Lecture 4: Markov Chain Monte Carlo

Cheng Zhang
School of Mathematical Sciences, Peking University

October 29, 2020
Limitations of Monte Carlo

- Direct sampling in high-dimensional spaces is often infeasible, very hard to get rare events
- Rejection sampling, Importance sampling
  - Do not work well if the proposal $q(x)$ is very different from $f(x)$ or $h(x)f(x)$.
  - Moreover, constructing appropriate $q(x)$ can be difficult. Making a good proposal usually requires knowledge of the analytic form of the target distribution - but if we had that, we wouldn’t even need to sample

- Intuition: instead of a fixed proposal $q(x)$, what if we use an adaptive proposal?
- In this lecture, we are going to talk about one of the most popular sampling methods, **Markov chain Monte Carlo**.
Stochastic processes is a family of random variables, usually indexed by a set of numbers (time). A discrete time stochastic process is simply a sequence of random variables, $X_0, X_1, \ldots, X_n$ defined on the same probability space.

One of the simplest stochastic processes (and one of the most useful) is the simple random walk.

Consider a simple random walk on a graph $G = (\Omega, E)$. The stochastic process starts from an initial position $X_0 = x_0 \in \Omega$, and proceeds following a simple rule:

$$p(X_{n+1} | X_n = x_n) \sim \text{Discrete}(\mathcal{N}(x_n)), \ \forall n \geq 0$$

where $\mathcal{N}(x_n)$ denotes the neighborhood of $x_n$. 
Two random walks on a $10 \times 10$ grid graph
The above simple random walk is a special case of another well-known stochastic process called Markov chains.

A Markov chain represents the stochastic movement of some particle in the state space over time. The particle initially starts from state $i$ with probability $\pi^{(0)}_i$, and after that moves from the current state $i$ at time $t$ to the next state $j$ with probability $p_{ij}(t)$.

A Markov chain has three main elements:

1. A state space $\mathcal{S}$
2. An initial distribution $\pi^{(0)}$ over $\mathcal{S}$
3. Transition probabilities $p_{ij}(t)$ which are non-negative numbers representing the probability of going from state $i$ to $j$, and $\sum_j p_{ij}(t) = 1$.

When $p_{ij}(t)$ does not depend on time $t$, we say the Markov chain is time-homegenous.
Markov Property

- **Chain rule (in probability)**

\[ p(X_n = x_n, \ldots, X_0 = x_0) = \prod_{i=1}^{n} p(X_i = x_i | X_{<i} = x_{<i}) \]

- **Markov property**

\[ p(X_{i+1} = x_{i+1} | X_i = x_i, \ldots, X_0 = x_0) = p(X_{i+1} = x_{i+1} | X_i = x_i) \]

- **Joint probability with Markov property**

\[ p(X_n = x_n, \ldots, X_0 = x_0) = \prod_{i=1}^{n} p(X_i = x_i | X_{i-1} = x_{i-1}) \]

fully determined by the transition probabilities
Consider the 2000 US presidential election with three candidates: Gore, Bush and Nader (just an illustrative example and does not reflect the reality of that election)

We assume that the initial distribution of votes (i.e., probability of winning) was $\pi = (0.49, 0.45, 0.06)$ for Gore, Bush and Nader respectively

Further, we assume the following transition probability matrix

<table>
<thead>
<tr>
<th></th>
<th>Gore</th>
<th>Bush</th>
<th>Nader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gore</td>
<td>0.94</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>Bush</td>
<td>0.05</td>
<td>0.95</td>
<td>0</td>
</tr>
<tr>
<td>Nader</td>
<td>0.05</td>
<td>0.01</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Example

A probabilistic graph presentation of the Markov chain
If we represent the transition probability a square matrix $P$ such that $P_{ij} = p_{ij}$, we can obtain the distribution of states in step $n$, $\pi^{(n)}$, as follows

$$\pi^{(n)} = \pi^{(n-1)} P = \ldots = \pi^{(0)} P^n$$

For the above example, we have

$$\pi^{(0)} = (0.4900, 0.4500, 0.0600)$$
$$\pi^{(10)} = (0.4656, 0.4655, 0.0689)$$
$$\pi^{(100)} = (0.4545, 0.4697, 0.0758)$$
$$\pi^{(200)} = (0.4545, 0.4697, 0.0758)$$
As we can see last, after several iterations, the above Markov chain converges to a distribution, 
\((0.4545, 0.4697, 0.0758)\)

In this example, the chain would have reached this distribution regardless of what initial distribution \(\pi^{(0)}\) we chose. Therefore, \(\pi = (0.4545, 0.4697, 0.0758)\) is the stationary distribution for the above Markov chain.

