8. Monte Carlo Simulations

Observables in classical statistical mechanics

- the average value of an observable A is

$$\langle A \rangle = \int p(\mathbf{r}_1, \mathbf{r}_2, \ldots) A(\mathbf{r}_1, \mathbf{r}_2, \ldots)\, d\mathbf{r}_1\, d\mathbf{r}_2\, \ldots$$
$$= \frac{\int \exp(-\beta U(\mathbf{r}_1, \mathbf{r}_2, \ldots)) A(\mathbf{r}_1, \mathbf{r}_2, \ldots)\, d\mathbf{r}_1\, d\mathbf{r}_2\, \ldots}{\int \exp(-\beta U(\mathbf{r}_1, \mathbf{r}_2, \ldots))\, d\mathbf{r}_1\, d\mathbf{r}_2\, \ldots}$$

- a force field specifies $U(\mathbf{r}_1, \mathbf{r}_2, \ldots)$ for a given system
- to calculate the average value of an observable, we need to draw samples from the Boltzmann distribution $p(\mathbf{r}_1, \mathbf{r}_2, \ldots)$

Sampling from a probability distribution

- for a probability distribution $p(x)$, drawing samples from it means generating a sequence of random numbers $\{x_1, x_2, \ldots\}$ such that the probability of x being in the sequence is $p(x)$
- in other words, the histogram of sampled numbers should match the probability distribution
- when $p(x)$ is a standard distribution, such as the uniform or normal distribution, specialized procedures are available to draw samples from it
- however, these specialized procedures do not work for general probability distributions

Sampling from standard distributions

- assume we know how to draw samples from a uniform distribution on the interval $[0, 1]$

$$x = \text{rand}()$$

- how to draw samples from a uniform distribution on the interval $[a, b]$

Sampling from standard distributions

- assume we know how to draw samples from a uniform distribution on the interval $[0, 1]$

$$x = \text{rand}()$$

- how to draw samples from a uniform distribution on the interval $[a, b]$

$$y = a + (b - a) \times x$$

- how to draw samples from a normal distribution with mean 0 and standard deviation 1 the Box-Muller transform

$$u = \text{rand}(); v = \text{rand}()$$
$$x = \sqrt{-2 \ln u} \cos(2\pi v)$$
$$y = \sqrt{-2 \ln u} \sin(2\pi v)$$

- many programming languages provide functions to draw samples from standard distributions

General methods for sampling

- Monte Carlo methods
  - rejection sampling
  - importance sampling
  - Markov chain Monte Carlo - Metropolis-Hastings algorithm
- Molecular dynamics simulations

Rejection sampling

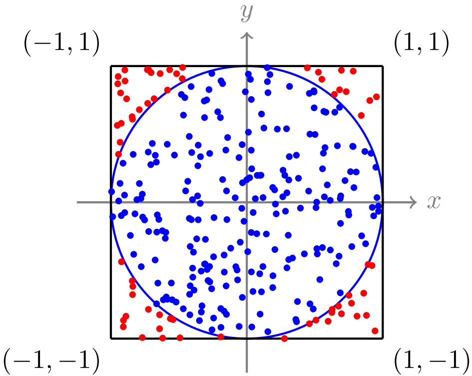
- how to draw samples from a uniform distribution inside a circle of radius 1 centered at the origin in 2D

Rejection sampling

- how to draw samples from a uniform distribution inside a circle of radius 1 centered at the origin in 2D

repeat the following steps

1. draw a random point (x, y) in the square $[-1, 1] \times [-1, 1]$
2. if the point is inside the circle, keep it; otherwise, discard it



Importance sampling

- the average value of an observable $A(x)$ with respect to a probability distribution $p(x)$ is

$$\langle A \rangle = \int p(x) A(x) \, dx$$

- if drawing samples from $p(x)$ is difficult whereas drawing samples from another probability distribution $q(x)$ is easy, the average value can be estimated as

$$\langle A \rangle = \int q(x) \frac{p(x)}{q(x)} A(x) \, dx \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i)}{q(x_i)} A(x_i),$$

where $\{x_1, x_2, \ldots, x_N\}$ are samples drawn from $q(x)$

- $w(x_i) = p(x_i)/q(x_i)$ is called the importance weight

Importance sampling

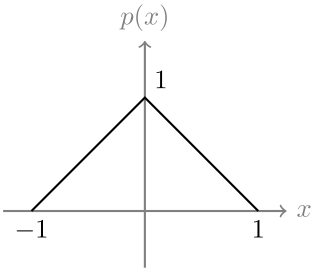
- to generate samples from a probability distribution $p(x)$ using importance sampling
  1. draw samples $\{x_1, x_2, \ldots, x_N\}$ from a probability distribution $q(x)$
  2. calculate the importance weights $\{w(x_1), w(x_2), \ldots, w(x_N)\}$
- the average value of an observable $A(x)$ with respect to $p(x)$ is

$$\langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N} w(x_i) A(x_i)$$

- for $q(x)$, $\{x_1, x_2, \ldots, x_N\}$ are samples with equal weights; for $p(x)$, $\{x_1, x_2, \ldots, x_N\}$ are samples with importance weights
- $\{x_1, x_2, \ldots, x_N\}$ and their importance weights can be used to generate samples of $p(x)$ with equal weights by resampling

