore turns the deterministic dynamics of (5) into a Langevin diffusion, with $A(\boldsymbol{\theta})$ and $\sqrt{s}B(\boldsymbol{\theta})$ acting as diffusion coefficients of standard Wiener processes.

As one might imagine, using these noisy terms in the dynamics without any correction leads to non-conservation of the total system energy; indeed, (Chen et al., 2014) showed that under certain conditions the entropy of such a system would strictly increase with time. Therefore, one needs to introduce additional terms in the dynamics if the Hamiltonian (4) is to be conserved in the stochastic setting. To do so, we turn to the Fokker-Planck equation.

THE FOKKER-PLANCK CORRECTIONS

The Fokker-Planck equation describes the evolution of the probability distribution of the parameters of a differential equation under stochastic forces. For a stochastic differential equation with diffusion coefficient $D(\boldsymbol{\theta})$, written as $\dot{\boldsymbol{\theta}} = f(\boldsymbol{\theta}) + N(0, 2D(\boldsymbol{\theta}))$, with the distribution of $\boldsymbol{\theta}$ being $p(\boldsymbol{\theta})$, the Fokker-Planck equation can be written as

$$\frac{\partial}{\partial t}\boldsymbol{p}(\boldsymbol{\theta}) = -\frac{\partial}{\partial \boldsymbol{\theta}}\left[f(\boldsymbol{\theta})p(\boldsymbol{\theta})\right] + \frac{\partial^2}{\partial \boldsymbol{\theta}^2}\left[D(\boldsymbol{\theta})p(\boldsymbol{\theta})\right], \quad (7)$$

where the notation $\frac{\partial^2}{\partial \boldsymbol{\theta}^2}$ denotes $\sum_{i,j} \frac{\partial}{\partial \theta_i}\frac{\partial}{\partial \theta_j}$.

Using this equation, we can derive the corrective terms for the stochastic noise in the dynamics (5). We propose correction terms consisting of the Hamiltonian equations for the position variables with suitable multiplicative terms to cancel out the diffusion noise, resulting in the following corrected dynamics:

$$\begin{aligned}
\dot{\boldsymbol{\theta}} &= \left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1} \\
\dot{\mathbf{p}} &= s\frac{1}{2}\left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\mathbf{G}(\boldsymbol{\theta})^{-1}\left(\frac{\mathbf{p}}{s}\right) \\
&\quad + s\frac{1}{2}q^2\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\mathbf{G}(\boldsymbol{\theta})^{-1} - \sqrt{s}B(\boldsymbol{\theta})\left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1} \\
&\quad - \frac{s}{2}(1+kT)\text{tr}\left(\mathbf{G}(\boldsymbol{\theta})^{-1}\nabla\mathbf{G}(\boldsymbol{\theta})\right) + s\nabla\tilde{\mathcal{L}}(\boldsymbol{\theta}) \\
\dot{s} &= sq\mathbf{G}(\boldsymbol{\theta})^{-1} \\
\dot{q} &= -gkT + \left(\frac{\mathbf{p}}{s}\right)^T \mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\mathbf{p}}{s} - \tilde{H}_{\text{inner}} - A(\boldsymbol{\theta})sq\mathbf{G}(\boldsymbol{\theta})^{-1}.
\end{aligned} \quad (8)$$

This choice can be justified using the Fokker-Planck equation, as we prove below.

Theorem 2. *The dynamics defined in* (8) *leave the probability distribution defined by* $p(\boldsymbol{\theta}, \boldsymbol{p}, s, q) \propto \exp\left(-H(\boldsymbol{\theta}, \boldsymbol{p}, s, q)\right)$ *invariant.*

Proof. Let us start with the *deterministic* Hamiltonian dynamics (5), and replace the log-likelihood terms therein

with their stochastic versions, without any corrections. Following the notation of (Yin & Ao, 2006), the dynamics can be represented in the following format:

$$\begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{p}} \\ \dot{s} \\ \dot{q} \end{bmatrix} = - \begin{bmatrix} 0 & 0 & 0 & -I \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{\theta}} H(\boldsymbol{\theta}, \mathbf{p}, s, q) \\ \frac{\partial}{\partial q} H(\boldsymbol{\theta}, \mathbf{p}, s, q) \\ \frac{\partial}{\partial \boldsymbol{\theta}} H(\boldsymbol{\theta}, \mathbf{p}, s, q) \\ \frac{\partial}{\partial \mathbf{p}} H(\boldsymbol{\theta}, \mathbf{p}, s, q) \end{bmatrix} + \boldsymbol{N} \tag{9}$$

where $\boldsymbol{N} = [0, N(0, 2\sqrt{s}B(\boldsymbol{\theta})), 0, N(0, 2A(\boldsymbol{\theta}))]^T$. Let us denote the anti-symmetric matrix above by X. Then, denoting $\nabla = [\partial/\partial\boldsymbol{\theta}; \partial/\partial\mathbf{p}; \partial/\partial s; \partial/\partial q]$, it is easy to see that $\text{tr}\left\{\nabla^T \nabla X y\right\} = 0$ for any $y(\boldsymbol{\theta}, \mathbf{p}, s, q)$.