**Stationary distribution.** A distribution of Markov chain states is called to be stationary if it remains the same in the next time step, i.e.,

\[\pi = \pi P\]
How can we find out whether such distribution exists?

Even if such distribution exists, is it unique or not?

Also, how do we know whether the chain would converge to this distribution?

To find out the answer, we briefly discuss some properties of Markov chains
Irreducibility

- Irreducible: A Markov chain is **irreducible** if the chain can move from any state to another state.

- Examples
  - The simple random walk is irreducible
  - The following chain, however, is reducible since Nader does not communicate with the other two states (Gore and Bush)

<table>
<thead>
<tr>
<th></th>
<th>Gore</th>
<th>Bush</th>
<th>Nader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gore</td>
<td>0.95</td>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td>Bush</td>
<td>0.05</td>
<td>0.95</td>
<td>0</td>
</tr>
<tr>
<td>Nader</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Aperiodicity

- **Period**: the period of a state $i$ is the greatest common divisor of the times at which it is possible to move from $i$ to $i$.

- For example, all the states in the following Markov chain have period 3.

\[
\begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{pmatrix}
\]

- **Aperiodic**: a Markov chain is said to be aperiodic if the period of each state is 1, otherwise the chain is periodic.
Recurrent vs. Transient

- **Recurrent** states: a state $i$ is called recurrent if with probability 1, the chain would ever return to state $i$ given that it started in state $i$.

<table>
<thead>
<tr>
<th></th>
<th>Gore</th>
<th>Bush</th>
<th>Nader</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gore</strong></td>
<td>0.94</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Bush</strong></td>
<td>0.05</td>
<td>0.95</td>
<td>0</td>
</tr>
<tr>
<td><strong>Nader</strong></td>
<td>0.05</td>
<td>0.01</td>
<td>0.94</td>
</tr>
</tbody>
</table>

- **Positive recurrent**: a recurrent state $j$ is called positive recurrent if the expected amount of time to return to state $j$ given that the chain started in state $j$ is finite.

- For a positive recurrent Markov chain, the stationary distribution exists and is unique.
Reversibility: a Markov chain is said to be reversible with respect to a probability distribution $\pi$ if $\pi_ip_{ij} = \pi_jp_{ji}$

- In fact, if a Markov chain is reversible with respect to $\pi$, then $\pi$ is also a stationary distribution

$$
\sum_i \pi_ip_{ij} = \sum_i \pi_jp_{ji} \\
= \pi_j \sum_i p_{ji} \\
= \pi_j
$$

since $\sum_i p_{ji} = 1$ for all transition probability matrices

- This is also known as detailed balance condition
We can define a Markov chain on a general state space $\mathcal{X}$ with initial distribution $\pi^{(0)}$ and transition probabilities $p(x, A)$ defined as the probability of jumping to the subset $A$ from point $x \in \mathcal{X}$.

Similarly, with Markov property, we have the joint probability

$$p(X_0 \in A_0, \ldots, X_n \in A_n) = \int_{A_0} \pi^{(0)}(dx_0) \cdots \int_{A_n} p(x_{n-1}, dx_n)$$

Example. Consider a Markov chain with the real line as its state space. The initial distribution is $\mathcal{N}(0, 1)$, and the transition probability is $p(x, \cdot) = \mathcal{N}(x, 1)$. This is just a Brownian motion (observed at discrete time)
Unlike the discrete space, we now need to talk about the property of Markov chains with a continuous non-zero measure $\phi$, on $\mathcal{X}$, and use sets $A$ instead of points.

