STAT 491/516: Stochastic Modeling of Scientific Data **Autumn 2026**

Lecture 7: Monte Carlo Methods

Instructor: Yen-Chi Chen

These notes are partially based on those of Mathias Drton.

7.1 Introduction

Monte Carlo methods refer to numerical methodologies based on computer simulation. In many cases, an estimator or a statistic may not have a closed-form so it is hard to numerically compute its value from a given dataset. In this case, we need to use some numerical method to evaluate its value. For instance, in Bayesian analysis, we know that posterior distribution is very important. However, if we are not using a conjugate prior, the posterior often does not have a closed-form so we cannot write down the posterior mean or MAP. On the other hand, if we can generate from a posterior distribution, we can use the 'sample average' of these generated points as a proxy to the posterior mean and the 'sample mode' of these points as a proxy to the MAP. This idea–generating points from the desired distribution and use them as a proxy for the desired quantity–is what Monte Carlo is about.

7.2 Concepts of Monte Carlo

Note: we will not cover much of this section in the lecture – please read it on your own.

7.2.1 Monte Carlo Integration

Assume we want to evaluate the following integration:

$$\int_0^1 e^{-x^3} dx.$$

What can we do? The function e^{-x^3} does not seem to have a closed form solution so we have to use some computer experiment to evaluate this number. The traditional approach to evaluate this integration is to use so-called the *Riemann Integration*, where we choose points $x_1, \cdots, x_K$ evenly spread out over the interval $[0, 1]$ and then we evaluate $f(x_1), \cdots, f(x_K)$ and finally use

$$\frac{1}{K}\sum_{i=1}^{K} f(x_i)$$

to evaluate the integration. When the function is smooth and $K \to \infty$, this numerical integration converges to the actual integration.

Now we will introduce an alternative approach to evaluate such an integration. First, we rewrite the integration as

$$\int_0^1 e^{-x^3} \cdot 1 dx = \mathbb{E}\left(e^{-U^3}\right),$$

where U is a uniform random variable over the interval $[0,1]$. Thus, the integration is actually an expected value of a random variable e^{-U^3}, which implies that evaluating the integration is the same as *estimating the expected value*. So we can generate IID random variables $U_1, \cdots, U_K \sim \mathsf{Uni}[0,1]$ and then compute $W_1 = e^{-U_1^3}, \cdots, W_K = e^{-U_K^3}$ and finally use

$$\bar{W}_K = \frac{1}{K}\sum_{i=1}^{K} W_i = \frac{1}{K}\sum_{i=1}^{K} e^{-U_i^3}$$

as a numerical evaluation of $\int_0^1 e^{-x^3} dx$. By the Law of Large Number,

$$\bar{W}_K \xrightarrow{P} \mathbb{E}(W_i) = \mathbb{E}\left(e^{-U_i^3}\right) = \int_0^1 e^{-x^3} dx,$$

so this alternative numerical method is statistically consistent.

In the above example, the integration can be written as

$$I = \int f(x)p(x)dx, \tag{7.1}$$

where f is some function and p is a probability density function. Let X be a random variable with density p. Then equation (7.1) equals

$$\int f(x)p(x)dx = \mathbb{E}(f(X)) = I.$$

Namely, the result of this integration is the same as the expected value of the random variable $f(X)$. The alternative numerical method to evaluate the above integration is to generate IID $X_1, \cdots, X_N \sim p$, N data points, and then use the sample average

$$\widehat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(X_i).$$

This method, the method of evaluating the integration via simulating random points, is called the integration by *Monte Carlo Simulation*.

An appealing feature of the Monte Carlo Simulation is that the statistical theory is rooted in the theory of sample average. We are using the sample average as an estimator of the expected value. We have already seen that the bias and variance of an estimator are key quantities of evaluating the quality of an estimator. What will be the bias and variance of our Monte Carlo Simulation estimator?

The bias is simple–we are using the sample average as an estimator of it expected value, so the $\mathbf{bias}(\widehat{I}_N) = 0$. The variance will then be

$$\begin{aligned}
\mathsf{Var}(\widehat{I}_N) &= \frac{1}{N}\mathsf{Var}(f(X_1)) \\
&= \frac{1}{N}\left(\mathbb{E}(f^2(X_1)) - \underbrace{\mathbb{E}^2(f(X_1))}_{I^2}\right) \\
&= \frac{1}{N}\left(\int f^2(x)p(x)dx - I^2\right).
\end{aligned} \tag{7.2}$$

Thus, the variance contains two components: $\int f^2(x)p(x)dx$ and I^2.

Given a problem of evaluating an integration, the quantity I is fixed. What we can choose is the number of random points N and the *sampling distribution* p! An important fact is that when we change the sampling distribution p, the function f will also change.

For instance, in the example of evaluating $\int_0^1 e^{-x^3}dx$, we have seen an example of using uniform random variables to evaluate it. We can also generate IID $B_1, \cdots, B_K \sim \mathsf{Beta}(2,2)$, K points from the beta distribution $\mathsf{Beta}(2,2)$. Note that the PDF of $\mathsf{Beta}(2,2)$ is

$$p_{\mathsf{Beta}(2,2)}(x) = 6x(1-x).$$

We can then rewrite

$$\int_0^1 e^{-x^3}dx = \int_0^1 \underbrace{\frac{e^{-x^3}}{6x(1-x)}}_{f(x)} \cdot \underbrace{6x(1-x)}_{p(x)} dx = \mathbb{E}\left(\frac{e^{-B_1^3}}{6B_1(1-B_1)}\right).$$

What is the effect of using different sampling distribution p? The expectation is always fixed to be I so the second part of the variance remains the same. However, the first part of the variance $\int f^2(x)p(x)dx$ depends how you choose p and the corresponding f.

Thus, different choices of p leads to a different variance of the estimator. We will talk about how to choose an optimal p in Chapter 4 when we talk about theory of importance sampling.

7.2.2 Estimating a Probability via Simulation

Here is an example of evaluating the power of a Z-test. Let $X_1, \cdots, X_{16}$ be a size 16 random sample. Let the null hypothesis and the alternative hypothesis be

$$H_0 : X_i \sim N(0,1), \quad H_a : X_i \sim N(\mu, 1),$$

where $\mu \neq 0$. Under the significance level α, the two-tailed Z-test is to reject H_0 if $\sqrt{16}|\bar{X}_{16}| \geq z_{1-\alpha/2}$, where $z_t = F^{-1}(t)$, where F is the CDF of the standard normal distribution. Assume that the true value of μ is $\mu = 1$. In this case, the null hypothesis is wrong and we should reject the null. However, due to the randomness of sampling, we may not be able to reject the null every time. So a quantity we will be interested in is: *what is the probability of rejecting the null under such μ?* In statistics, this probability (the probability that we reject H_0) is called the *power* of a test. Ideally, if H_0 is incorrect, we want the power to be as large as possible.

What will the power be when $\mu = 1$? Here is the analytical derivation of the power (generally denoted as β):

$$\begin{aligned}
\beta &= P(\text{Reject } H_0 | \mu = 1) \\
&= P(\sqrt{16}|\bar{X}_{16}| \geq z_{1-\alpha/2} | \mu = 1), \qquad \bar{X}_{16} \sim N(\mu, 1/16) \\
&= P(4 \cdot |N(1, 1/16)| \geq z_{1-\alpha/2}) \\
&= P(|N(4,1)| \geq z_{1-\alpha/2}) \\
&= P(N(4,1) \geq z_{1-\alpha/2}) + P(N(4,1) \leq -z_{1-\alpha/2}) \\
&= P(N(0,1) \geq z_{1-\alpha/2} - 4) + P(N(0,1) \leq -4 - z_{1-\alpha/2}).
\end{aligned} \tag{7.3}$$

Well...this number does not seem to be an easy one...

What should we do in practice to compute the power? Here is an alternative approach of computing the power using the Monte Carlo Simulation. The idea is that we generate N samples, each consists of 16 IID random variables from $N(1,1)$ (the distribution under the alternative). For each sample, we compute the Z-test statistic, $\sqrt{16}|\bar{X}_{16}|$, and check if we can reject H_0 or not (i.e., checking if this number is greater than or equal to $z_{1-\alpha/2}$). At the end, we use the ratio of total number of H_0 being rejected as an estimate of the

power β. Here is a diagram describing how the steps are carried out:

$$
\begin{aligned}
N(1,1) &\xrightarrow{\text{generates}} 16 \text{ observations} \xrightarrow{\text{compute}} \text{test statistic } \left(\sqrt{16}|\bar{X}_{16}|\right) \xrightarrow{\text{Reject } H_0} D_1 = \text{Yes}(1)/\text{No}(0) \\
N(1,1) &\xrightarrow{\text{generates}} 16 \text{ observations} \xrightarrow{\text{compute}} \text{test statistic } \left(\sqrt{16}|\bar{X}_{16}|\right) \xrightarrow{\text{Reject } H_0} D_2 = \text{Yes}(1)/\text{No}(0) \\
&\vdots \\
N(1,1) &\xrightarrow{\text{generates}} 16 \text{ observations} \xrightarrow{\text{compute}} \text{test statistic } \left(\sqrt{16}|\bar{X}_{16}|\right) \xrightarrow{\text{Reject } H_0} D_N = \text{Yes}(1)/\text{No}(0)
\end{aligned}
$$

Each sample will end up with a number D_i such that $D_i = 1$ if we reject H_0 and $D_i = 0$ if we do not reject H_0.

Because the Monte Carlo Simulation approach is to use the ratio of total number of H_0 being rejected to estimate β, this ratio is

$$\bar{D}_N = \frac{\sum_{j=1}^{N} D_j}{N}.$$

Is the Monte Carlo Simulation approach a good approach to estimate β? The answer is–yes it is a good approach of estimating β and moreover, we have already learned the statistical theory of such a procedure!

