

Bayesian Theory and Computation

Lecture 11: Scalable MCMC



Cheng Zhang

School of Mathematical Sciences, Peking University

Apr 02, 2025

- ▶ Large scale datasets are becoming more commonly available across many fields. Learning complex models from these datasets is the future
- ▶ While many modern MCMC methods have been proposed in recent years, they usually require expensive computation when the data size is large
- ▶ In this lecture, we will discuss recent development on Markov chain Monte Carlo methods that are applicable to large scale datasets
 - ▶ Best of both worlds: scalability, and Bayesian protection against overfitting



- **Stochastic differential equations** are widely used to model dynamical systems with noise

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t$$

where B denotes a Wiener process/Brownian motion

- Now suppose the probability density for X_t is $p(x, t)$, we are interested in how $p(x, t)$ evolves along time
- For example, does it converge to some distribution? If it does, how can we find it out?



- Consider $Y_t = g(X_t)$, where g is a test function with certain regularity. By Itô's formula

$$dY_t = \left(\nabla g(X_t) \cdot \mu + \frac{1}{2} \text{tr}(\nabla^2 g(X_t) \sigma \sigma^T) \right) dt + \nabla g(X_t) \sigma dB_t$$

- Integrate both side on time interval $[t, t+h]$ and take expectation

$$\mathbb{E} \frac{Y_{t+h} - Y_t}{h} = \frac{1}{h} \int_t^{t+h} \mathbb{E} \left(\nabla g \cdot \mu + \frac{1}{2} \text{tr}(\nabla^2 g \sigma \sigma^T) \right) ds$$

- Let $h \rightarrow 0$ and assume $g(x) \rightarrow 0$ as $\|x\| \rightarrow \infty$

$$\int g(x) \frac{\partial p(x, t)}{\partial t} dx = \int g(x) \left(-\nabla \cdot (\mu p) + \nabla^2 : \left(\frac{1}{2} \sigma \sigma^T p \right) \right) dx$$

- It turns out the $p(x, t)$ satisfies the **Fokker-Planck equation** (also known as the **Kolmogorov forward equation**)

$$\frac{\partial p(x, t)}{\partial t} = - \sum_i \frac{\partial}{\partial x_i} (\mu_i(x, t) p(x, t)) + \sum_{i, j} \frac{\partial^2}{\partial x_i \partial x_j} (D_{ij}(x, t) p(x, t))$$

where $D = \frac{1}{2} \sigma \sigma^T$ is the diffuse tensor

- Example: Wiener process $dX_t = dB_t$

$$\frac{\partial p(x, t)}{\partial t} = \frac{1}{2} \frac{\partial^2}{\partial x^2} p(x, t)$$

If $p(x, 0) = \delta(x)$, the solution is $p(x, t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}$



- Suppose that we have a large number of data items

$$\mathcal{D} = \{x_1, x_2, \dots, x_N\}$$

where $N \gg 1$

- The log-posterior (up to a constant) is

$$\log p(\theta|X) = \log p(\theta) + \sum_{i=1}^N \log p(x_i|\theta) \sim \mathcal{O}(N)$$

- How to reduce this computation in MCMC without damaging the convergence to the target distribution?



- ▶ Also known as stochastic approximation
- ▶ At each iteration
 - ▶ Get a subset (minibatch) x_{t_1}, \dots, x_{t_n} of data items where $n \ll N$
 - ▶ Approximate gradient of log-posterior using the subset

$$\nabla \log p(\theta_t | X) \approx \nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^n \nabla \log p(x_{t_i} | \theta_t)$$

- ▶ Take a gradient step

$$\theta_{t+1} = \theta_t + \frac{\epsilon_t}{2} \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^n \nabla \log p(x_{t_i} | \theta_t) \right)$$



- ▶ Major requirement for convergence on step-sizes

$$\sum_{t=1}^{\infty} \epsilon_t = \infty, \quad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

- ▶ Intuition
 - ▶ Step sizes cannot decrease too fast, otherwise will not be able to explore parameter space
 - ▶ Step sizes must decrease to zero, otherwise will not converge to a local mode