A chain is $\phi$-irreducible if for all $A \subseteq \mathcal{X}$ with $\phi(A) > 0$ and for all $x \in \mathcal{X}$, there exists a positive integer $n$ such that

$$p^n(x, A) = p(X_n \in A | X_0 = x) > 0$$

Similarly, we need to modify our definition of period.
A distribution $\pi$ is a stationary distribution if
\[ \pi(A) = \int_X \pi(dx)p(x, A), \quad \forall A \subseteq X \]

As for the discrete case, a continuous space Markov chain is reversible with respect to $\pi$ if
\[ \pi(dx)p(x, dy) = \pi(dy)p(y, dx) \]

Similarly, if the chain is reversible with respect to $\pi$, then $\pi$ is a stationary distribution.

Example. Consider a Markov chain on the real line with initial distribution $\mathcal{N}(1, 1)$ and transition probability $p(x, \cdot) = \mathcal{N}(\frac{x}{2}, \frac{3}{4})$. It is easy to show that the chain converges to $\mathcal{N}(0, 1)$ (Exercise)
Ergodicity

- **Ergodic:** a Markov chain is ergodic if it is both irreducible and aperiodic, with stationary distribution $\pi$

- **Ergodic Theorem.** For an ergodic Markov chain on the state space $\mathcal{X}$ having stationary distribution $\pi$, we have: (i) for all measurable $A \subseteq \mathcal{X}$ and $\pi$-a.e. $x \in \mathcal{X}$,

$$\lim_{t \to \infty} p_t(x, A) = \pi(A)$$

(ii) $\forall f$ with $\mathbb{E}_\pi |f(x)| < \infty$,

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} f(X_t) = \int_{\mathcal{X}} f(x)\pi(x)dx, \quad \text{a.s.}$$

In particular, $\pi$ is the **unique** stationary probability density function for the chain.
Now suppose we are interested in sampling from a distribution $\pi$ (e.g., the unnormalized posterior).

Markov chain Monte Carlo (MCMC) is a method that samples from a Markov chain whose stationary distribution is the target distribution $\pi$. It does this by constructing an appropriate transition probability for $\pi$.

MCMC, therefore, can be viewed as an inverse process of Markov chains.
The transition probability in MCMC resembles the proposal distribution we used in previous Monte Carlo methods.

Instead of using a fixed proposal (as in importance sampling and rejection sampling), MCMC algorithms feature adaptive proposals.

 Figures adapted from Eric Xing (CMU)
Suppose that we are interested in sampling from a distribution $\pi$, whose density we know up to a constant $P(x) \propto \pi(x)$

We can construct a Markov chain with a transition probability (i.e., proposal distribution) $Q(x'|x)$ which is symmetric; that is, $Q(x'|x) = Q(x|x')$

Example. A normal distribution with the mean at the current state and fixed variance $\sigma^2$ is symmetric since

$$
\exp \left( -\frac{(y - x)^2}{2\sigma^2} \right) = \exp \left( -\frac{(x - y)^2}{2\sigma^2} \right)
$$
In each iteration we do the following

- Draws a sample $x'$ from $Q(x'|x)$, where $x$ is the previous sample
- Calculated the acceptance probability

$$a(x'|x) = \min\left(1, \frac{P(x')}{P(x)}\right)$$

Note that we only need to compute $\frac{P(x')}{P(x)}$, the unknown constant cancels out

- Accept the new sample with probability $a(x'|x)$ or remain at state $x$. The acceptance probability ensures that, after sufficient many draws, our samples will come from the true distribution $\pi(x)$
Example: Gaussian Mixture Model

Adapted from Andrieu, Freitas, Doucet, Jordan, 2003
How do we know that the chain is going to converge to $\pi$?

Suppose the support of the proposal distribution is $\mathcal{X}$ (e.g., Gaussian distribution), then the Markov chain is irreducible and aperiodic.

We only need to verify the detailed balance condition

$$\pi(dx)p(x, dx') = \pi(x)dx \cdot Q(x'|x)a(x'|x)dx'$$

$$= \pi(x)Q(x'|x) \min \left( 1, \frac{\pi(x')}{\pi(x)} \right) dx dx'$$

$$= Q(x'|x) \min(\pi(x), \pi(x'))dx dx'$$

$$= Q(x|x') \min(\pi(x'), \pi(x))dx dx'$$

$$= \pi(x')dx' \cdot Q(x|x') \min \left( 1, \frac{\pi(x)}{\pi(x')} \right) dx$$

$$= \pi(dx')p(x', dx)$$
It turned out that symmetric proposal distribution is not necessary. Hastings (1970) later on generalized the above algorithm using the following acceptance probability for general $Q(x'|x)$

$$a(x'|x) = \min \left( 1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)} \right)$$

Similarly, we can show that detailed balanced condition is preserved.
Under mild assumptions on the proposal distribution $Q$, the algorithm is ergodic.

