Stochastic Gradient Hamiltonian Monte Carlo

Tianqi Chen, Emily B. Fox, and Carlos Guestrin

MODE Lab, University of Washington, Seattle, WA

Presented by: Wenzhao Lian
February 13, 2015
Motivation

- Markov Chain Monte Carlo (MCMC) is a common way to do posterior inference in Bayesian methods.
- Hamiltonian Monte Carlo (HMC) has been a recent popular approach of sampling from the posterior distribution, usually with better mixing rate.
- HMC requires intensive computation when evaluating the gradient of log-posterior on the full data, and when computing the probability of acceptance if MH is adopted.
Markov Chain Monte Carlo (MCMC) is a common way to do posterior inference in Bayesian methods.

Hamiltonian Monte Carlo (HMC) has been a recent popular approach of sampling from the posterior distribution, usually with better mixing rate.

HMC requires intensive computation when evaluating the gradient of log-posterior on the full data, and when computing the probability of acceptance if MH is adopted.

Problem: how to scale HMC to large datasets?
Hamiltonian Monte Carlo

Goal: to sample from the posterior of parameters $\theta$ given data $\mathcal{D}$.

\[
p(\theta|\mathcal{D}) \propto \exp(-U(\theta)) \tag{1}
\]

\[
U(\theta) = -\sum_{x \in \mathcal{D}} \log p(x|\theta) - \log p(\theta) \tag{2}
\]

\[
\pi(\theta, r) \propto \exp(-U(\theta) - \frac{1}{2}r^T M^{-1} r) = \exp(-H(\theta, r)) \tag{3}
\]

\[
H(\theta, r) = U(\theta) + K(r) \tag{4}
\]

$H(\theta, r)$ is the Hamiltonian, corresponding to the system total energy. $U(\theta)$ is the potential energy; $\theta$ is the position. $K(r) = \frac{1}{2}r^T M^{-1} r$ is the kinetic energy; $r$ is the momentum. This $K(r)$ corresponds to a Gaussian distribution $r \sim \mathcal{N}(0, M)$; in particular, $K(r) = K(-r)$. $M$ is a positive-definite “mass matrix” (often assumed to be a diagonal matrix).
HMC simulates the Hamiltonian dynamics to propose samples,

\[ d\theta = \mathbf{M}^{-1} r dt \]  
\[ dr = -\nabla U(\theta) dt \]

\[ \pi(\theta, r) \] is invariant.
HMC simulates the Hamiltonian dynamics to propose samples,

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

\[ dr = -\nabla U(\theta) dt \]

\( \pi(\theta, r) \) is invariant.

HMC propose samples via the Hamiltonian dynamics, which results in distant samples; resampling momentum \( r \) is usually adopted.

Demonstration of proposing samples via Hamiltonian dynamics
HMC simulates the Hamiltonian dynamics to propose samples,

\[ d\theta = M^{-1}r \, dt \]  \hspace{1cm} (7) \\
\[ dr = -\nabla U(\theta) \, dt \]  \hspace{1cm} (8)

\( \pi(\theta, r) \) is invariant.

For example, \( H(\theta, r) = U(\theta) + K(r) \), \( U(\theta) = \frac{\theta^2}{2} \), \( K(r) = \frac{r^2}{2} \).

Demonstration of proposing samples via Hamiltonian dynamics
Hamiltonian Monte Carlo

In practice, $\epsilon$-discretization is used to simulate Hamiltonian dynamics, e.g., the leapfrog method [4]; MH is introduced to correct the discretization error.