The estimator $\bar{D}_N$ is just a sample average and each D_j turns out to be a Bernoulli random variable with parameter

$$p = P(\text{Reject } H_0 | \mu = 1) = \beta$$

by equation (7.3). Therefore,

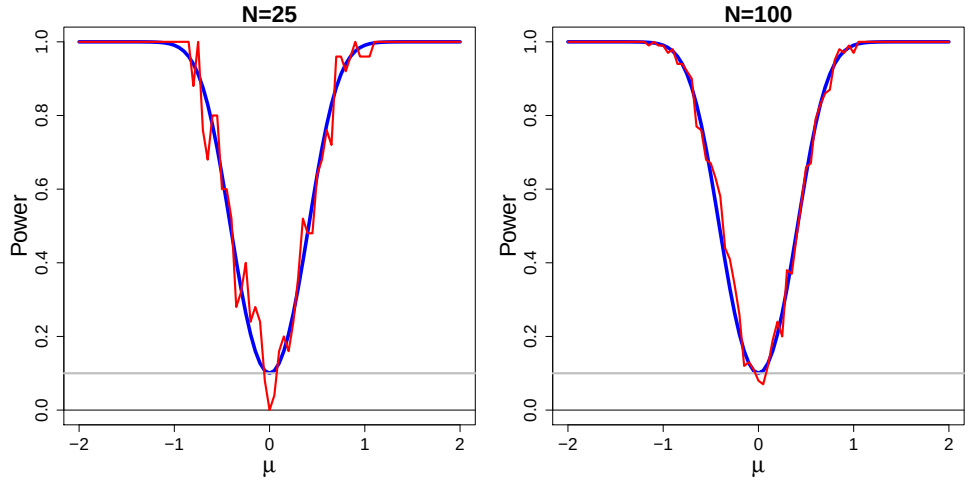
$$
\begin{aligned}
\textbf{bias}\left(\bar{D}_N\right) &= \mathbb{E}(\bar{D}_N) - \beta = p - \beta = 0 \\
\textbf{Var}\left(\bar{D}_N\right) &= \frac{p(1-p)}{N} = \frac{\beta(1-\beta)}{N} \\
\textbf{MSE}\left(\bar{D}_N, \beta\right) &= \frac{\beta(1-\beta)}{N}.
\end{aligned}
$$

Thus, the Monte Carlo Simulation method yields a consistent estimator of the power:

$$\bar{D}_N \xrightarrow{P} \beta.$$

Although here we study the Monte Carlo Simulation estimator of such a special case, this idea can be easily to generalize to many other situation as long as we want to evaluate certain numbers. In modern statistical analysis, most papers with simulation results will use some Monte Carlo Simulations to show the numerical results of the proposed methods in the paper.

The following two figures present the power β as a function of the value of μ (blue curve) with $\alpha = 0.10$. The red curves are the estimated power by Monte Carlo simulations using $N = 25$ and 100.



$\longrightarrow$ The gray line corresponds to the value of power being 0.10. Think about why the power curve (blue curve) hits the gray line at $\mu = 0$.

7.2.3 Estimating Distribution via Simulation

Monte Carlo Simulation can also be applied to estimate an unknown distribution as long as we can generate data from such a distribution. In Bayesian analysis, people are often interested in the so-called *posterior* distribution. Very often, we known how to generate points from a posterior distribution but we cannot write down its closed form. In this situation, what we can do is to simulate many points and estimate the distribution using these simulated points. So the task becomes:

$$\text{given } X_1, \cdots, X_n \sim F \text{ (or PDF } p\text{), we want to estimate } F \text{ (or the PDF } p\text{).}$$

Estimating the CDF using EDF. To estimate the CDF, a simple but powerful approach is to use the empirical distribution function:

$$\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x).$$

Estimating the PDF using histogram. If the goal is to estimate the PDF, then this problem is called *density estimation*, which is a central topic in statistical research. Here we will focus on the perhaps simplest approach: histogram.

For simplicity, we assume that $X_i \in [0,1]$ so $p(x)$ is non-zero only within $[0,1]$. We also assume that $p(x)$ is smooth and $|p'(x)| \le L$ for all x (i.e. the derivative is bounded). The histogram is to partition the set $[0,1]$ (this region, the region with non-zero density, is called the support of a density function) into several bins and using the count of the bin as a density estimate. When we have M bins, this yields a partition:

$$B_1 = \left[0, \frac{1}{M}\right), B_2 = \left[\frac{1}{M}, \frac{2}{M}\right), \cdots, B_{M-1} = \left[\frac{M-2}{M}, \frac{M-1}{M}\right), B_M = \left[\frac{M-1}{M}, 1\right].$$

In such case, then for a given point $x \in B_\ell$, the density estimator from the histogram will be

$$\widehat{p}_n(x) = \frac{\text{number of observations within } B_\ell}{n} \times \frac{1}{\text{length of the bin}} = \frac{M}{n}\sum_{i=1}^{n} I(X_i \in B_\ell).$$

The intuition of this density estimator is that the histogram assign equal density value to every points within the bin. So for B_ℓ that contains x, the ratio of observations within this bin is $\frac{1}{n}\sum_{i=1}^n I(X_i \in B_\ell)$, which should be equal to the density estimate times the length of the bin.

Now we study the bias of the histogram density estimator.

$$\begin{aligned}
\mathbb{E}\left(\widehat{p}_n(x)\right) &= M \cdot P(X_i \in B_\ell) \\
&= M \int_{\frac{\ell-1}{M}}^{\frac{\ell}{M}} p(u)du \\
&= M\left(F\left(\frac{\ell}{M}\right) - F\left(\frac{\ell-1}{M}\right)\right) \\
&= \frac{F\left(\frac{\ell}{M}\right) - F\left(\frac{\ell-1}{M}\right)}{1/M} \\
&= \frac{F\left(\frac{\ell}{M}\right) - F\left(\frac{\ell-1}{M}\right)}{\frac{\ell}{M} - \frac{\ell-1}{M}} \\
&= p(x^*), \quad x^* \in \left[\frac{\ell-1}{M}, \frac{\ell}{M}\right].
\end{aligned}$$

The last equality is done by the mean value theorem with $F'(x) = p(x)$. By the mean value theorem again, there exists another point x^{**} between x^* and x such that

$$\frac{p(x^*) - p(x)}{x^* - x} = p'(x^{**}).$$

Thus, the bias

$$\begin{aligned}
\textbf{bias}(\widehat{p}_n(x)) &= \mathbb{E}\left(\widehat{p}_n(x)\right) - p(x) \\
&= p(x^*) - p(x) \\
&= p'(x^{**}) \cdot (x^* - x) \\
&\le |p'(x^{**})| \cdot |x^* - x| \\
&\le \frac{L}{M}.
\end{aligned} \tag{7.4}$$

Note that in the last inequality we use the fact that both x^* and x are within B_ℓ, whose total length is $1/M$, so the $|x^* - x| \le 1/M$. The analysis of the bias tells us that the more bins we are using, the less bias the histogram has. This makes sense because when we have many bins, we have a higher resolution so we can approximate the fine density structure better.

Now we turn to the analysis of variance.

$$\begin{aligned}
\mathsf{Var}(\widehat{p}_n(x)) &= M^2 \cdot \mathsf{Var}\left(\frac{1}{n}\sum_{i=1}^n I(X_i \in B_\ell)\right) \\
&= M^2 \cdot \frac{P(X_i \in B_\ell)(1 - P(X_i \in B_\ell))}{n}.
\end{aligned}$$

By the derivation in the bias, we know that $P(X_i \in B_\ell) = \frac{p(x^*)}{M}$, so the variance

$$\begin{aligned}
\mathsf{Var}(\widehat{p}_n(x)) &= M^2 \cdot \frac{\frac{p(x^*)}{M} \times \left(1 - \frac{p(x^*)}{M}\right)}{n} \\
&= M \cdot \frac{p(x^*)}{n} - \frac{p^2(x^*)}{n}.
\end{aligned} \tag{7.5}$$

The analysis of the variance has an interesting result: the more bins we are using, the higher variance we are suffering.

Now if we consider the MSE, the pattern will be more inspiring. The MSE is

$$\mathsf{MSE}(\widehat{p}_n(x)) = \mathsf{bias}^2(\widehat{p}_n(x)) + \mathsf{Var}(\widehat{p}_n(x)) \leq \frac{L^2}{M^2} + M \cdot \frac{p(x^*)}{n} - \frac{p^2(x^*)}{n}. \tag{7.6}$$

An interesting feature of the histogram is that: *we can choose M, the number of bins.* When M is too large, the first quantity (bias) will be small while the second quantity (variance) will be large; this case is called *undersmoothing*. When M is too small, the first quantity (bias) is large but the second quantity (variance) is small; this case is called *oversmoothing*.

To balance the bias and variance, we choose M that minimizes the MSE, which leads to

$$M_{\mathsf{opt}} = \left(\frac{n \cdot L^2}{p(x^*)}\right)^{1/3}. \tag{7.7}$$

Although in practice the quantity L and $p(x^*)$ are unknown so we cannot chose the optimal M_{opt}, the rule in equation (7.7) tells us how we should change the number of bins when we have more and more sample size. Practical rule of selecting M is related to the problem of *bandwidth selection*, a research topic in statistics.

7.3 Importance Sampling

Let X be a random variable with PDF p. Consider evaluating the following quantity:

$$I = \mathbb{E}(f(X)) = \int f(x)p(x)dx,$$

where f is a known function. In the example of Lecture 2, we are interested in evaluating

$$\int_0^1 e^{-x^3}dx = \mathbb{E}(f(X)),$$

where $f(x) = e^{-x^3}$ and X is a uniform random variable over $[0, 1]$.

Here is how the importance sampling works. We first pick a proposal density (also called sampling density) q and generate random numbers $Y_1, \cdots, Y_N$ IID from q. Then the importance sampling estimator is

$$\widehat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(Y_i) \cdot \frac{p(Y_i)}{q(Y_i)}.$$

When $p = q$, this reduces to the simple estimator that uses sample means of $f(Y_i)$ to estimate its expectation.

Does this estimator a good estimator? Let's study its bias and variance. For the bias,

$$\begin{aligned}
\mathbb{E}(\widehat{I}_N) - I &= \mathbb{E}\left(f(Y_i) \cdot \frac{p(Y_i)}{q(Y_i)}\right) - I \\
&= \int f(y)\frac{p(y)}{q(y)}q(y)dy - I \\
&= \int f(y)p(y)dy - I = 0.
\end{aligned}$$

Thus, it is an unbiased estimator!