- ▶ **First order Langevin dynamics** can be described by the following stochastic different equation

$$d\theta_t = \frac{1}{2} \nabla \log p(\theta_t | X) dt + dB_t$$

- ▶ The above dynamical system converges to the target distribution $p(\theta | X)$ (easy to verify via the Fokker-Planck equation)
- ▶ Intuition
 - ▶ Gradient term encourages dynamics to spend more time in high probability areas
 - ▶ Brownian motion provides noise so that dynamics will explore the whole parameter space



- First order Euler discretization

$$\theta_{t+1} = \theta_t + \frac{\epsilon}{2} \nabla \log p(\theta_t | X) + \eta_t, \quad \eta_t = \mathcal{N}(0, \epsilon)$$

- Amount of noise is balanced to gradient step size
- With finite step size, there will be discretization errors. We can add MH correction step to fix it, and this is **MALA**!
- As step size $\epsilon \rightarrow 0$, acceptance rate goes to 1



- ▶ Introduced by Welling and Teh (2011)
- ▶ **Idea:** use stochastic gradients in Langevin dynamics

$$\theta_{t+1} = \theta_t + \frac{\epsilon_t}{2} g(\theta_t) + \eta_t, \quad \eta_t = \mathcal{N}(0, \epsilon_t)$$

$$g(\theta_t) = \nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^n \nabla \log p(x_{t_i} | \theta_t)$$

- ▶ Update is just stochastic gradient ascent plus Gaussian noise
- ▶ Noise variance is balanced with gradient step sizes
- ▶ require step size ϵ_t decrease to 0 slowly



- ▶ **Controllable** stochastic gradient noise. The stochastic gradient estimate $g(\theta_t)$ is unbiased, but it introduces noise

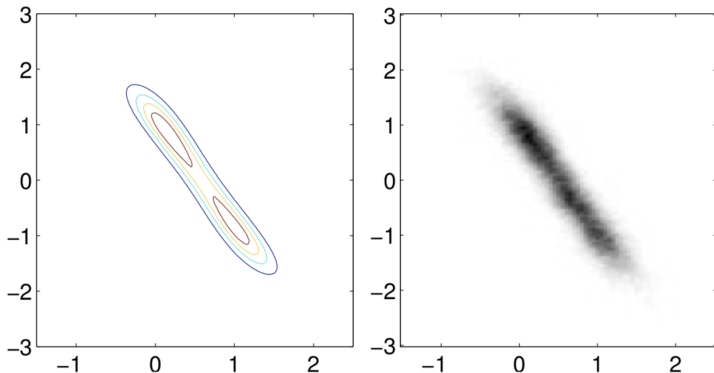
$$g(\theta_t) = \nabla \log p(\theta|X) + \mathcal{N}(0, V(\theta_t))$$

- ▶ Stochastic gradient noise $\sim \mathcal{N}(0, \mathcal{O}(\epsilon_t^2))$
- ▶ Injected noise $\eta_t \sim \mathcal{N}(0, \epsilon_t)$
- ▶ When $\epsilon_t \rightarrow 0$
 - ▶ **Stochastic gradient noise will be dominated by injected noise η_t** , so can be ignored. SGLD then recovers Langevin dynamics updates with decreasing step sizes
 - ▶ MH acceptance probability approaches 1, so we can ignore the expensive MH correction step
 - ▶ If ϵ_t approaches 0 slowly enough, the discretized Langevin dynamics is still able to explore the whole parameter space

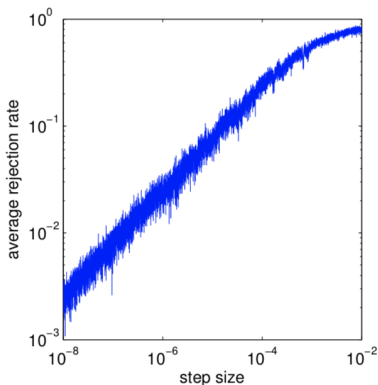
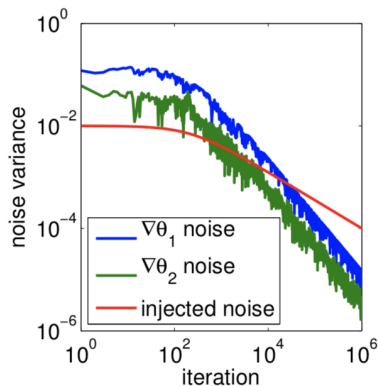