Therefore the right hand side of the Fokker-Planck equation (7) can be written as

$$\begin{aligned} &- \text{tr}\nabla^T \left\{p(\boldsymbol{\theta}, \mathbf{p}, s, q) X \nabla H\right\} + \text{tr}\left\{\nabla^T D \nabla p(\boldsymbol{\theta}, \mathbf{p}, s, q)\right\} \\ &= -\text{tr}\nabla^T \left\{p(\boldsymbol{\theta}, \mathbf{p}, s, q) X \nabla H\right\} \\ &+ \text{tr}\left\{(D + X)\nabla^T \nabla p(\boldsymbol{\theta}, \mathbf{p}, s, q)\right\}. \end{aligned}$$

Here we have used the shorthand ∇H to refer to the second matrix on the right hand side of equation (9), and D contains the diffusion terms from the stochastic noise (see §C of the supplementary for the exact formulation).

Note that $\nabla p = -p\nabla H$, since $p \propto \exp(-H)$. Therefore, if we simply replace X with $D + X$ in (9), the right hand side of the Fokker-Plank equation reduces to zero. Using $D + X$ in (9) is equivalent to the dynamics (8). □

As mentioned before, we use the generalized leapfrog algorithm to discretize the continuous differential equations of motion. The generalized leapfrog algorithm is a composition of a symplectic first-order Euler integrator with its adjoint. We describe the discretized version of the dynamics (8) in §B of the supplementary.

4. Experiments

4.1. Estimation of 1D Gaussian Distribution

We start off with a synthetic experiment on learning the parameters of a one-dimensional Gaussian distribution. We generate 5000 points from a standard normal distribution, and attempt to learn the mean and the variance using the discretized stochastic algorithm based on the dynamics (8). We call this algorithm stochastic gradient Riemann Nosé-Poincaré Hamiltonian Monte Carlo (SGR-NPHMC). We compare it to the SG-NHT algorithm of (Ding et al., 2014). Extensive comparisons of SG-NHT with related techniques like stochastic gradient Hamiltonian Monte Carlo and Langevin Dynamics have already been performed in the literature, hence we do not conduct comparisons with those methods here.

Table 1. RMSE and auto-correlation times of the sampled means, precisions from SGR-NPHMC runs on synthetic Gaussian data.

{A,B}	RMSE (μ)	RMSE (τ)	A.T. (μ)
0.01	0.0240	0.0328	14.8999
0.001	0.0244	0.0466	13.6332
0.0001	0.0289	0.0433	2.5899

For both SGR-NPHMC and SG-NHT, we use normal-Wishart priors on the mean and precision; the posterior distribution is proportional to $p(\mu, \tau|\mathbf{X}) \propto N(\mathbf{X}|\mu, \tau)\mathcal{W}(\tau|1, 1)$, where τ denotes the precision, and $\mathcal{W}$ denotes the Wishart distribution. We run both algorithms for 10^5 iterations and discard the first 5000 "burn-in" iterations. For our Riemannian algorithm we use the observed Fisher information plus the negative Hessian of the prior as the metric tensor, and perform one fixed point iteration to solve the implicit system of equations. For both algorithms we use 10 leapfrog iterations. Learning rates are fixed to 1e-3 and batchsizes to 100 for both algorithms.