Algorithm 1: Hamiltonian Monte Carlo

**Input:** Starting position $\theta^{(1)}$ and step size $\epsilon$

for $t = 1, 2 \cdots$ do

  Resample momentum $r$
  $r^{(t)} \sim \mathcal{N}(0, M)$
  $(\theta_0, r_0) = (\theta^{(t)}, r^{(t)})$

Simulate discretization of Hamiltonian dynamics in Eq. (4):

  $r_0 \leftarrow r_0 - \frac{\epsilon}{2} \nabla U(\theta_0)$

  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

  $r_m \leftarrow r_m - \frac{\epsilon}{2} \nabla U(\theta_m)$

  $(\hat{\theta}, \hat{r}) = (\theta_m, r_m)$

**Metropolis-Hastings correction:**

  $u \sim \text{Uniform}[0, 1]$

  $\rho = e^{H(\hat{\theta}, \hat{r}) - H(\theta^{(t)}, r^{(t)})}$

  if $u < \text{min}(1, \rho)$, then $\theta^{(t+1)} = \hat{\theta}$

end
Naive Stochastic Gradient HMC

The computation bottleneck is $\nabla U(\theta)$, requiring a pass over the entire dataset $\mathcal{D}$. One naive stochastic version is examining only a minibach $\tilde{\mathcal{D}}$.

$$\nabla \tilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|} \sum_{x \in \tilde{\mathcal{D}}} \nabla \log p(x|\theta) - \nabla \log p(\theta), \tilde{\mathcal{D}} \subset \mathcal{D}. \quad (9)$$

By central limit theorem,

$$\nabla \tilde{U}(\theta) \approx \nabla U(\theta) + \mathcal{N}(0, V(\theta)), \quad (10)$$

where $V(\theta)$ is the covariance of the stochastic gradient noise.
The computation bottleneck is $\nabla U(\theta)$, requiring a pass over the entire dataset $\mathcal{D}$. One naive stochastic version is examining only a minibach $\tilde{\mathcal{D}}$. 

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

(9)

By central limit theorem, 

$$
\nabla \tilde{U}(\theta) \approx \nabla U(\theta) + \mathcal{N}(0, V(\theta)),
$$

(10)

where $V(\theta)$ is the covariance of the stochastic gradient noise. Using $\epsilon$-discretization, the noisy Hamiltonian dynamics becomes,

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

(11)

$$
dr = -\nabla U(\theta) dt + \mathcal{N}(0, 2B(\theta) dt), 
$$

(12)

where $B(\theta) = \frac{1}{2}\epsilon V(\theta)$. 
Naive Stochastic Gradient HMC

**Theorem 1**

Let \( p_t(\theta, r) \) be the distribution of \((\theta, r)\) at time \( t \) with dynamics governed by the noisy Hamiltonian dynamics. Then the entropy \( h(p_t) \) increases with \( t \):

\[
\partial_t h(p_t) \geq 0.
\]

Simulation from \( U(\theta) = \frac{1}{2} \theta^2, \nabla \tilde{U}(\theta) = \theta + N(0, 4), \epsilon = 0.1 \)
Corollary 1

The distribution \( \pi(\theta, r) \propto \exp(-H(\theta, r)) \) is no longer invariant under the noisy Hamiltonian dynamics.

From the proof of the theorem, the noise-free Hamiltonian dynamics preserve entropy, while the additive noise term strictly increases entropy.
Corollary 1

The distribution $\pi(\theta, r) \propto \exp(-H(\theta, r))$ is no longer invariant under the noisy Hamiltonian dynamics.

From the proof of the theorem, the noise-free Hamiltonian dynamics preserve entropy, while the additive noise term strictly increases entropy. A straightforward solution: add a “friction” term to the dynamics.

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

Intuitively, the friction term $BM^{-1} r$ decreases the energy $H(\theta, r)$, thus reducing the influence of the noise.
Corollary 1

The distribution $\pi(\theta, r) \propto \exp(-H(\theta, r))$ is no longer invariant under the noisy Hamiltonian dynamics.

From the proof of the theorem, the noise-free Hamiltonian dynamics preserve entropy, while the additive noise term strictly increases entropy. A straightforward solution: add a “friction” term to the dynamics.

\[
\begin{align*}
  d\theta &= M^{-1}r dt \\
  dr &= -\nabla U(\theta) dt - BM^{-1}r dt + N(0, 2B dt)
\end{align*}
\]

Intuitively, the friction term $BM^{-1}r$ decreases the energy $H(\theta, r)$, thus reducing the influence of the noise.

