1 Monte Carlo Method

The idea of Monte Carlo method is to use i.i.d. samples \( \{x^{(i)}\}_{i=1}^N \) drawn from a target density distribution \( p(x) \) to approximate the distribution and statistics related to it. The strong law of large numbers tells us that the statistics calculated from the empirical distribution will converge almost surely to the true value derived from the true distribution, and central limit theorem would tell us how the distribution of estimation error converges.

To be more specific, the target distribution can be approximated by

\[
p_N(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x)
\]

where \( \delta_{x^{(i)}}(x) \) denotes the delta-Dirac mass located at \( x^{(i)} \). Therefore, the integral \( I_f = \int_{\mathcal{X}} f(x)p(x) \, dx \) can be approximated by \( I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)}) \). The strong law of large numbers tells us that the estimator is unbiased.

\[
I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)}) \xrightarrow{a.s.} \int_{\mathcal{X}} f(x)p(x) \, dx.
\]

If \( \sigma_f^2 < \infty \), where \( \sigma_f^2 = \mathbb{E}_{p(x)} [f^2(x)] - I^2(f) \), the variance of this estimation is given by \( \text{var}(I_N(f)) = \frac{\sigma_f^2}{N} \). Then the central limit theorem controls the distribution of the estimation error by

\[
\sqrt{N}(I_N(f) - I_f) \xrightarrow{d, N \to \infty} \mathcal{N}(0, \sigma_f^2).
\]
An interesting example to show the power of Monte Carlo method is Buffon’s needle problem, where we can use the times a needle dropped on the floor lies across a line between two strips to approximate the constant $\pi$.

## 2 Markov Chain Monte Carlo

### 2.1 Introduction

Though Monte Carlo method can be very powerful, it’s always not easy to sample from the target distribution, especially in high-dimensional space. There are many sampling frameworks, among which Markov Chain Monte Carlo is a broad type of sampling strategies which explore the state space of $\mathcal{X}$ by Markov chain mechanism.

The idea of MCMC is to construct a Markov chain such that samples from this chain can mimic samples from our desired distribution $p(x)$. From the limit theorem of Markov chain, we know that for all irreducible recurrent aperiodic Markov chain, there’s a unique stationary distribution, and when we run the chain by transition matrix for long enough, the chain will converge to this stationary distribution. This limit theorem suggests us that if we can construct our chain such that it’s irreducible, aperiodic and has the our target distribution as its stationary distribution $p(x)$, we’re able to generate samples from $p$ by taking samples from this chain after it reaches equilibrium. Now the only problem remains is how to construct the transition matrix such that the chain has above properties.

### 2.2 Metropolis-Hastings algorithm

Metropolis-Hastings algorithm is the most popular algorithm for MCMC. At each iteration $i$, the Markov chain is at $x^{(i)}$. It makes use of a proposal distribution $q(x^*|x^{(i)})$ which is easy to sample from to propose a candidate $x^*$, to suggest where to move. And then it randomly accepts the candidate $x^*$ and makes an actual transition in the chain by some acceptance probability $A(x^*,x^{(i)})$. The pseudo code is shown in algorithm 1.

Under the condition that the support of proposal distribution $q(\cdot)$ includes the supper of target
Algorithm 1 Metropolis-Hastings algorithm

Initialize $x^{(0)}$ randomly

for $i = 0$ to $N - 1$ do

Sample $u \sim U[0, 1]$
Sample $x^* \sim q(x^* | x^{(i)})$

if $u < \mathcal{A}(x^*, x^{(i-1)}) = \min \{ 1, \frac{p(x^*) q(x^{(i)} | x^*)}{p(x^{(i)}) q(x^* | x^{(i)})} \}$ then

\hspace{1cm} $x^{(i+1)} = x^*$

else

\hspace{1cm} $x^{(i+1)} = x^{(i)}$

end if

end for

distribution $p(\cdot)$, the correctness of this algorithm can be proved as follow.

- support $q(\cdot) \supseteq \text{support } p(\cdot)$, so the chain is irreducible

- allows for rejection, so the chain is aperiodic

- transition matrix $T(x^* | x^{(i)}) = q(x^* | x^{(i)}) \min \{ 1, \frac{p(x^*) q(x^{(i)} | x^*)}{p(x^{(i)}) q(x^* | x^{(i)})} \}$ satisfies reversibility

\[
p(x^{(i)}) q(x^* | x^{(i)}) \min \{ 1, \frac{p(x^*) q(x^{(i)} | x^*)}{p(x^{(i)}) q(x^* | x^{(i)})} \} = \min \{ p(x^{(i)}) q(x^* | x^{(i)}), p(x^*) q(x^{(i)} | x^*) \} = p(x^*) q(x^{(i)} | x^*) \min \{ 1, \frac{p(x^*) q(x^{(i)} | x^*)}{p(x^{(i)}) q(x^* | x^{(i)})} \}
\]

Therefore $p(x)$ is the stationary distribution of the constructed Markov chain with transition matrix $T$. If we run the chain for long enough until equilibrium, we’ll get true samples of $p(x)$ from this Markov chain.

