Markov Chain Monte Carlo (MCMC)

Dmitry Kropotov

Lomonosov Moscow State University

Contents

Ising model

Gibbs sampling

Metropolis-Hastings sampling

Hamiltonian Monte Carlo

Langevin Monte Carlo

Recap: Variational inference

Probabilistic model: $p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\theta})$.

Goal: find approximant $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ for posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$.

Method: $\log p(\boldsymbol{x}) \geq \mathrm{ELBO}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) - \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})} \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\lambda})}{p(\boldsymbol{\theta})} \to \max_{\boldsymbol{\lambda}}$.

Recap: Variational inference

Probabilistic model: $p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\theta})$.

Goal: find approximant $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ for posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$.

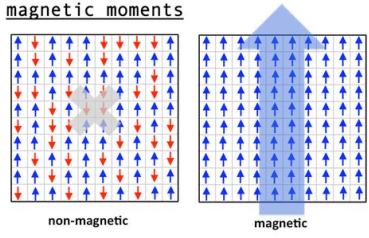
Method: $\log p(\boldsymbol{x}) \geq \mathrm{ELBO}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) - \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})} \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\lambda})}{p(\boldsymbol{\theta})} \to \max_{\boldsymbol{\lambda}}$.

Problem: $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ could be quite poor approximant for the true posterior!

Ising model

Ising model – a model from statistical physics describing magnetic properties of ferromagnetic solids depending on temperature. It reflects phase transition effect – a drastic magnetization changing within narrow temperature interval.

- Consider 2-dimensional atomic lattice.
- $x_i \in \{-1, +1\}$ – magnetic moment for one atom.
- Energy $E(\boldsymbol{x}) = -\sum_{(i,j)\in\mathcal{E}} x_i x_j - \sum_{i=1}^{n} h_i x_i$.
- Boltzmann probability distribution $p(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-\frac{1}{T}E(\boldsymbol{x})\right)$, where T – temperature, $Z = \sum_{\boldsymbol{x}} \exp\left(-\frac{1}{T}E(\boldsymbol{x})\right)$ – normalizing constant.



Ising model

Ising model:

$$E(\boldsymbol{x}) = -\sum_{(i,j)\in\mathcal{E}} x_i x_j - \sum_{i=1}^{n} h_i x_i,$$

$$p(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-\frac{1}{T}E(\boldsymbol{x})\right).$$

Would like to estimate for each temperature:

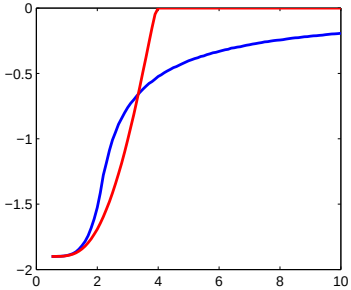
- Mean energy $\mathbb{E}_{p(\boldsymbol{x})}E(\boldsymbol{x})$;
- Energy variance $\mathbb{D}_{p(\boldsymbol{x})}E(\boldsymbol{x})$;
- Mean magnetization $\sqrt{\mathbb{E}_{p(\boldsymbol{x})}\mu^2}$, where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Key challenge: normalizing constant Z is a sum over 2^n components.

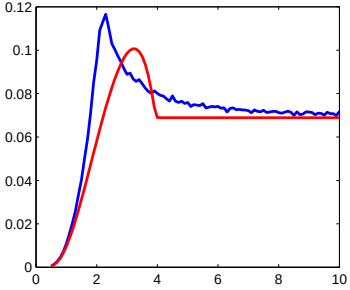
Variational inference for Ising model

Mean field approximation: $p(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-\frac{1}{T}E(\boldsymbol{x})\right) \approx q(\boldsymbol{x}) = \prod_{i=1}^{n} q_i(x_i)$.