However, the choice of $Q$ is important since it determines the speed of convergence to $\pi$ and the efficiency of sampling.

Usually, the proposal distribution depend on the current state. But it can be independent of current state, which leads to an independent MCMC sampler that is somewhat like a rejection/importance sampling method.

Some examples of commonly used proposal distributions:

- $Q(x' | x) \sim \mathcal{N}(x, \sigma^2)$
- $Q(x' | x) \sim \text{Uniform}(x - \delta, x + \delta)$

Finding a good proposal distribution is hard in general.
Recall the univariate Gaussian model with known variance:

\[ y_i \sim \mathcal{N}(\theta, \sigma^2) \]

\[ p(y|\theta, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} \exp\left( -\frac{(y_i - \theta)^2}{2\sigma^2} \right) \]

Note that there is a conjugate \( \mathcal{N}(\mu_0, \tau_0^2) \) prior for \( \theta \), and the posterior has a close form normal distribution.

Now let’s pretend that we don’t know this exact posterior distribution and use a Markov chain to sample from it.
We can of course write the posterior distribution up to a constant

\[ p(\theta | y) \propto \exp \left( \frac{(\theta - \mu_0)^2}{2\tau_0^2} \right) \prod_{i=1}^{n} \exp \left( -\frac{(y_i - \theta)^2}{2\sigma^2} \right) = P(\theta) \]

We use \( N(\theta^{(i)}, 1) \), a normal distribution around our current state, to propose the next step.

Starting from an initial point \( \theta^{(0)} \) and propose the next step \( \theta' \sim N(\theta^{(0)}, 1) \), we either accept this value with probability \( a(\theta' | \theta^{(0)}) \) or reject and stay where we are.

We continue these steps for many iterations.
As we can see, the posterior distribution we obtained using the Metropolis algorithm is very similar to the exact posterior.
Now suppose we want to model the number of half court shots Stephen Curry has made in a game using Poisson model

\[ y_i \sim \text{Poisson}(\theta) \]

He made 0 and 1 half court shots in the first two games respectively.

We used Gamma(1.4, 10) prior for \( \theta \), and because of conjugacy, the posterior distribution also had a Gamma distribution

\[ \theta | y \sim \text{Gamma}(2.4, 12) \]

Again, let’s ignore the closed form posterior and use MCMC for sampling the posterior distribution.
The prior is
\[ p(\theta) \propto \theta^{0.4} \exp(-10\theta) \]

The likelihood is
\[ p(y|\theta) \propto \theta^{y_1+y_2} \exp(-2\theta) \]

where \( y_1 = 0 \) and \( y_2 = 1 \)

Therefore, the posterior is proportional to
\[ p(\theta|y) \propto \theta^{0.4} \exp(-10\theta) \cdot \theta^{y_1+y_2} \exp(-2\theta) = P(\theta) \]
Symmetric proposal distributions such as

$$\text{Uniform}(\theta^{(i)} - \delta, \theta^{(i)} + \delta) \text{ or } \mathcal{N}(\theta^{(i)}, \sigma^2)$$

might not be efficient since they do not take the non-negative support of the posterior into account.

Here, we use a non-symmetric proposal distribution such as $\text{Uniform}(0, \theta^{(i)} + \delta)$ and use the Metropolis-Hastings (MH) algorithm instead.