Sampling from a triangular distribution

- how to draw samples from the following triangular distribution



  - using rejection sampling
  - using importance sampling

Rejection sampling and importance sampling

- both requires sampling from a proposal distribution $q(x)$
- their efficiency depends on the choice of $q(x)$
- $q(x)$ should be close to the target distribution $p(x)$
- $q(x)$ should be easy to sample from
- difficult to find a good proposal distribution for a general target distribution $p(x)$

Markov chain Monte Carlo

- a general method to sample from a probability distribution $p(x)$
- generates a sequence of samples $\{x_1, x_2, \ldots\}$, where each sample is generated using the previous sample, i.e., it is a Markov chain
- assume that the sample at step i is x_o, the sample at step $i+1$ is generated based on the transition probability $T(x_n|x_o)$
- the transition probability is designed such that the samples generated by the Markov chain converge to the target distribution $p(x)$
- one design principle is to make the transition probability satisfy the *detailed balance* condition

$$p(x_o)T(x_n|x_o) = p(x_n)T(x_o|x_n)$$

Detailed balance condition

- the detailed balance condition

$$p(x_o)T(x_n|x_o) = p(x_n)T(x_o|x_n)$$

implies that the probability flow going from x_o to x_n is equal to the probability mass going from x_n to x_o

- along with other conditions, the detailed balance condition ensures that $p(x)$ is invariant with respect to the transition probability

$$x_o \sim p(x) \implies x_n \sim p(x)$$

proof

$$\sum_{x_o} p(x_o)T(x_n|x_o) = \sum_{x_o} p(x_n)T(x_o|x_n) = p(x_n)\sum_{x_o} T(x_o|x_n) = p(x_n)$$

Metropolis-Hastings algorithm

- a special case of the Markov chain Monte Carlo method
- the algorithm
  1. given the sample x_o at step i, propose a new sample x_n using a proposal distribution $q(x_n|x_o)$
  2. calculate the acceptance probability
  $$\alpha(x_n|x_o) = \min\left(1, \frac{p(x_n)q(x_o|x_n)}{p(x_o)q(x_n|x_o)}\right)$$
  3. $x_{i+1} = x_n$ with probability $\alpha(x_n|x_o)$; otherwise, $x_{i+1} = x_o$
- the transition probability is
$$T(x_n|x_o) = q(x_n|x_o)\alpha(x_n|x_o)$$

Metropolis-Hastings algorithm

- it satisfies the *detailed balance* condition

$$p(x_o)T(x_n|x_o) = p(x_o)q(x_n|x_o)\alpha(x_n|x_o) = p(x_n)q(x_o|x_n)\alpha(x_o|x_n) = p(x_n)T(x_o|x_n)$$

- when $q(x_n|x_o) = q(x_o|x_n)$, the acceptance probability simplifies to

$$\alpha(x_n|x_o) = \min\left(1, \frac{p(x_n)}{p(x_o)}\right)$$

  and the detailed balance condition simplifies to

$$p(x_o)\alpha(x_n|x_o) = p(x_n)\alpha(x_o|x_n)$$

Metropolis-Hastings algorithm for Boltzmann distributions

- the Boltzmann distribution is

$$p(\mathbf{r}_1, \mathbf{r}_2, \ldots) = \frac{1}{Z} \exp(-\beta U(\mathbf{r}_1, \mathbf{r}_2, \ldots))$$

- the Metropolis-Hastings algorithm for the Boltzmann distribution

  1. given the sample $\mathbf{r}_o$ at step i, propose a new sample $\mathbf{r}_n$ using a proposal distribution $q(\mathbf{r}_n|\mathbf{r}_o)$
  2. calculate the acceptance probability

  $$\alpha(\mathbf{r}_n|\mathbf{r}_o) = \min\left(1, \frac{\exp(-\beta U(\mathbf{r}_n))}{\exp(-\beta U(\mathbf{r}_o))}\right) = \min\left(1, \exp(-\beta \Delta U)\right)$$

  where $\Delta U = U(\mathbf{r}_n) - U(\mathbf{r}_o)$
  3. $\mathbf{r}_{i+1} = \mathbf{r}_n$ with probability $\alpha(\mathbf{r}_n|\mathbf{r}_o)$; otherwise, $\mathbf{r}_{i+1} = \mathbf{r}_o$