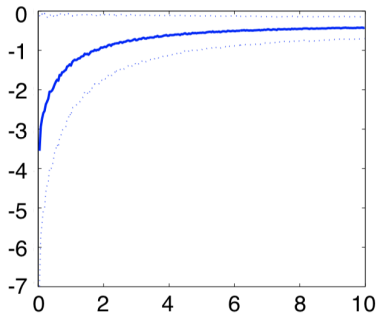
$$\theta_1 \sim \mathcal{N}(0, \sigma_1^2), \quad \theta_2 \sim \mathcal{N}(0, \sigma_2^2)$$

$$x_i \sim \frac{1}{2}\mathcal{N}(\theta_1, \sigma_x^2) + \frac{1}{2}\mathcal{N}(\theta_1 + \theta_2, \sigma_x^2)$$

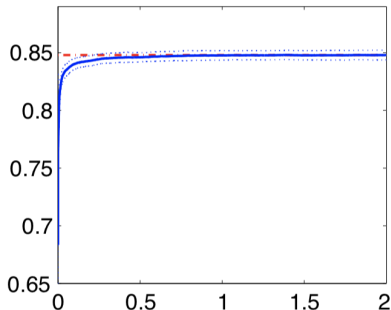


Noise and rejection probability





Log probability vs epochs



Test accuracy vs epochs



- ▶ Now that stochastic gradient scales MALA, it seems straightforward to use stochastic gradient for HMC

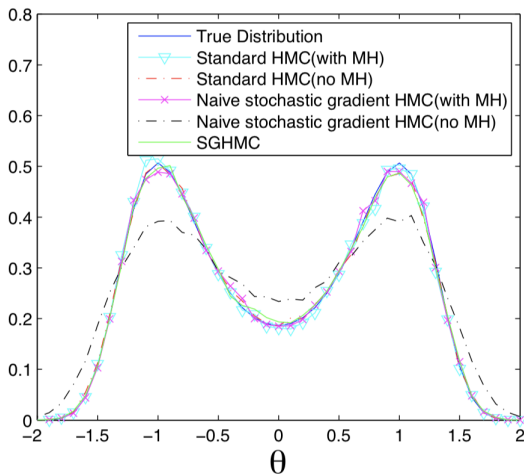
$$d\theta = M^{-1}r dt$$

$$dr = g(\theta)dt = -\nabla U(\theta)dt + \sqrt{\epsilon V(\theta)}dB_t$$

- ▶ However, the resulting dynamics does not leave $p(\theta, r)$ invariant (can be verified via Fokker-Planck equation)
- ▶ This deviation can be saved by MH correction, but that leads to a complex computation vs efficiency trade-off
 - ▶ Short runs reduce deviation, but requires more expensive HM steps and does not full utilize the exploration of the Hamiltonian dynamics
 - ▶ Long runs lead to low acceptance rates, waste of computation



$$U(\theta) = -2\theta^2 + \theta^4$$



- ▶ We can introduce friction into the dynamical system to reduce the influence of the gradient noise, which leads to the **second order Langevin dynamics**

$$\begin{aligned}d\theta &= M^{-1}r dt \\ dr &= -\nabla U(\theta)dt - CM^{-1}r dt + \sqrt{2C}dB_t\end{aligned}\tag{2}$$

- ▶ Consider the joint space $z = (\theta, r)$, rewrite (2)

$$dz = -[D + G]\nabla H(z)dt + \sqrt{2D}dB_t$$

where

$$G = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix}$$

- ▶ $p(\theta, r) \propto \exp(-H(\theta, r))$ is the unique stationary distribution of (2)



- ▶ Introduced by Chen et al (2014)
- ▶ Use stochastic gradient in the second order Langevin dynamics. In each iteration
 - ▶ resample momentum $r^{(t)} \sim \mathcal{N}(0, M)$ (optional),
 $(\theta_0, r_0) = (\theta^{(t)}, r^{(t)})$
 - ▶ simulate dynamics in (2)

$$\theta_i = \theta_{i-1} + \epsilon_t M^{-1} r_{i-1}$$

$$r_i = r_{i-1} + \epsilon_t g(\theta_i) - \epsilon_t C M^{-1} r_{i-1} + \mathcal{N}(0, 2C\epsilon_t)$$

