Stochastic Gradient Hamiltonian Monte Carlo

Han Xiao

Department of Computer Science, University of Helsinki

han.xiao@cs.helsinki.com

November 19, 2014

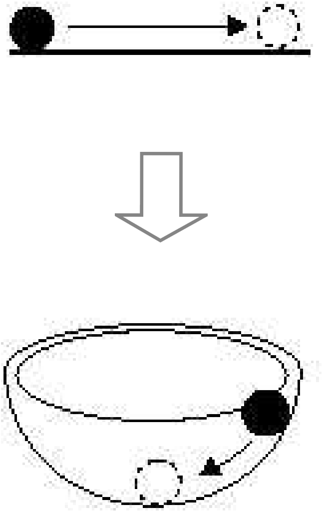
Overview

1. Hamiltonian Monte Carlo(HMC)
2. Naive Stochastic Gradient HMC(Naive SGHMC)
3. Stochastic Gradient HMC with friction(SGHMC)
4. Experiment

HMC: Motivation

- Random-walk approach: slow exploration.
- Explore faster?

HMC: Physical Analogy



HMC: Hamiltonian Dynamics

$$\begin{cases} d\theta &= \nabla K(r)\, dt = M^{-1} r\, dt \\ dr &= -\nabla U(\theta)\, dt \end{cases}$$

- θ: position/target variables
- r: momentum/auxiliary variables
- $K(r)$: the kinetic energy, $\frac{1}{2} M^{-1} r^2$
- $U(\theta)$: the potential energy
- M: mass matrix

HMC: Hamiltonian Dynamics

$$\begin{aligned}\pi(\theta, r) &\propto \exp(-U(\theta) - K(r)) \\ &\propto \exp(-U(\theta)) \exp(-K(r))\end{aligned}$$

- $U(\theta) = -\log(p(\theta|\mathcal{D}))$, $\mathcal{D}$, the observation

HMC: Discretization - Leapfrog Method

function $\text{LEAPFROG}(\theta_0, r_0, \epsilon, m)$
  $r_0 \leftarrow r_0 - \frac{\epsilon}{2}\nabla U(\theta_0)$
  for $i = 1$ to m **do**
    $\theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1}$
    $r_i \leftarrow r_{i-1} - \epsilon \nabla U(\theta_i)$
  end for
  $r_m \leftarrow r_m - \frac{\epsilon}{2}\nabla U(\theta_m)$
end function

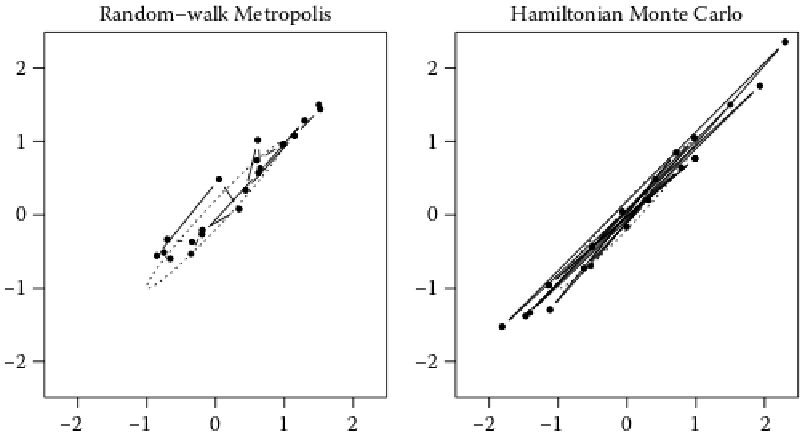
HMC: Put Together

function HMC(position $\theta^{(1)}$, step size ϵ, step number m)
  for $t = 1, 2 \cdots$ **do**
    Resample momentum r
    $r \sim \mathcal{N}(0, M)$
    $(\theta_0, r_0) \leftarrow (\theta^{(t)}, r)$

    Simulating Hamiltonian trajectory by leapfrog
    $\hat{\theta}, \hat{r} \leftarrow \mathrm{LEAPFROG}(\theta_0, r_0, \epsilon, m)$

    MH correction
    if $\mathcal{U}(0,1) < min(1, \exp H(\hat{\theta}, \hat{r}) - H(\theta, r))$ **then**
      $\theta^{(t+1)} = \hat{\theta}$
    else
      $\theta^{(t+1)} = \theta^{(t)}$
    end if
  end for
end function

HMC: Comparison to Random-walk



Stochastic Gradient HMC: Problem Definition

$$p(\theta|\mathcal{D}) \propto exp(-U(\theta))$$

$$U(\theta) = -\sum_{x\in\mathcal{D}} \log p(x|\theta) - \log p(\theta)$$

Where:

- θ: target variables
- $\mathcal{D}$: i.i.d observations

Stochastic Gradient HMC: Motivation

$$\nabla U(\theta) = \sum_{x \in \mathcal{D}} \nabla \log p(x|\theta) - \nabla \log p(\theta)$$

If $|\mathcal{D}| =$ one billion?