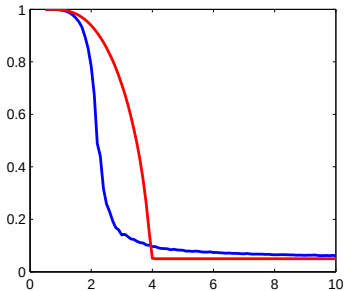
Recalculation: $q_i(x_i = 1) = 1/(1 + \exp(-(2/T)(h_i + \sum_{j:(i,j)\in\mathcal{E}} \mathbb{E}_{q_j} x_j))$.



$\mathbb{E}_{q(\boldsymbol{x})}E(\boldsymbol{x})$



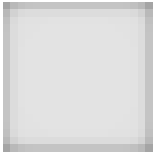
$\mathbb{D}_{q(\boldsymbol{x})}E(\boldsymbol{x})$



$\sqrt{\mathbb{E}_{q(\boldsymbol{x})}\mu^2}$

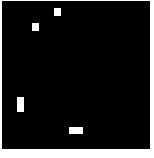
Variational inference for Ising model

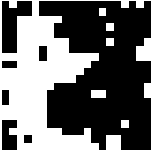
Variational approximations:



Low temperature | Critical temperature | High temperature

Samples from true posterior:





Low temperature | Critical temperature | High temperature

MCMC

Probabilistic model: $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$, $Z = \int \tilde{p}(\boldsymbol{x})d\boldsymbol{x}$.

MCMC methods construct samples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ from $p(\boldsymbol{x})$ using only $\tilde{p}(\boldsymbol{x})$.

Usage: $\mathbb{E}_{p(\boldsymbol{x})} f(\boldsymbol{x}) = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{x}_i)$.

MCMC

MCMC generate samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ using Markov Chain with some transition probability $q(\boldsymbol{x}|\boldsymbol{y})$ in the following way:

- Generate $\boldsymbol{x}_1$ from some initial distribution $p_0(\boldsymbol{x})$;
- Generate $\boldsymbol{x}_2$ from $q(\boldsymbol{x}|\boldsymbol{x}_1)$;
- Generate $\boldsymbol{x}_3$ from $q(\boldsymbol{x}|\boldsymbol{x}_2)$;
- etc.

Samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ are not independent, but still can be used for estimation $\mathbb{E}_{p(\boldsymbol{x})} f(\boldsymbol{x})$.

Markov chain properties

Consider a Markov chain with transition probability $q(\boldsymbol{x}|\boldsymbol{y})$.

Probability distribution $\pi(\boldsymbol{x})$ is called invariant under given Markov chain iff

$$\int q(\boldsymbol{x}|\boldsymbol{y})\pi(\boldsymbol{y})d\boldsymbol{y} = \pi(\boldsymbol{x}).$$

Consider initial distribution $p_0(\boldsymbol{x})$ and denote $p_i(\boldsymbol{x})$ – distribution of points after i steps of given Markov chain. Then the Markov chain is called ergodic iff

$$p_i(\boldsymbol{x}) \to \pi(\boldsymbol{x}),\ i \to \infty,\ \forall p_0(\boldsymbol{x}),$$

where $\pi(\boldsymbol{x})$ is invariant distribution for this chain.

For generating samples from distribution $p(\boldsymbol{x})$ using MCMC, the corresponding Markov chain should be ergodic with invariant distribution $p(\boldsymbol{x})$.

Gibbs sampling

Goal: generate samples from $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$, where $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k]$.

Suppose $\boldsymbol{x} \sim p(\boldsymbol{x})$. Then the next point $\boldsymbol{x}^{new}$ is generated as follows:

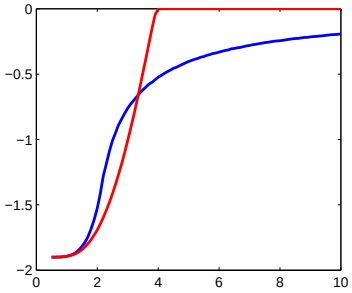
- $\boldsymbol{x}_1^{new} \sim p(\boldsymbol{x}_1 | \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_k)$;
- $\boldsymbol{x}_2^{new} \sim p(\boldsymbol{x}_2 | \boldsymbol{x}_1^{new}, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_k)$;
- $\ldots$;
- $\boldsymbol{x}_k^{new} \sim p(\boldsymbol{x}_k | \boldsymbol{x}_1^{new}, \boldsymbol{x}_3^{new}, \ldots, \boldsymbol{x}_{k-1}^{new})$;
- $\boldsymbol{x}^{new} = [\boldsymbol{x}_1^{new}, \boldsymbol{x}_2^{new}, \ldots, \boldsymbol{x}_k^{new}]$.