- ▶ update the parameter $(\theta^{(t+1)}, r^{(t+1)}) = (\theta_m, r_m)$, no MH correction step
- ▶ Similarly, the stochastic gradient noise is controllable, and when $\epsilon_t \rightarrow 0$, **SGHMC** recovers the second order Langevin dynamics



- ▶ Let $v = \epsilon M^{-1}r$, we can rewrite the update rule in SGHMC

$$\Delta v = \epsilon^2 M^{-1}g(\theta) - \epsilon M^{-1}Cv + \mathcal{N}(0, 2\epsilon^3 M^{-1}CM^{-1})$$

$$\Delta\theta = v$$

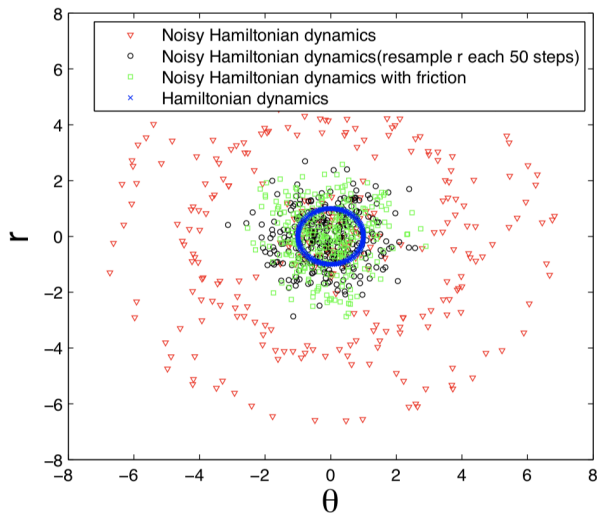
- ▶ Define $\eta = \epsilon^2 M^{-1}$, $\alpha = \epsilon M^{-1}C$, the update rule becomes

$$\Delta v = \eta g(\theta) - \alpha v + \mathcal{N}(0, 2\alpha\eta)$$

$$\Delta\theta = v$$

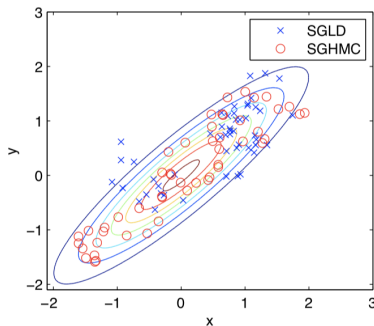
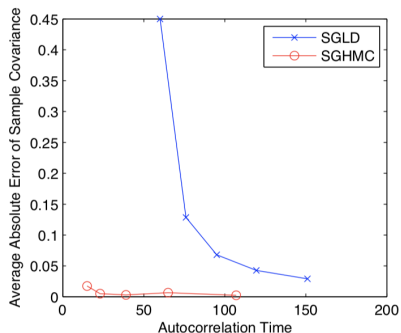
- ▶ If we ignore the noise term, this is basically SGD with momentum where η is the learning rate and $1 - \alpha$ the momentum coefficient
- ▶ This connection can be used to guide our choices of SGHMC hyper-parameters

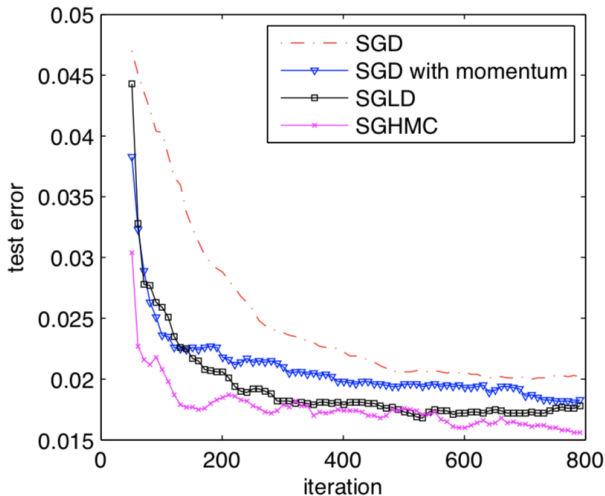




SGHMC vs SGLD on a bivariate Gaussian with correlation

$$U(\theta) = \frac{1}{2} \theta^T \Sigma^{-1} \theta, \quad \Sigma^{-1} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$