How about the variance?

$$\begin{aligned}\mathsf{Var}(\widehat{I}_N) &= \frac{1}{N}\mathsf{Var}\left(f(Y_i)\cdot\frac{p(Y_i)}{q(Y_i)}\right)\\ &= \frac{1}{N}\left\{\mathbb{E}\left(f^2(Y_i)\cdot\frac{p^2(Y_i)}{q^2(Y_i)}\right) - \underbrace{\mathbb{E}^2\left(f(Y_i)\cdot\frac{p(Y_i)}{q(Y_i)}\right)}_{I^2}\right\}\\ &= \frac{1}{N}\left(\int\frac{f^2(y)p^2(y)}{q(y)}dy - I^2\right).\end{aligned}$$

So only the first quantity depends on the choice of proposal density q. Thus, if we have multiple proposal density, say q_1, q_2, q_3, the best proposal will be the one that minimizes the integration $\int\frac{f^2(y)p^2(y)}{q(y)}dy$.

You may be curious about the optimal proposal density (the q that minimizes the variance). And here is a striking result about this optimal proposal density. First, we recall the Cauchy-Scharwz inequality–for any two functions $A(y)$ and $B(y)$,

$$\int A^2(y)dy\int B^2(y)dy \geq \left(\int A(y)B(y)dy\right)^2$$

and the $=$ holds whenever $A(y)\propto\cdot B(y)$ for some constant. One way to think about this is to view them as vectors–for any two vectors u, v, $\|u\|^2\|v\|^2 \geq \|u\cdot v\|^2$ and the equality holds whenever u and v are parallel to each other. Identifying $A^2(y) = \frac{f^2(y)p^2(y)}{q(y)}$ and $B^2(y) = q(y)$, we have

$$\int\frac{f^2(y)p^2(y)}{q(y)}dy\underbrace{\int q(y)dy}_{=1} \geq \left(\int\frac{f^2(y)p^2(y)}{q(y)}q(y)dy\right)^2 = I^2.$$

Namely, this tells us that the optimal choice $q_{\mathsf{opt}}(y)$ leads to

$$\mathsf{Var}(\widehat{I}_{N,\mathsf{opt}}) = \frac{1}{N}\left(I^2 - I^2\right) = 0,$$

a zero-variance estimator! Moreover, the optimal q satisfies

$$\sqrt{\frac{f^2(y)p^2(y)}{q_{\mathsf{opt}}(y)}} = A(y)\propto B(y) = \sqrt{q_{\mathsf{opt}}(y)},$$

implying

$$q_{\mathsf{opt}}(y)\propto f(y)p(y) \Longrightarrow q_{\mathsf{opt}}(y) = \frac{f(y)p(y)}{\int f(y)p(y)dy}. \tag{7.8}$$

This gives us a good news–the optimal proposal density has 0 variance and it is unbiased. Thus, we only need to sample it once and we can obtain the actual value of I. However, even if we know the closed form of $q_{\mathsf{opt}}(y)$, how to sample from this density is still unclear. In the next section, we will talk about a method called *Rejection Sampling*, which is an approach that can tackle this problem.

7.4 Rejection Sampling

Given a density function $f(x)$, the rejection sampling is a method that can generate data points from this density function f.

Here is how one can generate a random variable from f.

1. We first choose a number $M \geq \sup_x \frac{f(x)}{p(x)}$ and a proposal density p where we know how to draw sample from (p can be the density of a standard normal distribution).

2. Generate a random number Y from p and another random number U from $\mathsf{Uni}[0,1]$.

3. If $U < \frac{f(Y)}{M \cdot p(Y)}$, we set $X = Y$. Otherwise go back to the previous step to draw another new pair of Y and U.

The above procedure is called *rejection sampling* (or rejection-acceptance sampling). If we want to generate $X_1, \cdots, X_n$ from f, we can apply the above procedure multiple times until we accept n points.

Does this approach work? Now we consider the CDF of X.

$$\begin{aligned} P(X \leq x) &= P(Y \leq x | \mathsf{accept} Y) \\ &= P\left(Y \leq x | U < \frac{f(Y)}{M \cdot p(Y)}\right) \\ &= \frac{P\left(Y \leq x, U < \frac{f(Y)}{M \cdot p(Y)}\right)}{P\left(U < \frac{f(Y)}{M \cdot p(Y)}\right)}. \end{aligned} \tag{7.9}$$

Note that in the last equality, we used the definition of conditional probability.

For the numerator, using the feature of conditional probability,

$$\begin{aligned} P\left(Y \leq x, U < \frac{f(Y)}{M \cdot p(Y)}\right) &= \int P\left(Y \leq x, U < \frac{f(Y)}{M \cdot p(Y)} | Y = y\right) p(y) dy \\ &= \int P\left(y \leq x, U < \frac{f(y)}{M \cdot p(y)}\right) p(y) dy \\ &= \int I(y \leq x) P\left(U < \frac{f(y)}{M \cdot p(y)}\right) p(y) dy \\ &= \int_{-\infty}^{x} \frac{f(y)}{M \cdot p(y)} p(y) dy \\ &= \frac{1}{M} \int_{-\infty}^{x} f(y) dy \end{aligned}$$

Note that in the fourth equality, we use the fact that the choice of $M : M \geq \sup_x \frac{f(x)}{p(x)}$ ensures

$$\frac{f(y)}{M \cdot p(y)} \leq 1 \quad \forall y.$$

For the denominator, using the similar trick,

$$\begin{aligned} P\left(U < \frac{f(Y)}{M \cdot p(Y)}\right) &= \int P\left(U < \frac{f(Y)}{M \cdot p(Y)} | Y = y\right) p(y)dy \\ &= \int P\left(U < \frac{f(y)}{M \cdot p(y)}\right) p(y)dy \\ &= \int \frac{f(y)}{M \cdot p(y)} p(y)dy \\ &= \frac{1}{M} \int f(y)dy = \frac{1}{M}. \end{aligned}$$

Thus, putting altogether into equation (7.9), we obtain

$$P(X \le x) = \frac{P\left(Y \le x, U < \frac{f(Y)}{M \cdot p(Y)}\right)}{P\left(U < \frac{f(Y)}{M \cdot p(Y)}\right)} = \frac{\frac{1}{M}\int_{-\infty}^{x} f(y)dy}{\frac{1}{M}} = \int_{-\infty}^{x} f(y)dy,$$

which means that the random variable X does have the density f.

Here are some features about the rejection sampling:

- Using the rejection sampling, we can generate sample from any density f *as long as we know the closed form of* f.
- If we do not choose M well, we may reject many realizations of Y, U to obtain a single realization of X.
- There is an upper on M at the first step: $M \ge \sup_x \frac{f(x)}{p(x)}$.
- In practice, we want to choose M as small as possible because a small M leads to a higher chance of accepting Y. To see this, note that the denominator $P\left(U < \frac{f(Y)}{M \cdot p(Y)}\right) = P(\mathsf{Accept} Y) = \frac{1}{M}$. Thus, a small M leads to a large accepting probability.
- If you want to learn more about rejection sampling, I would recommend http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ARM.pdf.

7.4.1 Application in Bayesian Inference

Here we explain how to use the rejection sampling to sample from a posterior distribution. Let $\pi(\theta|X_1, \cdots, X_n)$ be the posterior distribution, $L(\theta|X_1, \cdots, X_n)$ be the likelihood function, and $\pi(\theta)$ be the prior distribution. Also, let $\widehat{\theta}_{MLE} = \mathsf{argmax}_\theta L(\theta|X_1, \cdots, X_n)$ be the MLE.

Then we can generate points from the posterior distribution with the followings:

1. Generate θ from prior distribution π and U from $\mathsf{Uni}(0, 1)$ independently.
2. Accept θ if $U < \frac{L(\theta|X_1, \cdots, X_n)}{L(\widehat{\theta}_{MLE}|X_1, \cdots, X_n)}$.

The θ's that are accepted from the above two steps are IID from the posterior distribution $\pi(\theta|X_1, \cdots, X_n)$.

Why this approach works? Well here is how each quantity corresponds to the ones in rejection sampling (after rescaling):

$$\begin{aligned}\pi(\theta|X_1,\cdots,X_n) &\sim f(x)\\ L(\theta|X_1,\cdots,X_n) &\sim \frac{f(x)}{p(x)}\\ \pi(\theta) &\sim p(x)\\ L(\widehat{\theta}_{MLE}|X_1,\cdots,X_n) &\sim M\end{aligned}$$

Note that the acceptance probability is

$$p_a = \frac{\int L(\theta|X_1,\cdots,X_n)\pi(\theta)d\theta}{L(\widehat{\theta}_{MLE}|X_1,\cdots,X_n)} = \frac{p(X_1,\cdots,X_n)}{L(\widehat{\theta}_{MLE}|X_1,\cdots,X_n)}.$$

So the normalization constant of the posterior distribution $p(X_1,\cdots,X_n)$ can be estimated using

$$\widehat{p}(X_1,\cdots,X_n) = \widehat{p}_a \cdot L(\widehat{\theta}_{MLE}|X_1,\cdots,X_n),$$

where $\widehat{p}_a$ is the empirical acceptance probability.

In the analysis of rejection sampling, there are two random quantities that are often of great interest. The first quantity occurs in the case where we are given a fixed amount of target sample, say m, and we are interested in the number of points we generate to accept m points. Let M be such a value. The behavior of M is something we would like to analyze because the difference $M-m$ informs us the amount of additional points we need to generate to obtain m points. Note that in ideal case, M is a random variable following a negative binomial distribution with parameter (m, p_a).

The other quantity occurs in situations where we have a fixed budget n_0 to generate points. Let N be the number of accepted points. This quantity N is also of research interest because we can use the ratio $\frac{N}{n_0}$ as an estimate of the acceptance probability p_a. Note that in the ideal case, the quantity N follows from a binomial distribution with parameter (n_0, p_a).

7.5 MCMC: Metropolis-Hastings Algorithm

Although rejection sampling is very powerful, it has a limitation that we need to know the MLE $\widehat{\theta}_{MLE}$, which is often a challenge problem when the likelihood function is non-convex. Moreover, in many cases such as Bayesian analysis, we may not know the exact value of the density function $f(x)$ but only know the value up to a constant. This is because the posterior

$$\pi(\theta|X_1,\cdots,X_n) \propto p(X_1,\cdots,X_n|\theta)\cdot\pi(\theta).$$

Evaluating the likelihood value and the prior are simple but computing the integral over θ is hard. It is desirable to have a method that allows us to sample from $\pi(\theta|X_1,\cdots,X_n)$ with only access to $p(X_1,\cdots,X_n|\theta)$ and $\pi(\theta)$.