Theorem 2

$\pi(\theta, r) \propto \exp(-H(\theta, r))$ is the unique stationary distribution of the dynamics described by (13) and (14).
Connection to first-order Langevin dynamics

The general Langevin equation is

\[
M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2K_b T B^{\frac{1}{2}}} R(t) \tag{15}
\]

\[
\langle R(t) \rangle = 0, \quad \langle R(t) R(t')^T \rangle = \delta(t - t') I.
\]

Letting \(K_b T = 1\), we have the dynamics specified by SGHMC in (13) and (14),

\[
M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2} B^{\frac{1}{2}} R(t) \tag{16}
\]

Letting the mass \(M\) go to 0, we get the first-order Langevin dynamics ("overdamped" Langevin dynamics),

\[
d\theta/dt = -B^{-1} \nabla U(\theta) + \sqrt{2} B^{-\frac{1}{2}} R(t) \tag{17}
\]

(17) corresponds to the SGLD in [1].
Connection to first-order Langevin dynamics

The general Langevin equation is

\[ M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2K_bT} B^{1/2} R(t) \]  \hfill (15)

\[ \langle R(t) \rangle = 0, \quad \langle R(t) R(t')^T \rangle = \delta(t - t') I. \]

Letting \( K_b = \frac{1}{T} \), we have the dynamics specified by SGHMC in (13) and (14),

\[ M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2}B^{1/2} R(t) \]  \hfill (16)
Connection to first-order Langevin dynamics

The general Langevin equation is

\[ M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2K_b T} B^{\frac{1}{2}} R(t) \]  

(15)

\[ \langle R(t) \rangle = 0, \langle R(t) R(t')^T \rangle = \delta(t-t') I. \]

Letting \( K_b = \frac{1}{T} \), we have the dynamics specified by SGHMC in (13) and (14),

\[ M \frac{d^2 \theta}{dt^2} = -\nabla U(\theta) - B \frac{d\theta}{dt} + \sqrt{2B} B^{\frac{1}{2}} R(t) \]  

(16)

Letting the mass \( M \) go to 0, we get the first-order Langevin dynamics (“overdamped” Langevin dynamics),

\[ \frac{d\theta}{dt} = -B^{-1} \nabla U(\theta) + \sqrt{2B}^{-\frac{1}{2}} R(t) \]  

(17)

(17) corresponds to the SGLD in [1].
Connection to first-order Langevin dynamics

Contrasting sampling of a bivariate Gaussian with correlation using SGHMC versus SGLD. Here, $U(\theta) = \frac{1}{2} \theta^T \Sigma^{-1} \theta$, $\nabla \tilde{U}(\theta) = \Sigma^{-1} \theta + \mathcal{N}(0, I)$ with $\Sigma_{11} = \Sigma_{22} = 1$ and correlation $\rho = \Sigma_{12} = 0.9$.

Left: Mean absolute error of the covariance estimation using ten million samples versus autocorrelation time of the samples as a function of 5 step size settings. Right: First 50 samples of SGHMC and SGLD.
In practice, \( \theta = M^{-1}r dt \)
\[ d\theta = M^{-1}r dt \]
\[ dr = -\nabla U(\theta) dt - BM^{-1}r dt + \mathcal{N}(0, 2B dt) \]
SGHMC in practice

\[
d\theta = M^{-1}r dt \\
dr = -\nabla U(\theta) dt - BM^{-1}r dt + \mathcal{N}(0, 2B dt)
\]

But in practice, \( B \) is not available, and we need to specify the friction term \( C \succeq B \), and consider the following dynamics,

\[
d\theta = M^{-1}r dt, \quad \text{(18)} \\
dr = -\nabla \tilde{U}(\theta) dt - CM^{-1}r dt + \mathcal{N}(0, 2(C - \hat{B}) dt) \\
= -\nabla U(\theta) dt - CM^{-1}r dt + \mathcal{N}(0, 2(C - \hat{B}) dt) + \mathcal{N}(0, 2B dt). \quad \text{(19)}
\]
SGHMC in practice

\[
\begin{align*}
    d\theta &= M^{-1}rdt \\
    dr &= -\nabla U(\theta)dt - BM^{-1}rdt + \mathcal{N}(0, 2Bdt)
\end{align*}
\]