Naive SGHMC: Stochastic Gradient

$$\nabla \widetilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\widetilde{\mathcal{D}}|} \sum_{x \in \widetilde{\mathcal{D}}} \nabla \log p(x|\theta) - \nabla \log p(\theta), \; \widetilde{\mathcal{D}} \subset \mathcal{D}$$

Naive SGHMC: Gradient Noise Assumption

$$\nabla \widetilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, V(\theta))$$

Where:

- V: the covariance of the stochastic gradient noise

Naive SGHMC: Dynamics

$$
\begin{cases}
d\theta &= M^{-1} r \, dt \\
dr &= -\nabla U(\theta) \, dt + \mathcal{N}(0, 2B) \, dt
\end{cases}
$$

Where:

- $B = \frac{1}{2}\epsilon V$: the diffusion matrix(?)

Naive SGHMC: Not Invariant Anymore

$$t \uparrow \Longrightarrow h(p_t) \uparrow$$
$$p_t \to \text{uniform}$$

- $p_t(\theta, r)$: the distribution of (θ, r) at t.
- $h(p_t)$: entropy of p_t

Naive SGHMC: The Dilemma

1. With MH correction, calculating $U(\theta)$ is expensive
2. Without MH correction, far from the target $\pi(\theta, t)$

Naive SGHMC: Fixing It

- Do MH using subset of data
- Eliminate/reduce the gradient noise

SGHMC: Dynamics

$$
\begin{cases} d\theta & = M^{-1} r \, dt \\ dr & = -\nabla U(\theta) \, dt - BM^{-1} r \, dt + \mathcal{N}(0, 2B) \, dt \end{cases}
$$

- $BM^{-1}r$: the friction term
- Commonly referred to as *second-order Langevin dynamics*

SGHMC: Invariance

It can be proved that:

$$\partial_t p_t(\theta, r) = 0$$

SGHMC: in Practice

- B is unknown but can be estimated
- Beneficial to use friction constant term C(user-specified)

SGHMC: in Practice

$$
\begin{cases}
d\theta = & M^{-1} r \, dt \\
dr = & -\nabla U(\theta) \, dt - C M^{-1} r \, dt \\
 & + \mathcal{N}(0, 2(C - \hat{B})) \, dt + \mathcal{N}(0, 2B) \, dt
\end{cases}
$$

- $\hat{B}$: estimated B

SGHMC: in Practice

- As $\epsilon \to 0$, $B = \frac{1}{2}\epsilon V \to 0$.
- Assume $\hat{B} = 0$ for simplicity and *goodness*.

$$\begin{cases} d\theta = & M^{-1} r \, dt \\ dr = & -\nabla U(\theta) dt - CM^{-1} r dt + \mathcal{N}(0, 2C)) dt \end{cases}$$

- **Invariant** again.

Algorithm

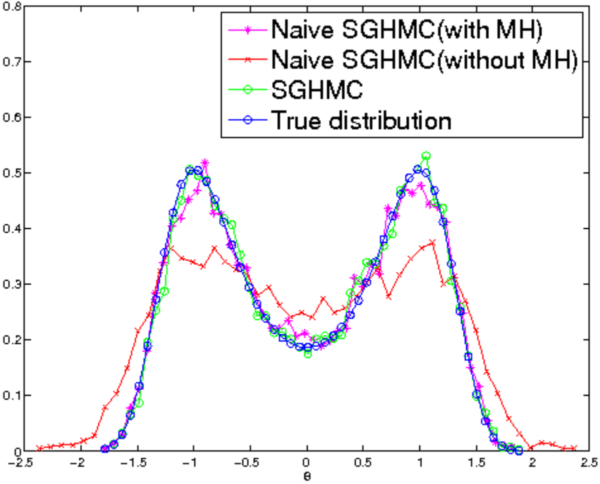
function STOCHASTIC-GRADIENT-HMC($\theta^{(1)}$, ϵ_t, m)
  for $t = 1, 2 \cdots$ **do**
    Optionally resample momentum r
    $r \ \sim \mathcal{N}(0, M)$
    $(\theta_0, r_0) \leftarrow (\theta^{(t)}, r)$