- ▶ Stochastic gradient in SGHMC introduces noise. With step size ϵ , the corresponding dynamics is

$$d\theta = M^{-1}r dt$$

$$dr = -\nabla U(\theta)dt - CM^{-1}dt + \sqrt{2(C + \frac{1}{2}\epsilon V(\theta))}dB_t$$

- ▶ If somehow we correct the mismatch between friction coefficient and the real noise level, we can improve the approximation accuracy for a finite ϵ
- ▶ But how can we do that given that the noise $V(\theta)$ is unknown?



- ▶ One missing key fact is the **thermal equilibrium condition**:

$$p(\theta, r) \propto \exp(-(U(\theta) + K(r))/T) \Rightarrow T = \frac{1}{d} \mathbb{E}(r^T r)$$

- ▶ Unfortunately, using stochastic gradients destroys the thermal equilibrium condition
- ▶ We can introduce an additional variable ξ that adaptively controls the mean kinetic energy, and use the following dynamics

$$\begin{aligned} d\theta &= r dt, & dr &= g(\theta) dt - \xi r dt + \sqrt{2A} dB_t \\ d\xi &= \left(\frac{1}{d} r^T r - 1\right) dt \end{aligned} \tag{3}$$

- ▶ (3) is known as the **Nosé-Hoover thermostat** in statistical physics.



- ▶ Introduced by Ding et al (2014)
- ▶ The algorithm
 - ▶ Initialized $\theta_0, r_0 \sim \mathcal{N}(0, I)$, and $\xi_0 = A$
 - ▶ For $t = 1, 2, \dots$

$$r_t = r_{t-1} + \epsilon_t g(\theta_{t-1}) - \epsilon_t \xi_{t-1} r_{t-1} + \sqrt{2A} \mathcal{N}(0, \epsilon)$$

$$\theta_t = \theta_{t-1} + \epsilon_t r_t$$

$$\xi_t = \xi_{t-1} + \epsilon_t ((r^{(t)})^T r^{(t)} / d - 1)$$

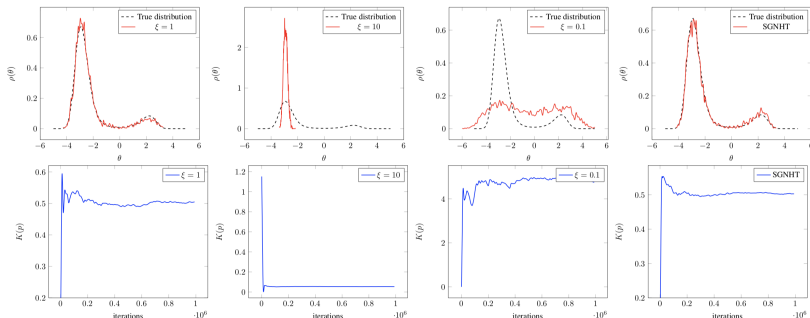
- ▶ The thermostat ξ helps to adjust the friction according to the real noise level, and maintains the right mean kinetic energy
 - ▶ When mean kinetic energy is high, ξ get bigger, increasing friction to cool down the system
 - ▶ When mean kinetic energy is low, ξ get smaller, reducing friction to heat up the system



$$U(\theta) = (\theta + 4)(\theta + 1)(\theta - 1)(\theta - 3)/14 + 0.5$$

$$g(\theta)\epsilon = -\nabla U(\theta)\epsilon + \mathcal{N}(0, 2B\epsilon), \quad \epsilon = 0.01, B = 1$$

For SGNHT, we set $A = 0$



- Consider the following stochastic differential equation

$$d\Gamma = v(\Gamma)dt + \mathcal{N}(0, 2D(\theta)dt)$$

where $\Gamma = (\theta, r, \xi)$.