But in practice, \( B \) is not available, and we need to specify the friction term \( C \geq B \), and consider the following dynamics,

\[
\begin{align*}
    d\theta &= M^{-1}rdt, \quad (18) \\
    dr &= -\nabla \tilde{U}(\theta)dt - CM^{-1}rdt + \mathcal{N}(0, 2(C - \tilde{B})dt) \\
    &= -\nabla U(\theta)dt - CM^{-1}rdt + \mathcal{N}(0, 2(C - \tilde{B})dt) + \mathcal{N}(0, 2Bdt). \quad (19)
\end{align*}
\]

Proposition 1

If \( \hat{B} = B \), then the dynamics of (18) and (19) yield the stationary distribution of 
\( \pi(\theta, r) \propto \exp(-H(\theta, r)) \).
SGHMC in practice

\[ d\theta = M^{-1}rdt \]
\[ dr = -\nabla U(\theta)dt - BM^{-1}rdt + \mathcal{N}(0, 2Bdt) \]

But in practice, \( B \) is not available, and we need to specify the friction term \( C \preceq B \), and consider the following dynamics,

\[ d\theta = M^{-1}rdt, \quad (18) \]
\[ dr = -\nabla \tilde{U}(\theta)dt - CM^{-1}rdt + \mathcal{N}(0, 2(C - \hat{B})dt) \]
\[ = -\nabla U(\theta)dt - CM^{-1}rdt + \mathcal{N}(0, 2(C - \hat{B})dt) + \mathcal{N}(0, 2Bdt). \quad (19) \]

Proposition 1

If \( \hat{B} = B \), then the dynamics of (18) and (19) yield the stationary distribution of \( \pi(\theta, r) \propto \exp(-H(\theta, r)) \).

Choice of \( \hat{B} \):

- simply setting \( \hat{B} = 0 \), since \( \hat{B} = \mathcal{O}(\epsilon) \); as \( \epsilon \to 0 \), \( C \) dominates;
- letting \( \hat{B} = \frac{1}{2}\epsilon\hat{V} \), where \( \hat{V} \) is the empirical Fisher information [2].
SGHMC in practice

User specified parameters:
\( \mathbf{M}, \mathbf{C}, \) the form of \( \hat{\mathbf{B}}, \) minibatch size, and discretization \( \epsilon_t. \)
Assuming \( \epsilon_t \to 0, \) no M-H step is adopted.

**Algorithm 2: Stochastic Gradient HMC**

```plaintext
for \( t = 1, 2 \cdots \) do
    optionally, resample momentum \( r \) as
    \( r^{(t)} \sim \mathcal{N}(0, \mathbf{M}) \)
    \( (\theta_0, r_0) = (\theta^{(t)}, r^{(t)}) \)
    simulate dynamics in Eq. (13):
    for \( i = 1 \) to \( m \) do
        \( \theta_i \leftarrow \theta_{i-1} + \epsilon_t \mathbf{M}^{-1} r_{i-1} \)
        \( r_i \leftarrow r_{i-1} - \epsilon_t \nabla \widetilde{U}(\theta_i) - \epsilon_t \mathbf{C}^{-1} \mathbf{r}_{i-1} + \mathcal{N}(0, 2(\mathbf{C} - \hat{\mathbf{B}})\epsilon_t) \)
    end
    \( (\theta^{(t+1)}, r^{(t+1)}) = (\theta_m, r_m) \), no M-H step
end
```
For clarity, the comparison is illustrated in the 1-D scenario. The SGD with momentum method [3]:

\[
\Delta \theta = r, \\
\Delta r = -\eta \nabla \tilde{f}(\theta) - (1 - \mu) r, 
\]

where \( \tilde{f}(\theta) \) is the stochastic version of \( f(\theta) \), \( \eta \) is the learning rate, and \( \mu \) is the momentum coefficient.
For clarity, the comparison is illustrated in the 1-D scenario. The SGD with momentum method [3]:

\[
\Delta \theta = r, \\
\Delta r = -\eta \nabla \tilde{f}(\theta) - (1 - \mu) r, 
\]

where \( \tilde{f}(\theta) \) is the stochastic version of \( f(\theta) \), \( \eta \) is the learning rate, and \( \mu \) is the momentum coefficient.