We set $\delta = 1$
We start from $\theta_0 = 1$ and follow these steps in each iteration

- Sample $\theta'$ from $\mathcal{U}(0, \theta^{(i)} + 1)$
- Calculate the acceptance probability

$$a(\theta'|\theta^{(i)}) = \min \left(1, \frac{P(\theta')\text{Uniform}(\theta^{(i)}|0, \theta' + 1)}{P(\theta^{(i)})\text{Uniform}(\theta'|0, \theta^{(i)} + 1)} \right)$$

- Sample $u \sim \mathcal{U}(0, 1)$ and set

$$\theta^{(i+1)} = \begin{cases} 
\theta' & u < a(\theta'|\theta^{(i)}) \\
\theta^{(i)} & \text{otherwise}
\end{cases}$$
Examples: Poisson Model with Gamma Prior
What if the distribution is multidimensional, \( i.e. \),
\[ x = (x_1, x_2, \ldots, x_d) \]

We can still use the Metropolis algorithm (or MH), with a multivariate proposal distribution, \( i.e. \), we now propose
\[ x' = (x'_1, x'_2, \ldots, x'_d) \]

For example, we can use a multivariate normal \( \mathcal{N}_d(x, \sigma^2 I) \), or a \( d \)-dimensional uniform distribution around the current state.
Here we construct a banana-shaped posterior distribution as follows

\[ y|\theta \sim \mathcal{N}(\theta_1 + \theta_2^2, \sigma_y^2), \quad \sigma_y = 2 \]

We generate data \( y_i \sim \mathcal{N}(1, \sigma_y^2) \)

We use a bivariate normal prior for \( \theta \)

\[ \theta = (\theta_1, \theta_2) \sim \mathcal{N}(0, I) \]

The posterior is

\[ p(\theta|y) \propto \exp \left( -\frac{\theta_1^2 + \theta_2^2}{2} \right) \cdot \exp \left( -\sum_i \frac{(y_i - \theta_1 - \theta_2^2)^2}{2\sigma_y^2} \right) \]

We use the Metropolis algorithm to sample from posterior, with a bivariate normal proposal distribution such as \( \mathcal{N}(\theta^{(i)}, (0.15)^2 I) \)
The first few samples from the posterior distribution of $\theta = (\theta_1, \theta_2)$, using a bivariate normal proposal.
Examples: Banana Shape Distribution

Posterior samples for $\theta = (\theta_1, \theta_2)$
Examples: Banana Shape Distribution

Trace plot of posterior samples for $\theta = (\theta_1, \theta_2)$
Sometimes, it is easier to decompose the parameter space into several components, and use the Metropolis (or MH) algorithm for one component at a time.

At iteration $i$, given the current state $(x_1^{(i)}, \ldots, x_d^{(i)})$, we do the following for all components $k = 1, 2, \ldots, d$:

- Sample $x'_k$ from the univariate proposal distribution $Q(x'_k | \ldots, x_{k-1}^{(i+1)}, x_k^{(i)}, \ldots)$

- Accept this new value and set $x_k^{(i+1)} = x'_k$ with probability

$$a(x'_k | \ldots, x_{k-1}^{(i+1)}, x_k^{(i)}, \ldots) = \min \left( 1, \frac{P(\ldots, x_{k-1}^{(i+1)}, x'_k, \ldots)}{P(\ldots, x_{k-1}^{(i+1)}, x_k^{(i)}, \ldots)} \right)$$

or reject it and set $x_k^{(i+1)} = x_k^{(i)}$
Note that in general, we can decompose the space of random variable into blocks of components.

Also, we can update the components sequentially or randomly.

As long as each transition probability individually leaves the target distribution invariant, their sequence would leave the target distribution invariant.

In Bayesian models, this is especially useful if it is easier and computationally less intensive to evaluate the posterior distribution when one subset of parameters change at a time.
In the example of banana-shaped distribution, we can sample $\theta_1$ and $\theta_2$ one at a time.

The first few samples from the posterior distribution of $\theta = (\theta_1, \theta_2)$, using a univariate normal proposal sequentially.
The Gibbs Sampler

- As the dimensionality of the parameter space increases, it becomes difficult to find an appropriate proposal distributions (e.g., with appropriate step size) for the Metropolis (or MH) algorithm.

- If we are lucky (in some situations we are!), the conditional distribution of one component, $x_j$, given all other components, $x_{-j}$ is tractable and has a close form so that we can sample from it directly.