It is easy to show than $p(\boldsymbol{x})$ is invariant under such transition probability. If all conditionals $p(\boldsymbol{x}_i | \boldsymbol{x}_{\setminus i}) > 0$, then the corresponding Markov chain would be ergodic.

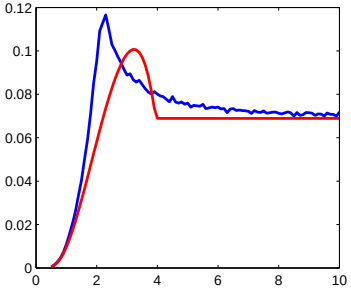
Gibbs sampling for Ising model

Ising model: $p(\boldsymbol{x}) = \frac{1}{Z} \exp\left(\frac{1}{T}\left(\sum_{i=1}^{n} h_i x_i + \sum_{(i,j)\in\mathcal{E}} x_i x_j\right)\right)$.

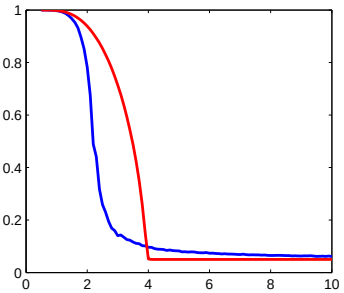
For Gibbs sampling we need to calculate all conditionals:
$p(x_i = 1 | \boldsymbol{x}_{\setminus i}) = 1/(1 + \exp(-(2/T)(h_i + \sum_{j:(i,j)\in\mathcal{E}} x_j)))$.



$\mathbb{E}_{q(\boldsymbol{x})} E(\boldsymbol{x})$



$\mathbb{D}_{q(\boldsymbol{x})} E(\boldsymbol{x})$



$\sqrt{\mathbb{E}_{q(\boldsymbol{x})} \mu^2}$

Gibbs sampling: conclusions

Gibbs sampling:

- Can be applied both for discrete and continuous variables;
- Doesn't have any parameters for tuning;
- Can be highly inefficient for large dimensional case.

Metropolis-Hastings sampling

Goal: generate samples from $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$.

Suppose we have some proposal distribution $r(\boldsymbol{x}|\boldsymbol{y})$, from which we can generate samples. Sampling scheme:

- Generate $\boldsymbol{x}_{trial} \sim r(\boldsymbol{x}|\boldsymbol{x}_{old})$;
- Calculate $A = \min\left(1, \frac{\tilde{p}(\boldsymbol{x}_{trial})r(\boldsymbol{x}_{old}|\boldsymbol{x}_{trial})}{\tilde{p}(\boldsymbol{x}_{old})r(\boldsymbol{x}_{trial}|\boldsymbol{x}_{old})}\right)$;
- $\boldsymbol{x}_{new} = \boldsymbol{x}_{trial}$ with probability A and $\boldsymbol{x}_{new} = \boldsymbol{x}_{old}$ with probability $1 - A$.

Metropolis sampling

Suppose that proposal is symmetric, i.e. $r(\boldsymbol{x}|\boldsymbol{y}) = r(\boldsymbol{y}|\boldsymbol{x})$. Then probability A is reduced to:

$$A = \min\left(1, \frac{\tilde{p}(\boldsymbol{x}_{trial})}{\tilde{p}(\boldsymbol{x}_{old})}\right).$$

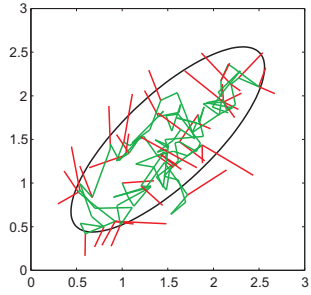
Sampling scheme:

- Generate $\boldsymbol{x}_{trial} \sim r(\boldsymbol{x}|\boldsymbol{x}_{old})$;
- If $\tilde{p}(\boldsymbol{x}_{trial}) \geq \tilde{p}(\boldsymbol{x}_{old})$, then $\boldsymbol{x}_{new} = \boldsymbol{x}_{trial}$;
- Else $\boldsymbol{x}_{new} = \boldsymbol{x}_{trial}$ with probability $\tilde{p}(\boldsymbol{x}_{trial})/\tilde{p}(\boldsymbol{x}_{old})$ and $\boldsymbol{x}_{new} = \boldsymbol{x}_{old}$ otherwise.

Metropolis sampling

Suppose we would like to sample from correlated 2-dimensional Gaussian distribution using MH sampling with the following symmetric proposal:

$$r(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{y}, \sigma^2 I).$$



For poor proposals MH sampling shows random walk behaviour!

Hamiltonian equations

Suppose we have a particle moving in some potential field.

- $\boldsymbol{x}$ – particle coordinates; $\boldsymbol{p}$ – particle momentum, $\boldsymbol{p} = m\frac{d\boldsymbol{x}}{dt}$;
- $U(\boldsymbol{x})$ – potential energy; $K(\boldsymbol{p}) = \boldsymbol{p}^T\boldsymbol{p}/m$ – kinetic energy;
- $H(\boldsymbol{x}, \boldsymbol{p}) = U(\boldsymbol{x}) + K(\boldsymbol{p})$ – Hamiltonian.

Hamiltonian equations:

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i},$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}.$$

Properties of Hamiltonian dynamics

Hamiltonian conservation:

$$\frac{dH(\boldsymbol{x},\boldsymbol{p})}{dt} = \sum_i \left[\frac{\partial H}{\partial x_i}\frac{dx_i(t)}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i(t)}{dt}\right] = \sum_i \left[\frac{\partial H}{\partial x_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial x_i}\right] = 0.$$

Reversibility:

$$[\boldsymbol{x}(t+s), \boldsymbol{p}(t+s)] = HamDyn(\boldsymbol{x}(t), \boldsymbol{p}(t), s),$$
$$[\boldsymbol{x}(t), \boldsymbol{p}(t)] = HamDyn(\boldsymbol{x}(t+s), \boldsymbol{p}(t+s), -s) = HamDyn(\boldsymbol{x}(t+s), -\boldsymbol{p}(t+s), s).$$

Hamiltonian Monte Carlo (HMC)

Goal: generate samples from $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$.

Let's introduce the following probabilistic model:

$$p(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{Z}\exp(-H(\boldsymbol{x}, \boldsymbol{p})) = \frac{1}{Z}\exp(-U(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{p}^T\boldsymbol{p})$$

$$U(\boldsymbol{x}) = -\log\tilde{p}(\boldsymbol{x}),$$

$\boldsymbol{p}$ – auxiliary variable.

Sample $(\boldsymbol{x}_1, \boldsymbol{p}_1), \ldots, (\boldsymbol{x}_m, \boldsymbol{p}_m)$ from $p(\boldsymbol{x}, \boldsymbol{p})$ and then discard all $\boldsymbol{p}_i$.

Hamiltonian Monte Carlo (HMC)

Probabilistic model:

$$p(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{Z} \exp(-H(\boldsymbol{x}, \boldsymbol{p})) = \frac{1}{Z} \exp(-U(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{p}^T \boldsymbol{p})$$

$$U(\boldsymbol{x}) = -\log \tilde{p}(\boldsymbol{x}),$$

$\boldsymbol{p}$ – auxiliary variable.