2.3 Gibbs Sampling

Gibbs Sampling is an important special case of Metropolis-Hastings algorithm with conditional distribution as proposal distribution $q(\cdot)$ and the acceptance rate 1. Algorithm details can be found in algorithm 2.

In algorithm 2, the inner loop can be viewed as the composition of $K$ (the number of dimensions) Markov chain, where the transition matrix is the product of all $K$ transition matrices. Therefore
Algorithm 2 Gibbs Sampling

Initialize $x^{(0)}_{1:K}$ randomly

for $i = 0$ to $N-1$ do

  for $j = 1$ to $K$ do
    Sample $x^{(i+1)}_j \sim p(x_j | x^{(i+1)}_1, x^{(i+1)}_2, \ldots, x^{(i+1)}_{j-1}, x^{(i+1)}_{j+1}, \ldots x^{(i+1)}_K)$
    \> conditional distribution as proposal distribution
    \> acceptance rate is always 1
  end for
end for

to prove its correctness, it suffices to show within each inner loop, the transition matrix satisfies the reversibility property. Here $q(x)$ is defined as

$$q(x^* | x^{(i)}) = \begin{cases} 
  p(x^*_j | x^{(i)}_{-j}) & \text{If } x^*_j = x^{(i)}_j, \\
  0 & \text{Otherwise}
\end{cases}$$

It will lead to

$$p(x^{(i)})q(x^* | x^{(i)}) = p(x^{(i)}_j) p(x^*_j | x^{(i)}_{-j}) q(x^* | x^{(i)})$$

$$= p(x^{(i)}_j) p(x^*_j | x^{(i)}_{-j}) p(x^*_j | x^{(i)}_{j})$$

$$= p(x^*_j) p(x^*_j | x^{(i)}_{-j}) p(x^{(i)}_j | x^*_j)$$

$$= p(x^*) q(x^{(i)} | x^*)$$

which means $A(x^*, x^{(i)}) = 1$ is true all the time.
The correctness of Gibbs sampling can also be shown directly

\[ \sum_{x \in \mathcal{X}} p(x) T(x' | x) \]

\[ = \sum_{x_1, x_2, \ldots, x_K} p(x_1, x_2, \ldots, x_K)p(x'_1|x_2, \ldots, x_K)p(x'_2|x_3, \ldots, x_K) \ldots p(x'_{K-1}|x', \ldots, x_{K-1}) \]

\[ = \sum_{x_2, \ldots, x_K} p(x_2, \ldots, x_K)p(x'_1|x_2, \ldots, x_K)p(x'_2|x_3, \ldots, x_K) \ldots p(x'_{K-1}|x', \ldots, x_{K-1}) \]

\[ = \sum_{x_2, \ldots, x_K} p(x'_1, x_2, \ldots, x_K)p(x'_2|x'_1, x_3, \ldots, x_K) \ldots p(x'_{K-1}|x', \ldots, x_{K-1}) \]

\[ = \sum_{x_3, \ldots, x_K} p(x'_1, x_2, x_3, \ldots, x_K) \ldots p(x'_{K-1}|x', \ldots, x_{K-1}) \]

\[ = \ldots \]

\[ = p(x'_1, x'_2, \ldots, x'_K) = p(x') \]

So \( p(x) \) is the stationary distribution under transition \( T \).

### 2.4 Discussion

There are some practical issues regarding MH algorithm that is worth notice. First of all we only need to know the target distribution \( p(x) \) up to a normalizing constant. Because the only thing matters in the algorithm is \( \frac{p(x_1)}{p(x_2)} \). Secondly, there are two critical hyper-parameters to tune for MH to work well in practice, which all relate to the mixing time of Markov chain. How long we wait till we assume the chain reaches equilibrium, called burn-in time, would effect whether our samples indeed come from the desired distribution. And how many steps we take between two samples, would influence the sample correlation. To deal with the correlation issue, we might want to run many Markov chains in parallel and gather the results from different chains.