- If that’s the case, we can sample from each component one at a time using their corresponding conditional distributions $P(x_j|x_{-j})$. 
This is known as the Gibbs sampler (GS) or “heat bath” (Geman and Geman, 1984)

Note that in Bayesian analysis, we are mainly interested in sampling from $p(\theta|y)$

Therefore, we use the Gibbs sampler when $P(\theta_j|y, \theta_{-j})$ has a closed form, e.g., there is a conditional conjugacy

One example is the univariate normal model. As we will see later, given $\sigma$, the posterior $P(\mu|y, \sigma^2)$ has a closed form, and given $\mu$, the posterior distribution of $P(\sigma^2|\mu, y)$ also has a closed form
The Gibbs sampler works as follows

1. Initialize starting value for \( x_1, x_2, \ldots, x_d \)
2. At each iteration, pick an ordering of the \( d \) variables (can be sequential or random)
   1. Sample \( x \sim P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d) \), i.e., the conditional distribution of \( x_i \) given the current values of all other variables
   2. Update \( x_i \leftarrow x \)
3. When we update \( x_i \), we immediately use its new value for sampling other variables \( x_j \)
GS is a Special Case of MH

- Note that in GS, we are not proposing anymore, we are directly sampling, which can be viewed as a proposal that will always be accepted.

- This way, the Gibbs sampler can be viewed as a special case of MH, whose proposal is

\[ Q(x'_i, x_{-i} | x_i, x_{-i}) = P(x'_i | x_{-i}) \]

- Applying MH with this proposal, we obtain

\[
\begin{align*}
 a(x'_i, x_{-i} | x_i, x_{-i}) &= \min \left( 1, \frac{P(x'_i, x_{-i}) Q(x_i, x_{-i} | x'_i, x_{-i})}{P(x_i, x_{-i}) Q(x'_i, x_{-i} | x'_i, x_{-i})} \right) \\
 &= \min \left( 1, \frac{P(x'_i, x_{-i}) P(x_i | x_{-i})}{P(x_i, x_{-i}) P(x'_i | x_{-i})} \right) = \min \left( 1, \frac{P(x'_i, x_{-i}) P(x_i, x_{-i})}{P(x_i, x_{-i}) P(x'_i, x_{-i})} \right) \\
&= 1
\end{align*}
\]
We can now use the Gibbs sampler to simulate samples from the posterior distribution of the parameters of a univariate normal $y \sim \mathcal{N}(\mu, \sigma^2)$ model, with prior

$$\mu \sim \mathcal{N}(\mu_0, \tau_0^2), \quad \sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$$

Given $(\sigma^{(i)})^2$ at the $i^{\text{th}}$ iteration, we sample $\mu^{(i+1)}$ from

$$\mu^{(i+1)} \sim \mathcal{N} \left( \frac{\mu_0 \tau_0^2 + \frac{n \bar{y}}{(\sigma^{(i)})^2}}{\frac{1}{\tau_0^2} + \frac{n}{(\sigma^{(i)})^2}}, \frac{1}{\tau_0^2} + \frac{n}{(\sigma^{(i)})^2} \right)$$

Given $\mu^{(i+1)}$, we sample a new $\sigma^2$ from

$$(\sigma^{(i+1)})^2 \sim \text{Inv-}\chi^2(\nu_0 + n, \frac{\nu_0 \sigma_0^2 + \nu n}{\nu_0 + n}), \quad \nu = \frac{1}{n} \sum_{j=1}^{n} (y_j - \mu^{(i+1)})^2$$
The following graphs show the trace plots of the posterior samples (for both $\mu$ and $\sigma$)

![Trace plots for $\mu$ and $\sigma$](image-url)
Gibbs sampling algorithms have been widely used in probabilistic graphical models

- Conditional distributions are fairly easy to derive for many graphical models (e.g., mixture models, Latent Dirichlet allocation)

- Have reasonable computation and memory requirements, only needs to sample one random variable at a time

- Can be Rao-Blackwellized (integrate out some random variable) to decrease the sampling variance. This is known as **collapsed Gibbs sampling**.
Energy-based models (EBMs) associate a scalar energy to each configuration of the variables of interest.

- We can modify the energy function so that its shape has desirable properties, e.g., plausible configurations would have lower energy.