The updates in SGHMC can be rewritten as

\[
\Delta \theta = r, \\
\Delta r = -\epsilon^2 M^{-1} \nabla \tilde{U}(\theta) - \epsilon M^{-1} Cr + \mathcal{N}(0, 2\epsilon^3 M^{-1}(C - \hat{B})M^{-1}).
\]

changing notation by defining \( \eta = \epsilon^2 M^{-1} \), \( \alpha = \epsilon M^{-1} C \), \( \hat{\beta} = \epsilon M^{-1} \hat{B} \), we have

\[
\Delta \theta = r, \\
\Delta r = -\eta \nabla \tilde{U}(\theta) - \alpha r + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta).
\]
Connection to SGD with momentum

For clarity, the comparison is illustrated in the 1-D scenario. The SGD with momentum method [3]:

\[
\begin{align*}
\Delta \theta &= r, \\
\Delta r &= -\eta \nabla \tilde{f}(\theta) - (1 - \mu)r,
\end{align*}
\]

where \( \tilde{f}(\theta) \) is the stochastic version of \( f(\theta) \), \( \eta \) is the learning rate, and \( \mu \) is the momentum coefficient.

The updates in SGHMC can be rewritten as

\[
\begin{align*}
\Delta \theta &= r, \\
\Delta r &= -\epsilon^2 M^{-1} \nabla \tilde{U}(\theta) - \epsilon M^{-1} Cr + \mathcal{N}(0, 2\epsilon^3 M^{-1}(C - \hat{B}) M^{-1}).
\end{align*}
\]

changing notation by defining \( \eta = \epsilon^2 M^{-1}, \alpha = \epsilon M^{-1} C, \hat{\beta} = \epsilon M^{-1} \hat{B} \), we have

\[
\begin{align*}
\Delta \theta &= r, \\
\Delta r &= -\eta \nabla \tilde{U}(\theta) - \alpha r + \mathcal{N}(0, 2(\alpha - \hat{\beta}) \eta).
\end{align*}
\]

If \( C = \hat{B} \), SGHMC reduces to SGD with momentum.
Denote the quantities:

- \(d\): parameter dimension of \(\theta\)
- \(|\tilde{D}|\): mini-batch size
Denote the quantities:

- \( d \): parameter dimension of \( \theta \)
- \( |\tilde{D}| \): mini-batch size

At each iteration, we need to

- estimate \( \nabla \tilde{U}(\theta) \): \( m(|\tilde{D}|, d) \),
- estimate \( \hat{B}(\theta) \): \( O(d^2|\tilde{D}|) \),
- simulate from \( \mathcal{N}(0, 2(\mathbf{C} - \hat{B})\epsilon) \): \( O(d^3) \).

In total, \( O(d^2|\tilde{D}| + d^3 + m(|\tilde{D}|, d)) \).
Denote the quantities:
- $d$: parameter dimension of $\theta$
- $|\tilde{D}|$: mini-batch size

At each iteration, we need to
- estimate $\nabla \tilde{U}(\theta)$: $m(|\tilde{D}|, d)$,
- estimate $\hat{B}(\theta)$: $O(d^2|\tilde{D}|)$,
- simulate from $\mathcal{N}(\mathbf{0}, 2(\mathbf{C} - \hat{\mathbf{B}})\epsilon)$: $O(d^3)$.

In total, $O(d^2|\tilde{D}| + d^3 + m(|\tilde{D}|, d))$.