Finally, I think there’s a trade off between acceptance rate and sample correlation, which highly depends on the proposal distribution we choose. For example, in the case of Gibbs sampling, we will always accept samples due to the property of conditional distribution. However, conditional distribution also results in high correlation between consecutive samples, which is problematic because the chain can stay in the same area of probability space for a long time.

The other extreme to the Gibbs sampling is called independent sampler, who takes \( q(x^*|x^{(i)}) =\)
$q(x^*)$, independent of $x^{(i)}$ as its proposal distribution. Then the acceptance rate is 
$
\mathcal{A}(x^*, x^{(i)}) = \min \left\{ 1, \frac{p(x^*) q(x^{(i)})}{p(x^{(i)}) q(x^*)} \right\}
$
which could be low if $q(x)$ is quite different from $p(x)$. The tradeoff can also be understood as if we take more information from $p(x)$ to construct $q(\cdot)$, we might suffer from sample correlation, while if we know too less, we’re in danger of low acceptance rate.

Other interesting variants of MH algorithm include using symmetric random walk proposal distribution where $q(x^*|x^{(i)}) = q(x^{(i)}|x^*)$ always holds.

Hamiltonian Monte Carlo, on the other hand, introduces Hamiltonian dynamics to design proposal distribution and then applies a Metropolis update. One good property of Hamiltonian dynamics is that the reversibility can be satisfied automatically. In HMC algorithm, we first define a Hamiltonian function in terms of the target distribution. In addition to the variables we are interested in (the position variables in Hamiltonian dynamics), we must as well introduce another auxiliary momentum variables, which commonly have independent Gaussian distributions due to its simplicity in sampling. The HMC method alternatively updates the momentum and the position variable with Metropolis updates in which, new state is proposed by computing a trajectory according to Hamiltonian dynamics, implemented using leapfrog method. The reason why people turn to Hamiltonian dynamics for proposal distribution is that it could reach high acceptance rate and low sample correlation at the same time compared to other methods, though at the expense of computational complexity. One reason I think HMC is able to reach that is because it incorporates information about the gradient of $p(x)$. However, even for HMC, there’s still the tradeoff between sample correlation and acceptance rate. And we need to tune the hyper-parameters to find a balance point for this tradeoff.

3 MCMC Simulation on Ising Model

The distribution of two dimensional Ising model is given by

$$p_\beta(S) = \frac{\exp(-\beta H(S))}{Z_\beta}$$
where \( H(S) = - \sum_{(i,j)} J_{ij} S_i S_j \), a sum over all neighboring interactions. \( \forall i \ S_i \in \{-1,1\} \) and \( J_{ij} \) indicates the interaction strength between \( S_i \) and \( S_j \). \( \beta = \frac{1}{k_B T} \). \( Z \) is the partition function which is given by

\[
Z_{\beta} = \sum_{S \in \mathcal{S}} \exp(-\beta H(S))
\]

which is a sum over exponentially large number of possible configurations, and usually hard to compute.

This model is a classic mathematical description for ferromagnetism in statistical mechanics. The discrete variables \( S_i \) represents magnetic dipole moments of atomic spins that can be in one of two states (+1 or -1). We say the interaction is ferromagnetic if \( J_{ij} > 0 \), antiferromagnetic if \( J_{ij} < 0 \), and noninteracting when \( J_{ij} = 0 \). The model allows the identification of phase transitions.

I implement a Gibbs sampling algorithm in Matlab for Ising model, where the conditional distribution is completely decided by its at most 4 neighbors. The pseudo code can be found in algorithm 3.

**Algorithm 3 Ising Model**

```plaintext
Initialize Spin\(^{(0)}\)_{L \times L} \text{ randomly}
for \( i = 0 \) to \( N-1 \) do
    randomly sample \( 0.2L^2 \) non-neighboring position
    sample the new spin by conditional distribution on these positions
    \( x \sim \text{Bernoulli}\left(\frac{\exp(-J_{\text{pos} E})}{\exp(-J_{\text{pos} E}) + \exp(J_{\text{pos} E})}\right)\)
    Update the Spin with \( x \)
end for
```

Each time I sample 20\% of non-conflicting positions to try to change its spins. I use the burn in time as \( 10^7 \) iterations and sample interval as 100 iterations. I changed the interaction strength from 0 to 1 of 20 different values, and for each interaction strength, I took 10000 samples. But unfortunately I was not able to see any trends in my simulation. Fig. 1 is a sample image of interaction strength 0.01 on \( 100 \times 100 \) grids.
Figure 1: Sample from Ising Model