- Energy-based probabilistic models define a probability distribution through an energy function as follows:

\[
p(x) = \frac{1}{Z} \exp(-E(x)), \quad Z = \sum_x \exp(-E(x))
\]

- EBMs can be learnt by maximizing the log-likelihood using stochastic gradient:

\[
-\frac{\partial \log p_\theta(x)}{\partial \theta} = \frac{\partial E_\theta(x)}{\partial \theta} - \mathbb{E}_{x \sim p_\theta(x)} \frac{\partial E_\theta(x)}{\partial \theta}
\]
In many cases, we do not have full observation, or we want to introduce latent variables to increase model capacity

\[ p(x) = \sum_h p(x, h) = \frac{1}{Z} \sum_h \exp(-E(x, h)) \] (1)

We can define free energy to turn (1) into a regular EBM

\[ p(x) = \frac{1}{Z} \exp(-\mathcal{F}(x)), \quad \mathcal{F}(x) = -\log \sum_h \exp(-E(x, h)) \]

An interesting form for the gradient

\[
- \frac{\partial \log p_\theta(x)}{\partial \theta} = \frac{\partial \mathcal{F}_\theta(x)}{\partial \theta} - \mathbb{E}_{x \sim p_\theta(x)} \frac{\partial \mathcal{F}_\theta(x)}{\partial \theta} \\
= \mathbb{E}_{h \sim p_\theta(h|x)} \frac{\partial E_\theta(x, h)}{\partial \theta} - \mathbb{E}_{x, h \sim p_\theta(x, h)} \frac{\partial E_\theta(x, h)}{\partial \theta}
\]
Restricted Boltzmann Machines (RBMs) are a particular form of EBMs where the energy function is a bilinear function of the visible and hidden variables

\[ E(v, h) = -b^T v - c^T h - h^T W v \]

The visible and hidden units are conditionally independent

\[ p(h|v) = \prod_i p(h_i|v), \quad p(v|h) = \prod_j p(v_j|h) \]
When \( v \) and \( h \) are binary variables, we have

\[
p(h_i = 1|v) = \text{sigmoid}(c_i + W_i v), \quad p(v_j = 1|h) = \text{sigmoid}(b_j + W_j^T h)
\]

Use Gibbs sampling for training and sampling

\[
h^{(n+1)} \sim \text{Bernoulli} \left( \text{sigmoid}(c + W v^{(n)}) \right)
\]

\[
v^{(n+1)} \sim \text{Bernoulli} \left( \text{sigmoid}(b + W^T h^{(n+1)}) \right)
\]

Contrastive Divergence:

\[
- \frac{\partial \log p_\theta(v)}{\partial \theta} \approx \frac{\partial \mathcal{F}_\theta(v)}{\partial \theta} - \frac{\partial \mathcal{F}_\theta(\tilde{v})}{\partial \theta}
\]

where \( \tilde{v} \) is a sample from the MCMC chain after \( k \) steps starting from the observed sample \( v \).
For more complex models, we might only have conditional conjugacy for one part of the parameters.

In such situations, we can combine the Gibbs sampler with the Metropolis method.

That is, we update the components with conditional conjugacy using Gibbs sampler and for the rest parameters, we use the Metropolis (or MH).
MCMC Diagnostics

- MCMC would converge to the target distribution if run sufficiently long
- However, it is often non-trivial to determine whether the chain has converged or not in practice
- Also, how do we measure the efficiency of MCMC chains?
- In what follows, we will discuss some practical advice for coding MCMC algorithms
Monitor convergence by plotting samples from multiple MH runs (chains)

- If the chains are well-mixed (left), they are probably converged
- If the chains are poorly-mixed (right), we may need to continue burn-in
An autocorrelation plot summarizes the correlation in the sequence of a Markov chain at different iteration lags.

A chain that has poor mixing will exhibit slow decay of the autocorrelation as the lag increases.
Since MCMC samples are correlated, *effective sample size* are often used to measure the efficiency when MCMC samples are used for estimation instead of independent samples.

The effective sample size (ESS) is defined as

$$\text{ESS} = \frac{n}{1 + 2 \sum_{k=1}^{\infty} \rho(k)}$$

where $\rho(k)$ is the autocorrelation at lag $k$.

ESS are commonly used to compare the efficiency of competing MCMC samplers for a given problem. Larger ESS usually means faster convergence.
One of the hardest problem to diagnose is whether or not the chain has become stuck in one or more modes of the target distribution.

In this case, all convergence diagnostics may indicate that the chain has converged, though it does not.

A partial solution: run multiple chains and compare the within- and between-chain behavior.