The Markov Chain Monte Carlo (MCMC) is a tool that allows us to do this. The idea of MCMC is to generate an ergodic Markov chain $\{X_n\}$ from a stationary distribution that is the same as the distribution we want to generate from. In the Bayesian setting, the stationary distribution will be the posterior distribution.

7.5.1 Discrete state case

Let $\pi(\theta)$ be the PDF/PMF that we want to generate from with $\theta \in S$ that is univariate. We start with considering a simple case where the state space S is discrete The **Metropolis-Hastings algorithm** is a simple approach to generate from $\pi(\theta)$ for a univariate θ. It proceeds as follows:

- Input: an initial value $x_0 \in S$ and a proposal function $q(i_0|i_1)$ with $i_0, i_1 \in S$.
- Start with an initial value $X_0 = x_0$.
- For $n = 0, \cdots, N$, do the following:
  1. Simulate a candidate value $Y_n \sim q(y|X_n = i)$, where i is the value of X_n. Suppose that $Y_n = j$.
  2. Compute the Metropolis-Hastings *acceptance probability*:
$$a_{ij} = \min\left\{\frac{\pi_j \times q(i|j)}{\pi_i \times q(j|i)}, 1\right\}$$
  3. Generate $U \sim \mathsf{Uni}(0, 1)$.
  4. Accept the candidate $Y = j$ if $U \leq a_{ij}$, otherwise set $X_{n+1} = X_n$. Namely,
$$X_{n+1} = \begin{cases} Y & \text{if } U \leq a_{ij} \\ X_n & \text{if } U > a_{ij} \end{cases}$$

Proposition 7.1 *Assume that $\pi(i) > 0$ for all $i \in S$ and that $q(i|j) > 0 \Leftrightarrow q(j|i) > 0$ for all $i, j \in S$. Then the Metropolis-Hastings algorithm generates a Markov chain with stationary distribution π.*

Proof: It is easy to see that the sequence $\{X_n\}$ forms a homogeneous Markov chain. Let $\mathbf{P} = \{p_{ij}\}$ be the transition probability matrix of X_n . We will prove this by showing that the detailed balance is statisfied. For the case of $i = j$, it is trivial so we assume $i \neq j$.

For $i \neq j$,
$$p_{ij} = P(X_{n+1} = j|X_n = i) = P(X_1 = j|X_0 = i) = a_{ij}q(j|i).$$
Therefore,
$$\begin{aligned} \pi_i p_{ij} = \pi_i a_{ij} q(j|i) &= \begin{cases} \pi_i q(j|i) \times \frac{\pi_j q(i|j)}{\pi_i q(j|i)} & \text{if } \frac{\pi_j q(i|j)}{\pi_i q(j|i)} \leq 1 \\ \pi_i q(j|i) \times 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} \pi_j q(i|j) & \text{if } \pi_j q(i|j) \leq \pi_i q(j|i) \\ \pi_i q(j|i) & \text{if } \pi_j q(i|j) > \pi_i q(j|i) \end{cases} \end{aligned}$$

Noe we consider $\pi_j p_{ji}$:
$$\begin{aligned} \pi_j p_{ji} = \pi_j a_{ji} q(i|j) &= \begin{cases} \pi_j q(i|j) \times \frac{\pi_i q(j|i)}{\pi_j q(i|j)} & \text{if } \frac{\pi_i q(j|i)}{\pi_j q(i|j)} \leq 1 \\ \pi_j q(i|j) \times 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} \pi_i q(j|i) & \text{if } \pi_i q(j|i) \leq \pi_j q(i|j) \\ \pi_j q(i|j) & \text{if } \pi_i q(j|i) > \pi_j q(i|j) \end{cases}, \end{aligned}$$
which is the same as $\pi_i p_{ij}$.

So π satisfies the detailed balance, it is a stationary distribution. ■

Remark 7.2 *Some remarks:*

- *If we choose $q(i|j)$ such that $\{X_n\}$ is irreducible, $\{X_n\}$ will be positive recurrent by the stationary distribution criterion even on an infinite state-space case. Then we can apply the Ergodic theorem so the average of $\{X_n\}$ converges to the average from stationary distribution. In the case of Bayesian analysis, we can use it to approximate the posterior mean.*
- *The Markov chain $\{X_n\}$ from Metropolis-Hastings algorithm is not necessarily aperiodic so the Basic limit theorem may not work.*

There are many different variants of the Metropolis-Hastings algorithm. For instance,

- Symmetry proposal: $q(i|j) = q(j|i)$. This is the original Metropolis algorithm. The acceptance probability reduces to $a_{ij} = \min\left\{\frac{\pi_j}{\pi_i}, 1\right\}$.
- Independence proposal: $q(j|i) = q(j)$. Note that the resulting sequence $\{X_n\}$ are not IID; it is still a Markov chain.

7.5.2 Continuous state case

Now we consider the case of continuous state space case. Let S be a state space that possibly contains infinite amount of elements. A sequence of RVs $X_0, X_1, \cdots$, is called a Markov chain on S if for all $n \geq 0$ and for all (Borel) set $A \subset S$,

$$P(X_{n+1} \in A \mid X_n, X_{n-1}, \cdots, X_0) = P(X_{n+1} \in A \mid X_n).$$

The Markov chain is homogeneous if

$$P(X_{n+1} \in A \mid X_n) = P(X_1 \in A \mid X_0).$$

The function $K(x, A) = P(X_1 \in A \mid X_0 = x)$ is called the **transition kernel**. Note that if there exist a function $f(x, y)$ such that

$$P(X_1 \in A \mid X_0 = x) = \int_A f(x, y)dy,$$

then $f(x, y)$ is called the **transition kernel density**. The transition kernel density is analogue of the transition probability matrices in the discrete cases.

Many notions from discrete state spaces can be generalized to continuous state spaces such as irreducibility, periodicity, etc. Note that the Chapman-Kolmogorov equation will become

$$K^{m+n}(x, A) = \int_S K^n(y, A)K^m(x, dy),$$

where $K^n(x, A) = P(X_n \in A \mid X_0 = x)$. In this case, a probability distribution Π on S is called a **stationary distribution** of $\{X_n\}$ with transition kernel $K(x, A)$ if for any (Borel) set $B \subset S$,

$$\Pi(B) = \int_S K(x, B)\Pi(dx).$$

Note that the above is the global balance equation. If the transition kernel density exists and Π has a density π, then the global balance equation can be expressed as

$$\pi(y) = \int_S \pi(x)f(x, y)dx.$$

Using these new notations, we define the Metropolis-Hastings algorithm for continuous state spaces as follows. Recall that our goal is to design an algorithm such that $\pi(x)$ is the density of the stationary distribution. The proposal probability $q(j \mid i)$ will now be replaced by the proposal density $q(y \mid x)$.

- Input: an initial value $x_0 \in S$ and a proposal function $q(x|y)$ with $x, y \in S$.
- Start with an initial value $X_0 = x_0$.
- For $n = 0, \cdots, N$, do the following:
  1. Simulate a candidate value $Y_n \sim q(\cdot|X_n)$.
  2. Compute the Metropolis-Hastings *acceptance probability*:
$$a(x, y) = \min\left\{\frac{\pi(y) \times q(x|y)}{\pi(x) \times q(y|x)}, 1\right\}$$
  3. Accept the candidate Y_n with a probability $a(X_n, Y_n)$. If we do not accept, we keep $X_{n+1} = X_n$. Namely,
$$X_{n+1} = \begin{cases} Y_n & \text{with a probability of } a(X_n, Y_n) \\ X_n & \text{with a probability of } 1 - a(X_n, Y_n) \end{cases}$$

To make sure that this is a Markov chain with irreducible state space, we choose the proposal density $q(y|x) > 0$ for all $x, y \in S$. To simplify the problem, here we assume that $S \subset \mathbb{R}$ but you can easily generalize it to higher dimensions. A common example is to use the *random walk* proposal:

$$Y_n = X_n + \epsilon_n,$$

where ϵ_n is some perturbation independent of X_n with $\mathbb{E}(\epsilon_n) = 0$. A concrete example is to choose $\epsilon \sim N(0, \sigma^2)$ for some pre-specified σ^2. In this case,

$$q(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{1}{2\sigma^2}\|x-y\|^2}.$$

When S is a connected set and $\pi(x) > 0$ for all $x \in S$, there is no need to choose a proposal that has infinite support. Instead, we can choose $q(y|x)$ such that there exists $\delta, \epsilon > 0$ so that $q(y|x) > \epsilon$ if $|x - y| < \delta$. This allows the perturbation ϵ_n to have a finite support. For instance, we may choose

$$\epsilon_n \sim \mathsf{Uni}(-\delta, \delta).$$

Note that often people choose the proposal q to be isotropic, namely,

$$q(y|x) = q(\|y - x\|).$$

Both normal perturbation and uniform perturbation are examples leading to an isotropic proposal. Isotropic proposals have an advantage that the update probability becomes very simple. If the value $\frac{\pi(y)}{\pi(x)}$ is greater than or equal to 1 (i.e., the proposed point has a higher value compared to the current point), we move to the proposed point. Otherwise with a probability of the density ratio, we move to the proposed point.

The choice of perturbation is often a challenging question. There are two quantities we want to optimize the MCMC algorithm – the acceptance rate and the exploration rate (speed of exploring the state space S). We want high acceptance rate as well as a high exploration rate but they are often inversely related to each other. To see this, consider the following example.