If using diagonal matrices for $\mathbf{C}$ and $\hat{\mathbf{B}}$,
- complexity reduces to $O(d|\tilde{D}| + d + m(|\tilde{D}|, d))$. 
Experiment I: Bayesian neural network

Model:

\[
p(y = i | \mathbf{x}) \propto \exp(\mathbf{A}_i^T \sigma(\mathbf{B}^T \mathbf{x} + \mathbf{b}) + \mathbf{a}_i) \tag{24}
\]

\[
p(\mathbf{A}) \propto \exp(-\lambda_A \| \mathbf{A} \|^2) \tag{25}
\]

\[
p(\mathbf{B}) \propto \exp(-\lambda_B \| \mathbf{B} \|^2) \tag{26}
\]

\[
p(\mathbf{a}) \propto \exp(-\lambda_a \| \mathbf{a} \|^2) \tag{27}
\]

\[
p(\mathbf{b}) \propto \exp(-\lambda_b \| \mathbf{b} \|^2) \tag{28}
\]

\[
\lambda_A, \lambda_B, \lambda_a, \lambda_b \sim Ga(1, 1) \tag{29}
\]

Sampling:

- sample \( p(\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b} | \lambda_A, \lambda_B, \lambda_a, \lambda_b, \mathcal{D}) \) using SGHMC or SGLD, with minibatch of 500 instances. Sample for 100 steps before updating hyper-parameters;

- sample \( p(\lambda_A, \lambda_B, \lambda_a, \lambda_b | \mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}) \) via Gibbs.
Experiment I: Bayesian neural network

Setup: MNIST, consisting of 60,000 training (10,000 used for validation) and 10,000 testing instances. $\mathbf{x} \in \mathbb{R}^{784}$, and 100 hidden units are used.

Sampling: 800 iterations with 50 as burn-in. Note that each iteration is over the entire training set.

Convergence of test error on MNIST, using SGD, SGD with momentum, SGLD, and SGHMC

![Graph showing convergence of test error on MNIST](image)

The results are shown in Table 1. Both SGHMC and SGLD are very similar. We also observed that the per-iteration running time of both methods are comparable. As such, the per-iteration running time of both methods are comparable. As such, the per-iteration running time of both methods are comparable. As such, the per-iteration running time of both methods are comparable.
Experiment II: Probablistic matrix factorization

Model:

\[ Y_{ij} \mid U, V \sim \mathcal{N}(U_i^T V_j + a_i + b_j, \tau^{-1}) \]  

(30)

\[ U_{ki} \sim \mathcal{N}(0, \lambda_U^{-1}) \]  

(31)

\[ V_{kj} \sim \mathcal{N}(0, \lambda_V^{-1}) \]  

(32)

\[ a_i \sim \mathcal{N}(0, \lambda_a^{-1}) \]  

(33)

\[ b_j \sim \mathcal{N}(0, \lambda_b^{-1}) \]  

(34)

\[ \lambda_U, \lambda_V, \lambda_a, \lambda_b \sim Ga(1, 1) \]  

(35)

Sampling:

- sample \( p(U, V, a, b \mid \lambda_U, \lambda_V, \lambda_a, \lambda_b, D) \) using SGHMC or SGLD, with minibatch of 4000 ratings. Sample for 2000 steps before updating hyper-parameters;

- sample \( p(\lambda_U, \lambda_V, \lambda_a, \lambda_b \mid U, V, a, b) \) via Gibbs.

Wenzhao Lian (Presenter)
Setup: Movielens dataset ml-1M\(^5\), consisting of about 1 million ratings of 3592 movies by 6040 users. Latent dimension is 20.
Sampling: 2,000,000 iterations with 100,000 as burn-in.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>0.8538 ± 0.0009</td>
</tr>
<tr>
<td>SGD WITH MOMENTUM</td>
<td>0.8539 ± 0.0009</td>
</tr>
<tr>
<td>SGLD</td>
<td>0.8412 ± 0.0009</td>
</tr>
<tr>
<td>SGHMC</td>
<td>0.8411 ± 0.0011</td>
</tr>
</tbody>
</table>

Predictive RMSE on the Movielens dataset using 5-fold cross validation.