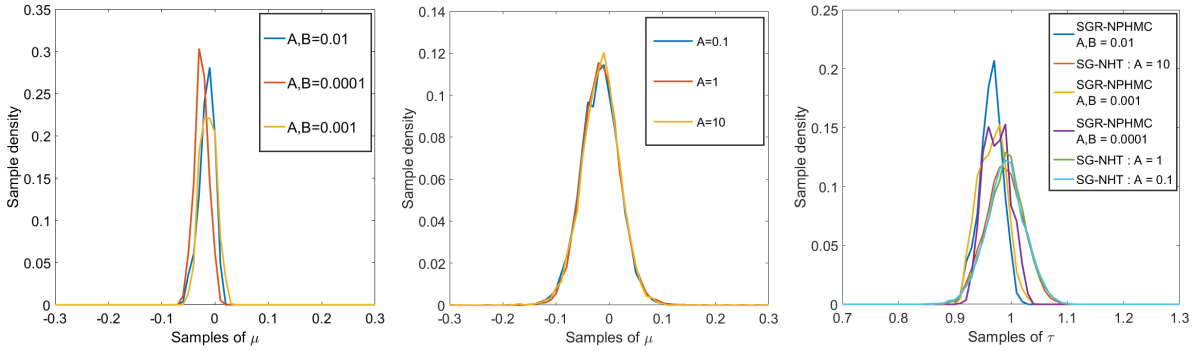
In Figure 1 we demonstrate the sensitivity of the algorithms to different values of the stochastic noise correction terms. The post-burnin samples of μ generated by both algorithms for various values of these terms are plotted in Figures 1a and 1b. 1c. Figure 1c shows the corresponding precision samples. We can see that SGR-NPHMC is as robust to stochastic noise as SG-NHT, and both algorithms generate acceptable samples of μ and τ post-burnin.

However the sampling trajectories in Figures 2a and 2b tell a different story. We see that SG-NHT overshoots the target value of μ by a large margin, as well as having a higher spread of the post-burnin samples. In contrast, SGR-NPHMC follows a more direct path to the target, and generates a tighter set of samples. This behavior can be attributed to the Riemann geometry cues and the resulting implicit system of update equations.

The higher variance in the samples shows up in the RMSE for the parameters. We show these numbers along with the autocorrelation times for both algorithms in Tables 1 and 2. As seen in the qualitative sample trajectories, the SGR-NPHMC generates samples with lower RMSE for all values of the noise corrector terms.

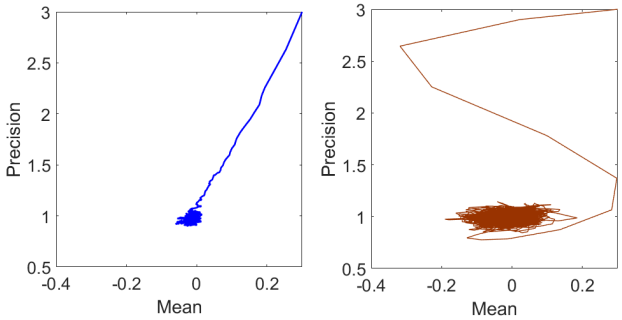
4.2. Parameter Estimation in Bayesian Logistic Regression

Our next experiment is on learning the parameters in a synthetic two-dimensional Bayesian logistic regression



(a) (b) (c)

Figure 1. Density plots for the mean (μ) and precision (τ) samples obtained from SGR-NPHMC and SG-NHT runs on the synthetic Gaussian dataset. Plots (a) and (b) show the sample densities of μ for SGR-NPHMC and SG-NHT respectively. Plot (c) shows the sample densities of τ for both algorithms. The true values were $\mu = 0, \tau = 1$. See the text for experimental details.



(a) (b)

Figure 2. Samples trajectories for μ and τ from (a) SGR-NPHMC and (b) SG-NHT runs on the synthetic Gaussian dataset. Both algorithms were initialized with $\mu_0 = 0.3, \tau_0 = 3$. For the former we used $\{A, B\} = 0.001$, and for the latter we had $A = 1$. The true values were $\mu = 0, \tau = 1$. Note the convergence patterns and sample spread of the two algorithms.

task. In this experiment we first generate 5000 datapoints from two bivariate normal distributions with means at $[1, -1]$ and $[-1, 1]$ and unit covariances, and then use a linear classifier with weights $(w_1, w_2) = [1, -1]$ to bag the points into two classes. We then estimate the classifier weights using Bayesian logistic regression. As with the previous experiment, we compare SGR-NPHMC and SG-NHT in terms of accuracy and autocorrelation time. We run both algorithms for 10^5 iterations, discard the first 5000 samples and use the rest to compute these metrics. SGR-NPHMC achieves lower RMSE for the parameters in this case as well, as seen in the corresponding tables provided in §D.1 of the supplementary. The sample trajectories shown in Figures 3a and 3b paint a similar picture to that of the previous section; SG-NHT overshoots by a wide margin before converging, and has higher sample variance as well. SGR-NPHMC follows a more efficient path to convergence, and post-convergence sample variance is lower.