Example 7.3 *Support that $S = \mathbb{R}$ and our target is a univariate standard normal distribution, i.e.,*

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2).$$

We use the proposal density with a uniform perturbation such that

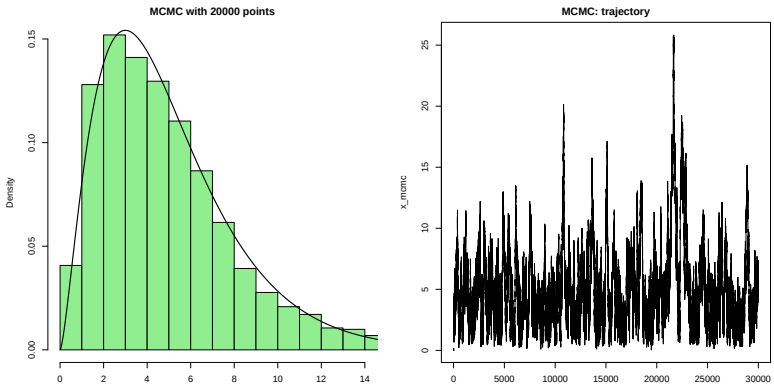
$$q(y|x) = \frac{1}{2\delta} I(|x - y| \leq \delta).$$

Let $U_1 \sim \mathsf{Uni}(-\delta, \delta)$. Then given a current state $x^{(t)}$, the next state will be

$$x^{(t+1)} = \begin{cases} x^{(t)} + U_1 & \text{with a probability } p_t = \min\{\exp\left[\left((x^{(t)})^2 - (x^{(t)} + U_1)^2\right)/2\right], 1\} \\ x^{(t)} & \text{with a probability } 1 - p_t \end{cases}$$

Here, as you can see, if δ is small, the chance that we accept the proposal is generally higher but our exploration of S will be slow (leading to a Markov chain with a high dependence). On the other hand, if δ is large, the chance we accept the proposal is low but we can quickly explore state space.

Example 7.4 (MCMC for a χ^2 distribution) *In the following, we use the MCMC to generate points from a χ^2_5:*

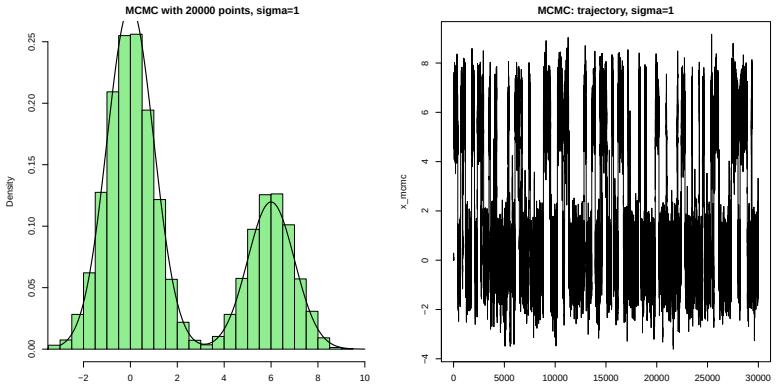


We use $q(y|x) \sim N(x - y, 0.5^2)$ and an initial point 0.5 and run the MCMC to generate 30000 points. Note that generally, the initial point is not important and we will remove the first few points in the MCMC chain (this is also called a burn out). The left panel shows the histogram of the MCMC points and the black curve denotes the true density curve. In the right panel, we display the trajectory of the MCMC; this plot is also called the trace plot. The trace plot will be useful in examining the behavior of the MCMC (we will talk about it later).

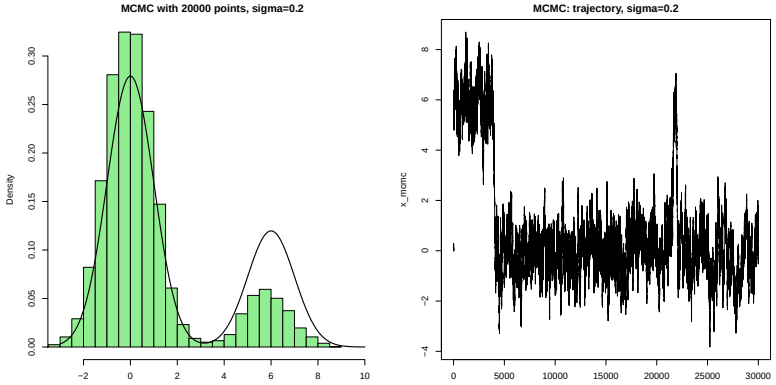
Example 7.5 (MCMC for a Gaussian mixture) *To show that MCMC can be applied a wide class of problem, we implement it to generate points from a Gaussian mixture model with the density*

$$p(x) = 0.7\phi(x; 0, 1) + 0.3\phi(x; 5, 1),$$

where $\phi(x; \mu, \sigma^2)$ is the PDF of a normal distribution with mean μ and variance σ^2. We first consider the case where the proposal $q(y|x) \sim N(x - y, 1)$:

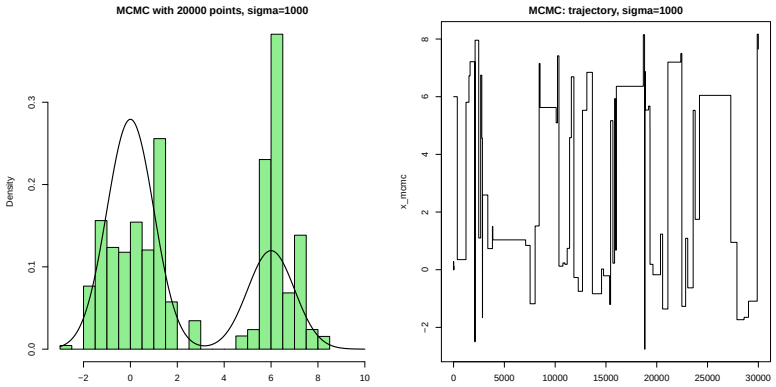


The MCMC did a good job in generating points from the Gaussian mixture. Now we consider an interesting case where we decrease the variance of the proposal $q(y|x)$ to $N(x-y, 0.2^2)$. Here is the result of MCMC after running it 30000 *times:*



We see a biased result in our MCMC! In the left panel, the smaller bump was underestimated. The trajectory plot in the right panel explains what was happening – at the beginning, the MCMC was moving around the second mode (small bump centered at 5*) and then it switches to the big bump and stuck there. If we compare this to the trajectory where we have a proposal with a higher variance, we see that when the proposal has a higher variance, MCMC switches between the two bumps very frequently but when the proposal has a low variance, it does not switch that frequently. In other word, the variance of the proposal determines the speed of mixing in the MCMC. When MCMC has a slow speed of mixing (i.e., variance of the proposal is low), we need to run it a lot longer to obtain a stable result.*

So should we always choose a huge variance in our proposal? Not really. Remembered that in the previous example, we have demonstrated that a high variance proposal may lead to an MCMC with a low acceptance rate. Here is what will happen if we increase the variance of the proposal to 1000*:*



Again, we see a biased result and the trace plot shows several flat line, indicating that the chain was staying in the same value for a long time, which is what we expect when the acceptance probability is low.

7.6 MCMC: Gibbs Sampling (♠♠♠)

Gibbs sampling is an alternative approach to sample from a target PDF/PMF based on the idea of MCMC when the target PDF/PMF is multivariate. The appealing feature of Gibbs sampling is that we only need to know the conditional PDF/PMF of the target rather than the joint PDF/PMF. But we do need to know how to sample from the conditional PDF/PMF of each variable given the others.

To illustrate the idea, we start with discrete state space with two variables x_1, x_2 such that $x_1 \in S_1$ and $x_2 \in S_2$. Our goal is to generate from the target PDF/PMF $\pi(x_1, x_2) = P(X_1 = x_1, X_2 = x_2)$. Let $X_1 = x_1^{(0)}$ and $X_2 = x_2^{(0)}$ be the initial starting point. The Gibbs sampling uses the following iterative updates:

1. $P(X_1 = x_1^{(t+1)} | X_2 = x_2^{(t)}) = \pi(X_1 = x_1^{(t+1)} | X_2 = x_2^{(t)})$.

2. $P(X_2 = x_2^{(t+1)} | X_1 = x_1^{(t+1)}) = \pi(X_2 = x_2^{(t+1)} | X_1 = x_1^{(t+1)})$.

Thus, the transition probability matrix becomes

$$\mathbf{P}(x_1^{(t+1)}, x_2^{(t+1)} | x_1^{(t)}, x_2^{(t)}) = \pi(X_2 = x_2^{(t+1)} | X_1 = x_1^{(t+1)}) \pi(X_1 = x_1^{(t+1)} | X_2 = x_2^{(t)}).$$

An interesting fact is that the resulting Markov chain does not satisfy the detailed balance (so the chain is not reversible) but it does satisfy the global balance. To derive the global balance, we need to show that

$$\pi(x_1, x_2) = \sum_{y_1 \in S_1, y_2 \in S_2} \pi(y_1, y_2) \mathbf{P}(x_1, x_2 | y_1, y_2).$$

The right-hand sided (RHS) equals

$$\begin{aligned}
\mathsf{RHS} &= \sum_{y_1 \in S_1, y_2 \in S_2} \pi(y_1, y_2) \mathbf{P}(x_1, x_2 | y_1, y_2) \\
&= \sum_{y_1 \in S_1, y_2 \in S_2} \pi(y_1, y_2) \pi(X_2 = x_2 | X_1 = x_1) \pi(X_1 = x_1 | X_2 = y_2) \\
&= \sum_{y_1 \in S_1, y_2 \in S_2} \pi(y_1, y_2) \frac{\pi(x_1, x_2)}{\pi(X_1 = x_1)} \frac{\pi(x_1, y_2)}{\pi(X_2 = y_2)} \\
&= \sum_{y_2 \in S_2} \pi(X_2 = y_2) \frac{\pi(x_1, x_2)}{\pi(X_1 = x_1)} \frac{\pi(x_1, y_2)}{\pi(X_2 = y_2)} \\
&= \frac{\pi(x_1, x_2)}{\pi(X_1 = x_1)} \sum_{y_2 \in S_2} \pi(x_1, y_2) \\
&= \pi(x_1, x_2)
\end{aligned}$$

so it satisfies the global balance.