HMC generation scheme:

- Generate $\boldsymbol{p}$ from $p(\boldsymbol{p}|\boldsymbol{x}_{old}) = \frac{1}{Z} \exp(-(1/2)\boldsymbol{p}^T \boldsymbol{p})$;
- Solve Hamiltonian equation for some time interval starting from $(\boldsymbol{x}_{old}, \boldsymbol{p})$ and obtain $(\boldsymbol{x}_{new}, \boldsymbol{p}_{new})$;
- Due to reversibility property introduced proposal is symmetric. Accept the new point $\boldsymbol{x}_{new}$ with probability $\min(1, \exp(-H(\boldsymbol{x}_{new}, \boldsymbol{p}_{new}) + H(\boldsymbol{x}_{old}, \boldsymbol{p})))$.

With ideal solving of Hamiltonian equation the acceptance probability is always 1 due to Hamiltonian conservation property. In practice we have here a discretization error.

Solving Hamiltonian equations

Hamiltonian equations:

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i},$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}.$$

Euler's method:

$$p_i(t+\varepsilon) = p_i(t) + \varepsilon \frac{dp_i(t)}{dt} = p_i(t) - \varepsilon \frac{\partial U(\boldsymbol{x}(t))}{\partial x_i};$$
$$x_i(t+\varepsilon) = x_i(t) + \varepsilon \frac{dx_i(t)}{dt} = x_i(t) + \varepsilon p_i(t).$$

- Local discretization error: $O(\varepsilon^2)$;
- Global discretization error: $O(T\varepsilon)$.

Solving Hamiltonian equations

Hamiltonian equations:

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i},$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}.$$

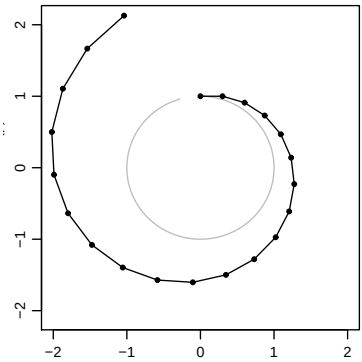
Leapfrog method:

$$p_i(t + \varepsilon/2) = p_i(t) - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t))}{\partial x_i};$$
$$x_i(t + \varepsilon) = x_i(t) + \varepsilon p_i(t + \varepsilon/2);$$
$$p_i(t + \varepsilon) = p_i(t + \varepsilon/2) - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t + \varepsilon))}{\partial x_i}.$$

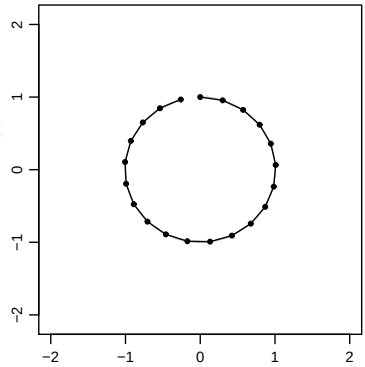
- Local discretization error: $O(\varepsilon^3)$;
- Global discretization error: $O(T\varepsilon^2)$.

Leapfrog method has reversibility property.

Solving Hamiltonian equations



Euler's method



Leapfrog

Leapfrog method

Initial formulation:

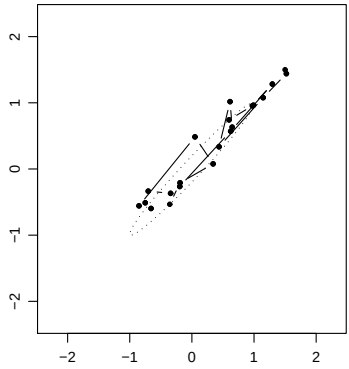
$$
\begin{aligned}
p_i(t+\varepsilon/2) &= p_i(t) - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t))}{\partial x_i};\\
x_i(t+\varepsilon) &= x_i(t) + \varepsilon p_i(t+\varepsilon/2);\\
p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t+\varepsilon))}{\partial x_i}.
\end{aligned}
$$

Equivalent formulation:

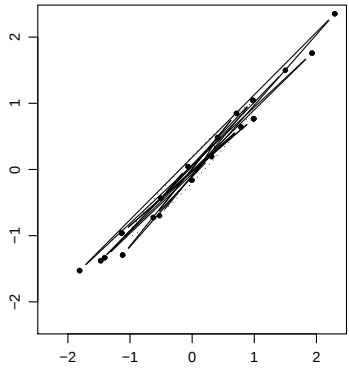
$$
\begin{aligned}
x_i(t+\varepsilon) &= x_i(t) - \frac{\varepsilon^2}{2}\frac{\partial U(\boldsymbol{x}(t))}{\partial x_i} + \varepsilon p_i(t);\\
p_i(t+\varepsilon) &= p_i(t) - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t))}{\partial x_i} - (\varepsilon/2)\frac{\partial U(\boldsymbol{x}(t+\varepsilon))}{\partial x_i}.
\end{aligned}
$$

Coordinate update = gradient descent step + noise.

HMC Example



Random walk Metropolis



HMC

Tuning HMC

- Initial HMC with high density point (possibly using optimization method);
- Consider rejection rate in interval $[1/4, 3/4]$;
- For detecting "burn in" period consider several Markov chains initialized from different starting points with monitoring moving average of some scalar statistics;
- Monitor autocorrelation function and estimate effective sample size;
- etc.

Langevin Monte Carlo

Let's consider HMC with just one step in Leapfrog integration:

1. Sample $\boldsymbol{p}$ from standard Normal distribution;
2. Calculate $\boldsymbol{x}_{new} = \boldsymbol{x} - \frac{\varepsilon^2}{2}\nabla_{\boldsymbol{x}} U(\boldsymbol{x}) + \varepsilon \boldsymbol{p}$;
3. Calculate $\boldsymbol{p}_{new} = \boldsymbol{p} - (\varepsilon/2)\nabla_{\boldsymbol{x}} U(\boldsymbol{x}) - (\varepsilon/2)\nabla_{\boldsymbol{x}} U(\boldsymbol{x}_{new})$. Accept $\boldsymbol{x}_{new}$ with probability $\min(1, \exp(-U(\boldsymbol{x}_{new}) + U(\boldsymbol{x}) - K(\boldsymbol{p}_{new}) + K(\boldsymbol{p})))$.

Step 2 is equivalent to: $\boldsymbol{x}_{new} \sim \mathcal{N}(\boldsymbol{x} - \frac{\varepsilon^2}{2}\nabla_{\boldsymbol{x}} U(\boldsymbol{x}), \varepsilon^2 I)$.

Due to small ε Langevin Monte Carlo sometimes is used without accept/rejection step.

Stochastic case

Probabilistic model: $p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\theta})$.

Stochastic optimization for MAP:
$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} + \alpha[\nabla_\theta \log p(\boldsymbol{\theta}) + \frac{n}{n_{batch}} \sum_{i \in batch} \nabla_\theta \log p(\boldsymbol{x}_i|\boldsymbol{\theta}_{old})]$.

Variational inference:
$\boldsymbol{\lambda}_{new} = \boldsymbol{\lambda}_{old} + \alpha[\frac{n}{n_{batch}} \sum_{i \in batch} \nabla_\lambda \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) - \nabla_\lambda KL(q(\boldsymbol{\theta}|\boldsymbol{\lambda})||p(\theta))$.

HMC sampling: use stochastic gradient for $\nabla_{\boldsymbol{x}} U(\boldsymbol{x})$ in leapfrog integration. However, rejection probability requires calculation of full likelihood:

$$A = \min(1, \exp(-H(\boldsymbol{x}_{new}, \boldsymbol{p}_{new}) + H(\boldsymbol{x}_{old}, \boldsymbol{p})).$$

In Langevin Monte Carlo we don't need to use rejection. Hence, it is easy to come to stochastic version.

Conclusion

- MCMC algorithms are able to sample from desired distributions without approximations;
- Due to big amount of sampling they are usually slower comparing to Variational Inference;
- MCMC and Variational Inference can be used together on correspondingly testing and training phase;
- Many MCMC algorithm require careful tuning for efficient sampling;
- There exist scalable MCMC algorithms. See the next lecture.