    Simulate noisy Hamiltonian trajectory
    for $i = 1$ to m **do**
      $\theta_i \leftarrow \theta_{i-1} + \epsilon_t M^{-1} r_{i-1}$
      $r_i \leftarrow r_{i-1} - \epsilon_t \nabla \widetilde{U}(\theta_i) - \epsilon_t C M^{-1} r_{i-1} + \epsilon_t \mathcal{N}(0, 2(C - \hat{B}))$
    end for
    No MH correction
    $(\theta^{(t+1)}, r^{(t+1)}) = (\theta_m, r_m)$
  end for
end function

SGHMC: How It Works

- $U(\theta) = -2\theta^2 + \theta^4$
- $\nabla \widetilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, 4)$
- $\epsilon = 0.1, \ \ m = 10$
- $C = 1$
- Sampled 10000 data points

SGHMC: How It Works



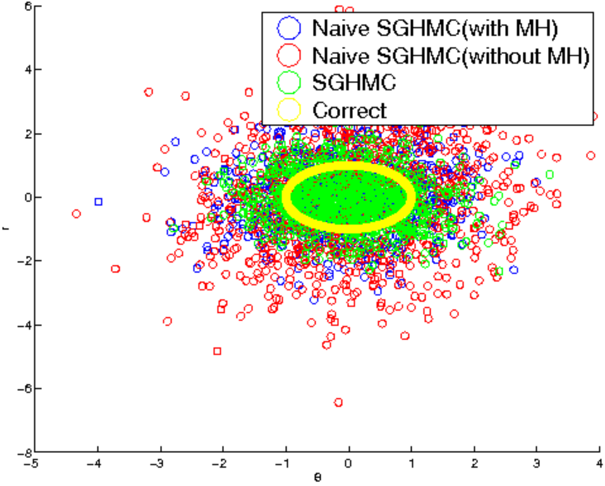
SGHMC: How It Works

METHOD	ACCEPTANCE RATE
Naive SGHMC(with MH)	0.62
Naive SGHMC(without MH)	1.0
SGHMC(with MH)	0.76

SGHMC: How It Works

- $U(\theta) = \frac{1}{2}\theta^2$
- $\nabla \widetilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, 4)$
- $\epsilon = 0.1, \;\; m = 10$
- $C = 2$
- Sampled 1000 data points

SGHMC: How It Works



SGHMC: Connection to SGLD

When the friction is large, SGHMC reduces to *Stochastic Gradient Langevin Dynamics*.

Illustration:

1. suppose $BM^{-1} = \frac{1}{dt}$.
2. r converges to $\mathcal{N}(MB^{-1}\nabla U(\theta), M)$ fast.
3. dynamics for θ becomes:

$$d\theta = -M^{-1}\nabla U(\theta)dt^2 + \mathcal{N}(0, 2M^{-1}dt^2)$$

where:

- M^{-1}, the preconditioning matrix in SGLD

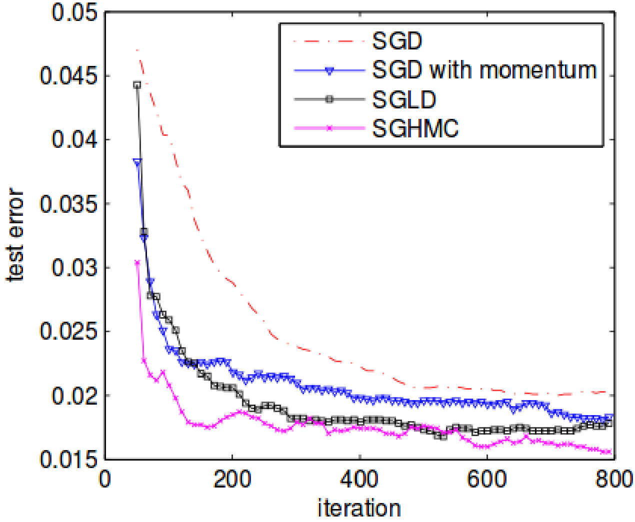
Bayesian Neural Networks for Classification: Data

- Handwritten digit classification
- MNIST dataset
- 60,000 instances for training and 10,000 for testing

Bayesian Neural Networks for Classification: Model

- 2-layer Bayesian neural network
- 100 hidden variables using sigmoid unit
- Output layer using softmax

Bayesian Neural Networks for Classification: Result



Online Bayesian Probabilistic Matrix Factorization

- Recommend movies to users
- 1 million ratings of ≈ 4000 movies by ≈ 6000 users

Online Bayesian Probabilistic Matrix Factorization

METHOD	**RMSE**
SGD	0.8538 ± 0.0009
SGD with momentum	0.8539 ± 0.0009
SGLD	0.8412 ± 0.0009
SGHMC	0.8411 ± 0.0011

Thank you!