When we have d variables $x_1, \cdots, x_d$, the Gibbs sampler is often done by using a **sequential scan**. Let $x_{1:j} = (x_1, \cdots, x_j)$. Given $x^{(0)} = (x_1^{(0)}, \cdots, x_d^{(0)})$, we update using the following way

1. $P(X_1 = x_1^{(t+1)} | X_{2:d} = x_{2:d}^{(t)}) = \pi(X_1 = x_1^{(t+1)} | X_{2:d} = x_{2:d}^{(t)})$.
2. $P(X_2 = x_2^{(t+1)} | X_1 = x_1^{(t+1)}, X_{3:d} = x_{3:d}^{(t)}) = \pi(X_2 = x_2^{(t+1)} | X_1 = x_1^{(t+1)}, X_{3:d} = x_{3:d}^{(t)})$.
3. $\cdots$
4. $P(X_d = x_d^{(t+1)} | X_{1:(d-1)} = x_{1:(d-1)}^{(t+1)}) = \pi(X_d = x_d^{(t+1)} | X_{1:(d-1)} = x_{1:(d-1)}^{(t+1)})$.

Namely, we keep updating from x_1, then x_2, then all the way to x_d and we always use the latest value of other variables.

In addition to the sequential scan, the **random scan** is also a popular approach. Let $x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d)$. The random scan updates as follows:

1. Randomly select an index $i \in \{1, \cdots, d\}$ from a multinomial distribution.
2. For the selected index i, we update it by $x_i^{(t+1)} \sim \pi(x_i | x_{-i}^{(t)})$ and set $x_{-i}^{(t+1)} = x_{-i}^{(t)}$.

It is very simple to generalize Gibbs sampler to continuous state space. We just replace the PMF in the above by a PDF. Then the sequential scan and random scan can be defined easily.

Remark 7.6 *Some remarks:*

- *Sometimes, we only know the value of $\pi(X_1 = x_1 | X_2 = x_2)$ up to some constant rather than able to directly sample from it. This occurs in the case of computing the posterior distribution of multiple parameters. In this case, we can combine the Metropolis-Hastings algorithm and Gibbs sampler – we use the Metropolis-Hastings algorithm for sampling from $\pi(x_1|x_2)$ and $\pi(x_2|x_1)$ and use the Gibbs sampling to obtain the joint result.*

- *In cases we know how to sample from a conditional PDF/PMF with multiple variables, we do need to update each variable once at a time but we can just update multiple variable in one shot. This is called the block Gibbs sampling.*

Gibbs sampling relies on the conditional PDF/PMF $\pi(x_i|x_{-i})$. This quantity determines the transition probability. So the property of the corresponding Markov chain relies on $\pi(x_i|x_{-i})$ for each i. In Markov chain theory, the irreducibility is an important property and it turns out that if π satisfies the *positivity condition*, then the Gibbs sampler creates a Markov chain that is irreducible. For a density $\pi(x_1, \cdots, x_d)$, it satisfies positivity condition if every marginal density $\pi_i(x_i) > 0$ implies that $\pi(x_1, \cdots, x_d) > 0$.

Proposition 7.7 *If a density $\pi(x_1, \cdots, x_d)$ satisfies the positivity condition, then $\pi(x_i|x_{-i}) > 0$ for any x_i, x_{-i} such that $\pi_i(x_i) > 0$ and $\pi(x_{-i}) > 0$.*

The positivity condition implies that the support of the joint density is the Cartesian product of the support of the marginals. This proposition further shows that if the target density satisfies positivity condition, then the Gibbs sampler is irreducible. Note that positivity is sufficient for irreducibility but not necessary.

Example 7.8 (Linear regression) *We demonstrate the use of Gibbs sampling in Bayesian inference for a linear model. Let $\mathbf{X} = (X_1, \cdots, X_n)$ and $\mathbf{Y} = (Y_1, \cdots, Y_n)$ be the covariate and the response and $X_i \in \mathbb{R}^d$. We assume that the model*

$$\mathbf{Y}|\mathbf{X}, \beta, \sigma^2 \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I}_d),$$

where β and σ^2 are the parameter of interest (regression coefficient and the errors). We use a Gaussian prior on β and a Gamma prior on σ^{-2} (inverse of variance):

$$\beta \sim N(\mathbf{m}, \mathbf{V}), \sigma^{-2} \sim \Gamma(\alpha_0, \beta_0).$$

Note that this prior is conjugate with the normal linear model which will simplify our calculation a lot. The corresponding posterior distribution is

$$\begin{aligned}
\pi(\beta, \sigma^2|\mathbf{Y}, \mathbf{X}) &\propto L_n(\beta, \sigma^2) \cdot \pi(\beta) \cdot \pi(\sigma^2) \\
L_n(\beta, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)\right\} \\
\pi(\beta) &= \frac{1}{(2\pi)^{d/2}\mathsf{det}(\mathbf{V})} \exp\left\{-(\beta - \mathbf{m})^T\mathbf{V}^{-1}(\beta - \mathbf{m})\right\} \\
\pi(\sigma^2) &= \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}(\sigma^{-2})^{(\alpha_0 - 1)}e^{-\frac{\beta_0}{\sigma^2}}
\end{aligned}$$

Now we are going to demonstrate the power of Gibbs sampler. First given $\sigma^{2(t)}$, we update $\beta^{(t+1)}$ according to the conditional distribution given by the posterior:

$$\begin{aligned}
\pi(\beta|\mathbf{X}, \mathbf{Y}, \sigma^{2(t)}) &\propto L_n(\beta, \sigma^{2(t)}) \cdot \pi(\beta) \\
&\propto \exp\left\{-\frac{1}{2\sigma^{2(t)}}(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)\right\} \cdot \exp\left\{-(\beta - \mathbf{m})^T\mathbf{V}^{-1}(\beta - \mathbf{m})\right\} \\
&\sim N(\mathbf{m}^*, \mathbf{V}^*), \\
\mathbf{m}^* &= \mathbf{W}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} + (\mathbf{I}_n - \mathbf{W})\mathbf{m} \\
\mathbf{W} &= (\mathbf{X}^T\mathbf{X} + \mathbf{V}^{-1}\sigma^{(t)})^{-1}\mathbf{X}^T\mathbf{X} \\
\mathbf{V}^* &= \mathbf{W}(\mathbf{X}^T\mathbf{X})^{-1}\sigma^{2(t)}
\end{aligned}$$

So we just draw a new random vector $\beta^{(t+1)}$ from the above normal distribution. Then we update the variance $\sigma^{2(t+1)}$ by drawing its inverse from

$$\begin{aligned}\pi(\sigma^{-2}|\mathbf{X},\mathbf{Y},\beta^{(t+1)}) &\propto L_n(\beta^{(t+1)},\sigma^2)\cdot\pi(\sigma^2)\\ &= \frac{1}{(2\pi\sigma^2)^{n/2}}\exp\left\{-\frac{1}{2\sigma^2}(\mathbf{Y}-\mathbf{X}\beta^{(t+1)})^T(\mathbf{Y}-\mathbf{X}\beta^{(t+1)})\right\}\cdot(\sigma^{-2})^{(\alpha_0-1)}e^{-\frac{\beta_0}{\sigma^2}}\\ &\sim \Gamma\left(\alpha_0+\frac{n}{2},\beta_0+\frac{1}{2}(\mathbf{Y}-\mathbf{X}\beta^{(t+1)})^T(\mathbf{Y}-\mathbf{X}\beta^{(t+1)})\right).\end{aligned}$$

Thus, given an initial choice of $\beta^{(0)}$ and $\sigma^{2(0)}$, we update β and then update σ^2 and iterates this procedure several times. Then the pairs

$$\{(\beta^t,\sigma^{2(t)}) : t=\tau,\tau+1,\cdots\}$$

are points from the posterior distribution $\pi(\beta,\sigma^2|\mathbf{Y},\mathbf{X})$. Note that we burn out the points before time point τ to make sure the chain is stable. We can then use them to find the posterior mean and MAP and construct credible interval.

7.7 Convergence analysis

Consider the case that we apply an MCMC and obtain a set of points $Y_1,\cdots,Y_n$ from the stationary distribution π. One can think of π as the posterior distribution that we are generating from. Assume that we want to infer the quantity $h_0=\mathbb{E}(h(Y))$ when $Y\sim\pi$. One example is the posterior mean – in this case $h(x)=x$. Note that marginally each Y_i is from the stationary distribution.

From ergodic theory, we can use the average

$$\widehat{h}_n=\frac{1}{n}\sum_{i=1}^n h(Y_i)$$

as an estimator of h_0. We know that $\mathbb{E}(h(Y_i))=h_0$ so it is an unbiased estimator and ergodic theory implies that $\widehat{h}_n\overset{a.s.}{\to}h_0$. But this does not tell us about how fast $\widehat{h}_n$ converges to h_0.

Since it is an unbiased estimator, the mean square error reduces to variance. So we now explore the variance of $\widehat{h}_n$. Be ware, the variance of $\widehat{h}_n$ is NOT $\frac{1}{n}\mathsf{Var}(h(Y_i))$ because $Y_1,\cdots,Y_n$ are not IID – they form a Markov chain. In general, the variance can be written as

$$\mathsf{Var}(\widehat{h}_n)=\frac{1}{n^2}\left\{\sum_{i=1}^n\mathsf{Var}(h(Y_i))+2\sum_{i<j}\mathsf{Cov}(h(Y_i),h(Y_j))\right\}.$$

The covariance part is what makes the variance hard to approximate.

Let $h_i = h(Y_i)$. The variance has the following decomposition:

$$\begin{aligned}
\mathsf{Var}(\widehat{h}_n) &= \frac{1}{n^2}\left\{\sum_{i=1}^{n}\mathsf{Var}(h(Y_i)) + 2\sum_{i<j}\mathsf{Cov}(h(Y_i),h(Y_j))\right\} \\
&= \frac{1}{n^2}\left\{n\mathsf{Var}(h_1) + 2\sum_{i=1}^{n-1}\mathsf{Cov}(h_i,h_{i+1}) + 2\sum_{i=1}^{n-2}\mathsf{Cov}(h_i,h_{i+2}) + \cdots\right\} \\
&= \frac{\mathsf{Var}(h_1)}{n}\left\{1 + 2\frac{n-1}{n}\mathsf{Corr}(h_1,h_2) + 2\frac{n-2}{n}\mathsf{Corr}(h_1,h_3) + \cdots\right\} \\
&\approx \frac{\mathsf{Var}(h_1)}{n}\left\{1 + 2\mathsf{Corr}(h_1,h_2) + 2\mathsf{Corr}(h_1,h_3) + \cdots\right\} \\
&= \frac{\mathsf{Var}(h_1)}{n}\left\{1 + 2\sum_{i=1}^{\infty}\mathsf{Corr}(h_1,h_{i+1})\right\} \\
&= \frac{\mathsf{Var}(h_1)}{n}\kappa_h
\end{aligned}$$

as $n \to \infty$ and $\kappa_h = 1 + 2\sum_{i=1}^{\infty}\mathsf{Corr}(h_1, h_{i+1})$ is the variance inflation factor.