Table 2. RMSE and auto-correlation times of the sampled means and precisions from SG-NHT runs on synthetic Gaussian data.

{A}	RMSE (μ)	RMSE (τ)	A.T. (μ)
0.1	0.0364	0.0386	13.5715
1	0.0375	0.0471	17.2241
10	0.0365	0.0416	13.5715

4.3. Topic Modeling using Hierarchical Gamma Processes

For this experiment we compare the algorithms in a high-dimensional topic modeling scenario using hierarchical Gamma processes. In particular, we use the Poisson factor analysis framework of (Zhou & Carin, 2015). We model the observed counts of V vocabulary terms in N documents as $\mathbf{D}_{V\times N} = \text{Poi}(\mathbf{\Phi\Theta})$, where $\mathbf{\Phi}_{V\times K}$ is the factor load matrix that encodes the relative importance of the vocabulary terms in the K latent topics, and $\mathbf{\Theta}_{K\times N}$ models the counts of the topics in the documents.

We put a Dirichlet prior on the columns of $\mathbf{\Phi}$ using normalized Gamma variables: $\phi_{v,k} = \frac{\gamma_v}{\sum_v \gamma_v}$, with $\gamma_v \sim \Gamma(\alpha, 1)$. Then we have $\theta_{n,k} \sim \Gamma(r_k, \frac{p_j}{1-p_j})$, where the document-specific mixing probabilities p_j have $\beta(a_0, b_0)$ priors. Next, we use two different formulations of r_k: (a) we set r_ks to the weights of a discrete Gamma process with equal atom weights, as in (Zhou & Carin, 2015); and (b) we set r_ks to the atom weights generated by the constructive Gamma process definition of (Roychowdhury & Kulis, 2014). See the respective papers for the details of the constructions. We call these two formulations γNB, and γGP respectively.

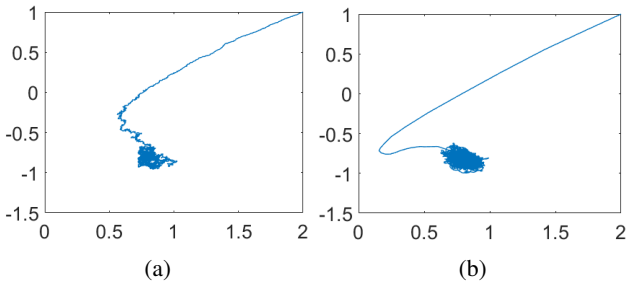


Figure 3. Samples trajectories for w_1 and w_2 from (a) SGR-NPHMC and (b) SG-NHT runs on the synthetic Bayesian logistic regression dataset. Both algorithms were initialized with $w_1 = 2, w_2 = 1$. For the former we used $\{A, B\} = 0.001$, and for the latter we had $A = 1$. The true values were $w_1 = 1, w_2 = -1$. Note the convergence patterns of the two algorithms, and the spread of the samples thereafter.

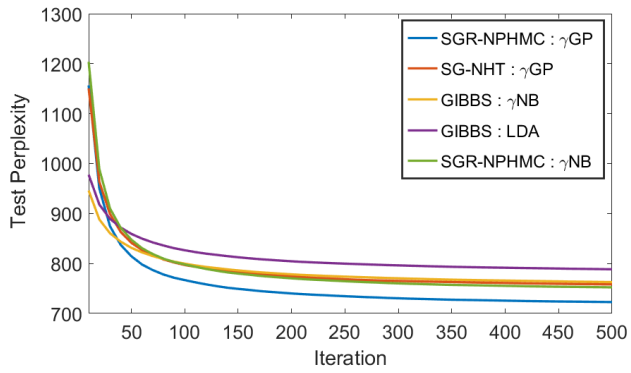


Figure 4. Test perplexities as a function of post-burnin iterations for the 20-Newsgroups dataset.

We use two public datasets for this experiment, the 20-Newsgroups and Reuters Corpus Volume 1 corpora from (Srivastava et al., 2013). The first has a vocabulary of 2,000 words spread over 18,845 documents. The second has 804,414 documents and a vocabulary of size 10,000. We use the same training/validation/test split as (Gan et al., 2015), where the 20 Newsgroups dataset is split chronologically into 11,314 training and 7,531 test documents, and the Reuters dataset into 794,414 training and 10,000 test documents. After training the stochastic algorithms, following standard methodology we learn document-specific parameters from 80% of the words in the test set, and calculate test perplexities on the remaining 20%. The perplexity formulation is detailed in §D.2 of the supplementary.