- $p(\Gamma) \propto \exp(-H(\Gamma))$ is the stationary distribution if

$$\nabla \cdot (p(\Gamma)v(\Gamma)) = \nabla \nabla^T : (p(\Gamma)D)$$

We can construct H such that the marginal distribution is $p(\theta) \propto \exp(-U(\theta))$.

- For SGNHT, $H(\Gamma) = U(\theta) + \frac{1}{2}r^T r + \frac{d}{2}(\xi - A)^2$

$$v(\Gamma) = \begin{bmatrix} r \\ -\nabla U(\theta) - \xi r \\ r^T r/d - 1 \end{bmatrix}, \quad D(\theta) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



- ▶ Introduced by Ma et al (2015)
- ▶ Assume target distribution $p(\theta|X)$ is the marginal distribution of $p(z) \propto \exp(-H(z))$
- ▶ We consider the following stochastic differential equation

$$dz = -(D(z) + Q(z))\nabla H(z)dt + \Gamma(z)dt + \sqrt{2D(z)}dB_t$$

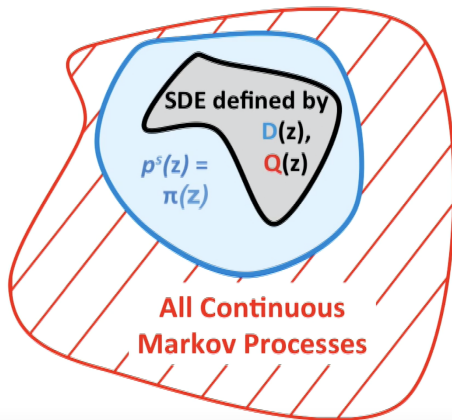
$$\Gamma_i(z) = \sum_{j=1}^d \frac{\partial}{\partial z_j} (D_{ij}(z) + Q_{ij}(z))$$

- ▶ $Q(z)$ is a skew-symmetric curl matrix
- ▶ $D(z)$ denotes the positive semidefinite diffusion matrix
- ▶ The above dynamics leaves $p(z)$ invariant



All existing samplers can be written in framework

- ▶ HMC
- ▶ Riemannian HMC
- ▶ Langevin Dynamics (LD)
- ▶ Riemannian LD



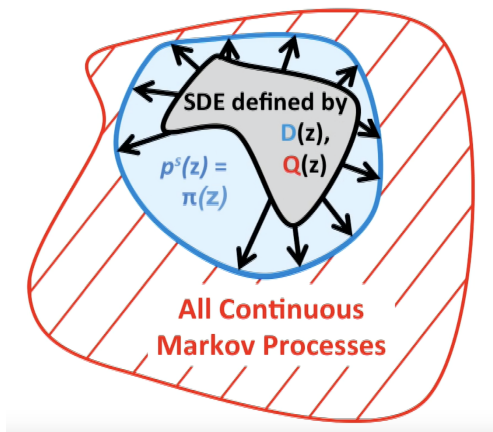
Adapted from Emily Fox 2017



All existing samplers can be written in framework

- ▶ HMC
- ▶ Riemannian HMC
- ▶ Langevin Dynamics (LD)
- ▶ Riemannian LD

Any valid sampler has a D and Q in the framework



Adapted from Emily Fox 2017



- Consider ϵ -discretization

$$z_{t+1} = z_t - \epsilon_t ((D(z_t) + Q(z_t)) \nabla H(z_t) + \Gamma(z_t)) + \mathcal{N}(0, 2\epsilon_t D(z_t))$$

- The gradient computation in $\nabla H(z_t)$ could be expensive, can be replaced with stochastic gradient $\nabla \tilde{H}(z_t)$

$$z_{t+1} = z_t - \epsilon_t ((D(z_t) + Q(z_t)) \nabla \tilde{H}(z_t) + \Gamma(z_t)) + \mathcal{N}(0, 2\epsilon_t D(z_t))$$

- The gradient noise is still controllable

$$\nabla \tilde{H}(z_t) = \nabla H(z_t) + (\mathcal{N}(0, V(\theta)), 0)^T$$

- stochastic gradient noise $\sim \mathcal{N}(0, \epsilon_t^2 V(\theta))$
- injected noise $\sim \mathcal{N}(0, 2\epsilon_t D(z_t))$