Therefore, $\mathsf{Var}(\widehat{h}_n) = \frac{\mathsf{Var}(h_1)}{n_{\mathsf{eff}}}$, where $n_{\mathsf{eff}} = n/\kappa_h$ is called the *effective sample size*. Here, from the construction of κ_h, you see that if the dependency of the Markov chain is large, κ_h will be large and the effective sample size is small. Note that κ_h can be estimated using spectral analysis for time series and if we have an estimate $\widehat{\kappa}_h$, we can obtain a variance estimator $\widehat{\mathsf{Var}}(\widehat{h}_n) = \frac{1}{\widehat{n}_{\mathsf{eff}}}\widehat{\mathsf{Var}}(h_1)$, where $\widehat{\mathsf{Var}}(h_1)$ is simply the sample variance of $h_1, \cdots, h_n$.

7.8 Hamiltonian Monte Carlo (♠♠♠)

The Hamiltonian Monte Carlo (HMC) is a new MCMC approach that has been shown to work better than the usual MH algorithm. It is based on the idea of Hamiltonian dynamics.

The high-level idea of HMC is to generate a proposal from a *better proposal distribution*' and modify the acceptance part so the it has a *higher acceptance rate*. In the usual MH algorithm, we are directly sample from a proposal density $q(y|x)$. The HMC modifies this process using two components: a random momentum (velocity) vector ω and the Hamiltonian dynamics. The momentum is required for every coordinate of the position x. Thus, if $x \in \mathbb{R}^d$, then we also need a vector of d elements for the momentum. As the name suggests, the momentum vector determines how we move x during the dynamics. The randomness is due to the random momentum vector (and the later acceptance part).

The rough idea of one-run HMC is as follows. Starting at the location x_0:

1. (Proposal step 1) We draw a random momentum vector $\omega_0 \sim p(\omega) \propto e^{-V(\omega)}$, where $V(\omega)$ is called the kinematic energy. Often $p(\omega)$ is taken to be a multivariate Gaussian.
2. (Proposal step 2) Then we apply the Hamiltonian dynamics at location x_0 and velocity ω_0 with the Hamiltonian (energy) $H(x,\omega) = -\log\pi(x) + V(\omega)$ and let the dynamics run for time T. This changes (x_0,ω_0) to (x_T,ω_T). Note that the pair (x,ω) is called the state.
3. (Acceptance step) We accept the new location x_T with a probability of

$$a(x_0,\omega_0,x_T,\omega_T) = \min\left\{1, \frac{\exp(-H(x_T,\omega_T))}{\exp(-H(x_0,\omega_0))}\right\}.$$

To approximate the distribution of π, we will iterate the HMC several times.

Note that in the second step (Hamiltonian dynamics), the dynamics is deterministic. Namely, if we start with the same location and the same momentum, we always end up being in the same destination. So for HMC the proposal density $q(x_T|x_0)$ is determined by the density $p(\omega) \propto e^{-V(\omega)}$ and the initial location x_0

To understand what happen in the HMC, we first note that the Hamiltonian contains two parts.

Potential energy. The targeted density $\pi(x)$ is incorporated into the HMC through the Hamiltonian

$$H(x,\omega) = \underbrace{-\log \pi(x)}_{=U(x)} + V(\omega).$$

The quantity $U(x) = -\log \pi(x)$ is also known as the potential energy.

Kinematic energy. The momentum is drawn from the Kinematic energy. The density $p(\omega) \propto e^{-V(\omega)}$ is crucial in the performance of an HMC algorithm. The function V has to be coordinatewise symmetric to ensure the detailed balance equation. In general, we will choose $V(\omega) = \sum_{j=1}^{d} \frac{\omega_j^2}{2m_j}$, where d is the dimension of x and m_j is called the mass of the j-th coordinate. Although this looks fancy, but it implied an extremely sample distribution of $p(\omega)$:

$$p(\omega) \propto e^{-\frac{1}{2}\omega^T M^{-1}\omega} \sim N(0, M),$$

where $M = \mathsf{diag}(m_1, \cdots, m_d)$. So in fact, we are generating the momentum from a multivariate Gaussian (and all coordinates are independent).

Hamiltonian dynamics. The Hamiltonian dynamics governs the usual motion of an object under a specified potential energy and the kinematic energy. It provides excellent description on many physical phenomena such as how planets orbiting around a star. When $H(x,\omega)$ is given, the Hamiltonian dynamics is a deterministic equation of motion. Suppose we start with a location $x(0)$ and a momentum $\omega(0)$, the trajectory of the state $\{x(t), \omega(t) : t \in [0,\infty)\}$ is determined by

$$\frac{dx_j(t)}{dt} \equiv x_j'(t) = \frac{\partial H(x(t),\omega(t))}{\partial \omega_j(t)}, \quad \frac{d\omega_j(t)}{dt} \equiv \omega_j'(t) = -\frac{\partial H(x(t),\omega(t))}{\partial x_j(t)}$$

for $j = 1, \cdots, d$. A powerful feature of Hamiltonian dynamics is that

> *Even if we only have access to $r(x) \propto \pi(x)$, we can still compute the dynamics since the potential energy $U(x) = -\log \pi(x) = -\log r(x) + C_0$ for some constant C_0. We can totally ignore the constant C_0 in practice.*

Because kinematic energy is $V(\omega) = \sum_{j=1}^{d} \frac{\omega_j^2}{m_j}$, the change in location (in j-th coordinate) is simply

$$x_j'(t) = \frac{\omega_j(t)}{m_j}.$$

Thus, given an initial state $(x(0), \omega(0)) = (x_0, \omega_0)$, after running the Hamiltonian dynamics for time T, we will move to the state $(x(T), \omega(T)) = (x_T, \omega_T)$. The new state (x_T, ω_T) is the new proposal. Then in the HMC, we will make a acceptance decision to see if we will accept this proposal.

Here is one caveat in the HMC:

> *No matter we accept or reject the proposal, we will draw a new momentum in the next iteration.*

Namely, only the location x_T will be kept after this iteration. The momentum will be deleted and we will draw a new momentum (from the kinematic energy) without using any information from the previous iteration.

Potential informs momentum: better proposal. The Hamiltonian dynamics allow the target density $\pi(x)$ changes the momentum vector via the equation

$$\omega_j'(t) = -\frac{\partial H(x(t), \omega(t))}{\partial x_j(t)} = \omega_j'(t) = -\frac{\partial \log \pi(x(t))}{\partial x_j(t)}.$$

Thus, even if the original momentum $\omega(0)$ may be pointing toward a bad direction (with least density), the dynamics will adjust its orientation so that it tends to point toward a higher density area.

7.8.1 Features of the HMC

- **(P1): Energy conservation of the Hamiltonian dynamics: high acceptance rate.** The Hamiltonian dynamics has a powerful property called *energy conservation*, which implies that the *acceptance probability is very high.* It is not hard to see that the change of Hamiltonian energy with respect to time is

$$\begin{aligned}\frac{dH(x(t), \omega(t))}{dt} &= \sum_{j=1}^{d} \left\{ \frac{\partial H(x(t), \omega(t))}{\partial \omega_j(t)} \frac{d\omega_j(t)}{dt} + \frac{\partial H(x(t), \omega(t))}{\partial x_j(t)} \frac{dx_j(t)}{dt} \right\} \\ &= 0.\end{aligned}$$

  Namely, the Hamiltonian energy will always stay the same during the dynamics. This is a powerful property! Now we examine the acceptance probability:

$$a(x_0, \omega_0, x_T, \omega_T) = \max \left\{ 1, \frac{\exp(-H(x_T, \omega_T))}{\exp(-H(x_0, \omega_0))} \right\}.$$

  The acceptance probability uses the ratio between the initial Hamiltonian energy and the final Hamiltonian energy after applying the dynamics. Because the Hamiltonian energy is conserved during the dynamics, this ratio will always be 1! Namely, *the acceptance probability is* 1 if we apply a real Hamiltonian dynamics. In fact, we need this acceptance step because in practice, we are using a numerical approximation to the Hamiltonian dynamics so there could be some energy loss due to the approximation. So the final acceptance step is to account for this numerical error.

- **(P2): Unique trajectory.** Given initial conditions $x(0) = x_0, \omega(0) = \omega_0$, the Hamiltonian dynamics creates a unique trajectory $(x(t), \omega(t))$. So the only randomness in the HMC is the initial velocity ω_0.

- **(P3): Time-reversal.** Suppose the Hamiltonian dynamics starts at $x(0) = a, \omega(0) = u$ and at time T we obtain $x(T) = b, \omega(T) = w$. Then we have a reversed-time result that if we start the dynamics at $x(0) = b, \omega(0) = -w$, we will obtain $x(T) = a, \omega(T) = u$.

7.8.2 HMC and detailed balance.

Here we have seen that the HMC tends to give a better proposal and have a high acceptance rate. But to make sure we are indeed sampling from the desired density, we need to show that the generated points converge to a stationary distribution that is the desired density π. First, it is easy to see that the generated points form a Markov chain since in each iteration, we only use the information from the previous location. So we only need to show that π satisfies the detailed balance equation of the transition under HMC. In the

HMC (that we indeed perform the Hamiltonian dynamics), since the dynamics is deterministic (property (P2)), given the time T being fixed, the mapping

$$(x_0, \omega_0) \to (x_T, \omega_T)$$

is deterministic. Namely, there exists ϕ_1, ϕ_2 such that $x_T = \phi_1(x_0, \omega_0)$ and $\omega_T = \phi_2(x_0, \omega_0)$. An interesting fact about Hamiltonian dynamics is that if we reverse the time, the trajectory will remain the same (property (P3)). Namely, if we start the dynamics with initial location x_T and momentum $-\omega_T$, after time T we will come back to x_0 and ω_0. Namely, $x_0 = \phi_1(x_T, -\omega_T), \omega_0 = \phi_2(x_T, -\omega_T)$. Thus, there is a one-one correspondence between $(x_0, \omega_0) \leftrightarrow (x_T, -\omega_T)$.