For SGR-NPHMC and SG-NHT on the γGP model, we run three parallel NPHMC chains; one each for the two constituent parameters of the atom weights (E_ks and T_ks) and one for the hyperparameters (α, γ and c). See §4.1 of (Roychowdhury & Kulis, 2014) for the exact formulation. We estimate the ϕs using Riemann Hamiltonian Monte Carlo updates (Girolami & Calderhead, 2011), as we found it to mix better than Gibbs sampling for the stochastic algorithms. For all Riemannian HMC chains we use the diagonal metric tensor $G(\boldsymbol{\theta}) = \text{diag}(\boldsymbol{\theta})^{-1}$, as first studied in (Patterson & Teh, 2013) (see §3.1.2 for the resulting dynamics). We used $K = 200$ latent topics for all algorithms. For SGR-NPHMC we set the learning rates of all three NPHMC chains to $1e$-4, and for SG-NHT we use a stable learning rate of $1e$-6. Batchsize was set to 100 for both algorithms. We used 2,000 iterations for burn-in and collected samples for test perplexity evaluation thereafter.

Table 3. Test perplexities on 20-Newsgroups and Reuters datasets.

METHOD	MODEL	20-NEWSGROUPS	REUTERS
GIBBS	γNB	763	-
GIBBS	LDA	788	-
SG-NHT	γGP	758	929
SGR-NPHMC	γNB	752	930
SGR-NPHMC	γGP	723	904

Figure 4 shows the perplexities evaluated on the 20-Newsgroups dataset. The perplexities at the end of the test runs for both 20-Newsgroups and Reuters are shown in Table 3. We can see the SGR-NPHMC algorithms for both γNB and γGP models outperforming SG-NHT for γGP on 20-newsgroups. For the larger Reuters dataset, the γGP-based SGR-NPHMC performs best, followed by SG-NHT and the γNB-based SGR-NPHMC.

5. Conclusion

We have proposed a novel Hamiltonian MCMC algorithm using a modified Nosé-Poincaré Hamiltonian augmented with Riemann preconditioning for both real and extended momenta, as well as correction terms arising from the Fokker-Planck equations to ensure sampling from the canonical ensemble in the presence of stochastic noise. We have derived a discretized sampler using the generalized leapfrog algorithm, and have shown robust performance in synthetic and high dimensional real-world datasets.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work was partially supported by NSF awards IIS 1217433 and DMS #1418265.

References

Bond, S. D., Leimkuhler, B. J., and Laird, B. B. The Nosé-Poincaré Method for Constant Temperature Molecular Dynamics. *J. Comput. Phys*, 151:114–134, 1999.

Chen, T., Chen, E., and Guestrin, C. Stochastic Gradient Hamiltonian Monte Carlo. In *ICML*, 2014.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. Bayesian Sampling using Stochastic Gradient Thermostats. In *NIPS*, 2014.

Duane, S., Kennedy, A.D., Pendleton, B.J., and Roweth, D. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Gan, Z., Chen, C., Henao, R., Carlson, D., and Carin, L. Scalable Deep Poisson Factor Analysis for Topic Modeling. In *ICML*, 2015.

Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Hoover, W.G. Canonical dynamics: Equilibrium phase-space distributions. *Physical Review A (General Physics)*, 31(3):1695–1697, 1985.

Leimkuhler, B. and Reich, S. *Simulating Hamiltonian Dynamics*. Cambridge University Press, 2004.

Neal, R. M. MCMC using Hamiltonian dynamics. In Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L. (eds.), *Handbook of Markov Chain Monte Carlo*, pp. 113–162. Chapman & Hall / CRC Press, 2011.

Nosé, S. A molecular dynamics method for simulations in the canonical ensemble. *Molecular Physics*, 52(2):255–268, 1984.

Patterson, S. and Teh, Y. W. Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex. In *NIPS*, 2013.

Robbins, H. and Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 1951.

Roychowdhury, A. and Kulis, B. Gamma Processes, Stick-Breaking, and Variational Inference, 2014. arXiv:1410.1068.

Srivastava, N., Salakhutdinov, R., and Hinton, G. E. Modeling documents with deep Boltzmann machines. In *UAI*, 2013.

Welling, M. and Teh, Y. W. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *ICML*, 2011.

Yin, L. and Ao, P. Existence and Construction of Dynamical Potential in Nonequilibrium Processes without Detailed Balance. *Journal of Physics A: Mathematical and General*, 39(27):8593, 2006.

Zhou, M. and Carin, L. Negative Binomial Process Count and Mixture Modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):307–320, 2015.