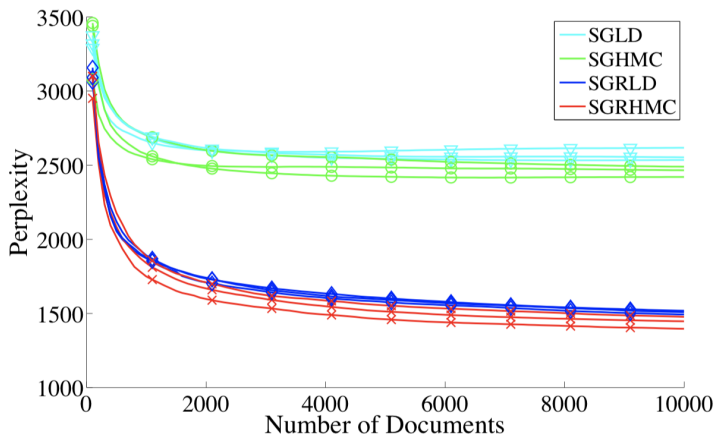
- ▶ As shown before, previous stochastic gradient MCMC algorithms all cast into this framework
- ▶ Moreover, the framework helps to develop new samplers without requiring significant physical intuition
- ▶ Consider $H(\theta, r) = U(\theta) + \frac{1}{2}r^T r$, modify D and Q to account for the geometry

$$D(\theta, r) = \begin{pmatrix} 0 & 0 \\ 0 & G(\theta)^{-1} \end{pmatrix}, \quad Q(\theta, r) = \begin{pmatrix} 0 & -G(\theta)^{-1/2} \\ G(\theta)^{-1/2} & 0 \end{pmatrix}$$

Note that this works for any positive definite $G(\theta)$, not just the fisher information metric

Applied SGRHMC to online LDA

- each entry was analyzed on the fly



- ▶ Reduce the computation in MH correction step via subsets of data (Korattikara et al 2014)
- ▶ Divide and conquer: divide the entire data set into small chunks, run MCMC in parallel for these subsets of data, and merge the results for the true posterior approximation (Scott et al 2016)
- ▶ Using deterministic approximation instead of stochastic gradients. This may introduce some bias, but remove the unknown noise for gradient estimation, allowing for better exploration efficiency
 - ▶ **Gaussian processes**: Rasmussen 2003, Lan et al 2016
 - ▶ **Reproducing kernel Hilbert space**: Strathmann et al 2015
 - ▶ **Random Bases**: Zhang et al 2017



- ▶ M. Welling and Y.W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th International Conference on Machine Learning (ICML'11), pages 681–688, June 2011.
- ▶ T. Chen, E.B. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In Proceeding of 31st International Conference on Machine Learning (ICML'14), 2014.
- ▶ N. Ding, Y. Fang, R. Babbush, C. Chen, R.D. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. In Advances in Neural Information Processing Systems 27 (NIPS'14). 2014.

- ▶ Ma, Y.A., Chen, T. and Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In Advances in Neural Information Processing Systems 2917–2925.
- ▶ A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In Proceedings of the 30th International Conference on Machine Learning (ICML'14), 2014.
- ▶ Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I. and McCulloch, R. E. (2016). Bayes and Big Data: The Consensus Monte Carlo Algorithm. International Journal of Management Science and Engineering Management 11 78-88.



- ▶ Rasmussen, C. E. (2003). “Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals.” *Bayesian Statistics*, 7: 651–659.
- ▶ Lan, S., Bui-Thanh, T., Christie, M., and Girolami, M. (2016). Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems. *J. Comput. Phys.*, 308:81–101.
- ▶ Strathmann, H., Sejdinovic, D., Livingstone, S., Szabo, Z., and Gretton, A. Gradient-free Hamiltonian monte carlo with efficient kernel exponential families. In *Advances in Neural Information Processing Systems*, pp. 955–963, 2015

- ▶ Zhang, C., Shahbaba, B., and Zhao, H. (2017).
“Hamiltonian Monte Carlo acceleration using surrogate functions with random bases.” *Statistics and Computing*, 1–18.