To show the detail balanced, we need to show that

$$\pi(x)p(x \to y) = \pi(y)p(x \to x),$$

where $p(x \to y)$ is the transition density.

Here is an intuitive explanation about the detailed balanced. Suppose that there is only one ω such that $\phi_1(x, \omega) = y$ and let $\tilde{\omega} = \phi_2(x, \omega)$ be the corresponding velocity. Then we also have $\phi_1(y, -\tilde{\omega}) = x$ and $\phi_2(y, -\tilde{\omega}) = \omega$. Thus, there is also only one η such that the dynamics moves (y, η) to (x, ω) and the choice is $\eta = -\tilde{\omega}$.

In this case, $p(x \to y) = p(\omega)$ because ω is the only choice that moves x into y. Similarly, we have $p(y \to x) = p(-\tilde{\omega})$. Then

$$\begin{aligned}
\pi(x)p(x \to y) &= \pi(x)p(\omega) \\
&= \frac{1}{Z_0} \exp\{-U(x) - V(\omega)\} \\
&= \frac{1}{Z_0} \exp\{-H(x, \omega)\} \\
&= \frac{1}{Z_0} \exp\{-H(y, \tilde{\omega})\} \qquad \text{(Energy conservation)} \\
&= \frac{1}{Z_0} \exp\{-U(y) - V(\tilde{\omega})\} \\
&= \pi(y)p(\tilde{\omega}) \\
&= \pi(y)p(-\tilde{\omega}) \qquad (\omega \text{ is coordinatewise symmetric}) \\
&= \pi(y)p(y \to x)
\end{aligned}$$

so the detailed balance is satisfied. Actually, this idea can be generalized to the case where we have more than one momentum leading to y; there is always a one-one correspondence between ω and $\tilde{\omega}$ and the detailed balance is always satisfied.

7.8.3 The HMC algorithm and the leapfrog method

The practical usage of the HMC involves a discretized step of the dynamics. This discretization is called the leapfrog method. Suppose that x_0 is the input location and we are only able to evaluate $r(x) \propto \pi(x)$. Also, let ϵ be the step size in the discretization and L is the number of updates in the dynamics. Namely, $\epsilon \cdot L = T$ is the time that we apply the dynamics.

1. Generate the initial momentum $\omega_0 \sim N(0, M^{-1})$.
2. Set $x^{(0)} = x_0$.

3. For the momentum, make a half-step update:

$$\omega^{(0)} = \omega_0 - \frac{\epsilon}{2}\nabla \log r(x^{(0)}).$$

4. For $\ell = 1, \cdots, L-1$, do the followings:
   (a) Update position: $x^{(\ell)} = x^{(\ell-1)} + \epsilon \cdot \omega^{(\ell-1)}$.
   (b) Update momentum: $\omega^{(\ell)} = \omega^{(\ell-)} - \epsilon \cdot \nabla \log r(x^{(\ell)})$.
5. Make one last update on the position: $x^{(L)} = x^{(L-1)} + \epsilon \cdot \omega^{(L-1)}$.
6. Make another half-step update of the momentum:

$$\omega^{(L)} = \omega^{(L-1)} - \frac{\epsilon}{2}\nabla \log r(x^{(L)}).$$

7. Compute the acceptance probability:

$$a(x_0, \omega_0, x^{(L)}, \omega^{(L)}) = \min\left\{1, \frac{\exp(-H(x^{(L)}, \omega^{(L)}))}{\exp(-H(x_0, \omega_0))}\right\}.$$

8. Accept $x_{\text{new}} = x^{(L)}$ with a probability of $a(x_0, \omega_0, x^{(L)}, \omega^{(L)})$. If we reject, then $x_{\text{new}} = x_0$.
9. Return x_{new}.

A practical challenge is how do we numerically approximate the dynamics part in the HMC algorithm. In the dynamics, the momentum and the location are updated simultaneously. But in practice, we have to make a choice on which one to update first. This leads to a problem that the algorithm is non-symmetric with respect to time. To see this, suppose that we start at x_1 with a momentum ω_1 and move to x_2 with a momentum ω_2. The actual Hamiltonian dynamics is time-reversible, meaning that if we apply the algorithm to $(x_2, -\omega_2)$, we will get back (x_1, ω_1). However, if we only use the leapfrog procedure (step 4), we will not move $(x_2, -\omega_2)$ back to (x_1, ω_1). So the `half step update` (step 3) before and after the `for` loop (step 6) is to resolve this problem and make the algorithm symmetric with respect to time.

Here is an `R` code for the HMC from https://arxiv.org/pdf/1206.1901.pdf, and excellent introduction to the HMC.

```
HMC = function (U, grad_U, epsilon, L, current_q)
{
  q = current_q
  p = rnorm(length(q),0,1)  # independent standard normal variates
  current_p = p
  # Make a half step for momentum at the beginning
  p = p - epsilon * grad_U(q) / 2
  # Alternate full steps for position and momentum
  for (i in 1:L)
  {
    # Make a full step for the position
    q = q + epsilon * p
    # Make a full step for the momentum, except at end of trajectory
    if (i!=L) p = p - epsilon * grad_U(q)
  }
```

```
  # Make a half step for momentum at the end.
  p = p - epsilon * grad_U(q) / 2
  # Negate momentum at end of trajectory to make the proposal symmetric
  p = -p
  # Evaluate potential and kinetic energies at start and end of trajectory
  current_U = U(current_q)
  current_K = sum(current_p^2) / 2
  proposed_U = U(q)
  proposed_K = sum(p^2) / 2
  # Accept or reject the state at end of trajectory, returning either
  # the position at the end of the trajectory or the initial position
  if (runif(1) < exp(current_U-proposed_U+current_K-proposed_K))
  {
    return (q)  # accept
  }
  else {
    return (current_q)  # reject
  }
}
```

Although the HMC works well theory, it requires tuning a couple of parameters for practical use. The choice of step size ϵ and the number of updates L and the mass M will all affect the final performance. Detailed discussion about the tuning can be found in Section 5.4 of https://arxiv.org/pdf/1206.1901.pdf.

7.9 Langevin dynamics and score-based diffusion (♠♠♠)

In modern machine learning and artificial intelligence, the MCMC and HMC are not popular methods since they do not scale very well with respect to the dimension. The modern state-of-the-art is the *diffusion models*, which based on the idea of *Langevin dynamics* and Markov chains.

Suppose we want to sample from a PDF $p(x) \propto f(x)$ such that the gradient $g(x) = \nabla_x \log f(x)$ can be evaluated everywhere. The Langevin dynamics operates via the following Markov chain: we start at an initial point x_0 and then iterates the following equation:

$$x_{t+1} = x_t + \epsilon g(x_t) + \sqrt{2\epsilon} z_t, \tag{7.10}$$

where $z_1, \cdots$ are IID $N(0, \mathbf{I}_d)$ and $\epsilon > 0$ is a tiny number representing the *stepsize*.

From Langevin dynamics theory, when $\epsilon \to 0$, the sequence $\{x_0, x_1, x_2, \cdots\}$ behaves like a sequence of random variables from $p(x)$ since it is a Markov chain with a stationary distribution approaching $p(x) \propto f(x)$.

7.9.1 Score-based diffusion

In the score-based diffusion model, the objective aims at learning the score function of the data distribution, defined as the gradient of the log-probability density with respect to the data.

$$s(x) = \nabla_x \log p(x), \tag{7.11}$$

where $x \in \mathbb{R}^d$. Note that this score function is *NOT the conventional score function in likelihood inference*; in likelihood inference, the score is the gradient of the log-likelihood with respect to parameter θ, not the variable x.

In practice, we model the score function via a neural net $f_\theta(x) \in \mathbb{R}^d$ where θ is learned by the following risk minimization:

$$\begin{aligned}\min_\theta \int \|f_\theta(x) - s(x)\|^2 p(x)dx &= \min_\theta \int [\|f_\theta(x)\|^2 + 2\nabla_x \cdot f_\theta(x)]p(x)dx + C \\ &= \min_\theta \mathbb{E}\left[\|f_\theta(X)\|^2 + 2\nabla_x \cdot f_\theta(X)\right] + C,\end{aligned}$$

where the first equality follows from the integration by parts. With the above objective function, estimating θ can be done via a simple empirical risk minimization:

$$\widehat{\theta} = \mathsf{argmin}_\theta \frac{1}{n}\sum_{i=1}^n \|f_\theta(X_i)\|^2 + 2\nabla_x \cdot f_\theta(X_i).$$

Once we have a good approximation of the score $s(x)$, we can generate samples using Langevin dynamics. This is an iterative process that follows the gradient of the log density:

$$x_{t+1} = x_t + \frac{\epsilon}{2}s(x_t) + \sqrt{\epsilon}z_t, \quad z_t \sim N(0, I) \tag{7.12}$$

The stationary distribution of this process (when $\epsilon \to 0$ and $t \to \infty$) is the target data distribution $p(x)$. The model objective is to train a neural network f_θ to approximate this score function.

Note that there is another family of diffusion models called *denoising diffusion models.* See the following note for an introduction:

> Chen, Y. C. (2025). A Frequentist Statistical Introduction to Variational Inference, Autoencoders, and Diffusion Models. arXiv preprint arXiv:2510.18777. https://arxiv.org/abs/2510.18777

7.9.2 Conditional generation

An interesting feature of the score-based diffusion is that the generation of a conditional model can be done in a cool way. To build a conditional model that can sample from $p(x|y)$ (e.g., generate an image x given a text label y), we need the conditional score $s(x|y) = \nabla_x \log p(x|y)$. Using Bayes' rule, this can be decomposed:

$$\underbrace{s(x|y)}_{\text{Conditional Score}} = \underbrace{s(x)}_{\text{Unconditional Score}} + \underbrace{\nabla_x \log p(y|x)}_{\text{Classifier Gradient}} \tag{7.13}$$

The unconditional score $s(x)$ can be learned by a standard pre-trained diffusion model. The second term can be provided by training a separate classifier that predicts y from a noisy input x and then using its gradients to guide the diffusion process.