MONTE CARLO METHODS

MARIA CAMERON

Contents

1. Introduction

Monte Carlo methods are those where one evaluates something nonrandom using pseudorandom numbers. More precisely, one evaluates a nonrandom quantity as the expected value of a random variable. On the contrary, *simulations* produce random variables with a certain distribution without the purpose of evaluating some expected value.

Typically, the error in Monte-Carlo methods decays as $n^{-1/2}$ where n is the number of samples that is worse than the error decay rate in most deterministic methods (it is usually at least as good as n^{-1}). So, why bother? The reason is that, in many cases, deterministic

methods cannot be used due to the "curse of dimensionality" or the large scale of the problem. In some of these cases, Monte-Carlo methods can be efficient.

For example, to find the mean magnetization in a 3D Ising model with N sites, one needs to average the value of the magnetization over 2^N different spin configurations. If we are considering a 3D $10 \times 10 \times 10$ grid, then $N = 1000$, and $2^{1000} \sim 10^{301}$, a huge number, that makes the deterministic calculation infeasible. On the contrary, a Monte Carlo calculation gives an accurate enough estimate in a reasonable time.

2. Pseudorandom numbers

Pseudorandom numbers are generated by pseudorandom number generators. A pseudorandom number generator produces a deterministic sequence of numbers starting from a seed state that can be specified by the user. Good pseudorandom number generators produce sequences that cannot be distinguished from random numbers by simple tests. In C, the operators `rand()` and `random()` produce a uniformly distributed pseudorandom number in the interval [0 . . . `RAND_MAX`], where `RAND_MAX` is a constant defined in the library `stdlib.h`. It is platform-dependent. In most modern computers,

$$\texttt{RAND\_MAX} = 2^{31} - 1 = 2147483647,$$

the maximal `int` in C. (The range of `int` in C is from -2^{31} to $2^{31}-1$.) However, the function `rand()` produces a periodic sequence with period `RAND_MAX` while the sequence produced by `random()` is indistinguishable from the sequence of random numbers uniformly distributed between 0 and `RAND_MAX` for all practical purposes to the best of my knowledge. Below is a C program generating pseudorandom numbers using `random()` and `rand()` and its output. Note that the sequence generated by `rand()` is periodic with period `RAND_MAX` while the sequence generated by `random()` is not. Therefore, you can use `rand()` only if you need the number of samples significantly less that `RAND_MAX`. If `RAND_MAX`] $= 2147483647 \sim 2 \cdot 10^9$ as above, do not use `rand()` if you you need more that 10^7 samples.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(void);

 int main() {
  int r;
  int i;
  printf("RAND_MAX = %i\n",RAND_MAX); // 2^31 - 1
  printf("Using random():\n");
  for(i = 0; i < RAND_MAX-1; i++) {
  r = random();  /* Generate a random integer */
  if( i < 10 ) printf("%i\n", r);
  }
  printf("\n");
```

```
  for(i = 0; i < RAND_MAX-1; i++) {
  r = random();  /* Generate a random integer */
  if( i < 10 ) printf("%i\n", r);
  }
  /* ... */
  printf("\nUsing rand()\n");
  for(i = 0; i < RAND_MAX-1; i++) {
  r = rand();  /* Generate a random integer */
  if( i < 10 ) printf("%i\n", r);
  }
  printf("\n");
  for(i = 0; i < RAND_MAX-1; i++) {
  r = rand();  /* Generate a random integer */
  if( i < 10 ) printf("%i\n", r);
  }
  return 1;
}

Marias-iMac:Desktop mariacameron$ gcc RandomNumbers.c -lm -O3
Marias-iMac:Desktop mariacameron$ ./a.out
RAND_MAX = 2147483647
Using random():
1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421

377605215
52479496
850182889
2103788022
905904603
1692932299
1981079694
174340263
1245720282
1365390958
```

```
Using rand()
16807
282475249
1622650073
984943658
1144108930
470211272
101027544
1457850878
1458777923
2007237709

16807
282475249
1622650073
984943658
1144108930
470211272
101027544
1457850878
1458777923
2007237709
Marias-iMac:Desktop mariacameron$
```

In order to generate a uniformly distributed random variable on $(0,1)$, i.e., $\eta \sim \mathcal{U}(0,1)$, we code:

```
eta = (double)random()/RAND_MAX;
```

Matlab is excellent for programming prototypes of algorithms. It has a number of tools to generate pseudorandom numbers:

- Function `rand` generates a random variable uniformly distributed in (0,1).
- Function `randi` generates a random integer uniformly distributed in the provided interval.
- Function `randn` generates standard normal random variables, i.e., Gaussian random variables with mean 0 and variance 1.
- Function `randperm` generates a random permutation of numbers from 1 to `n` where `n` is user-supplied.

For more details, read
https://www.mathworks.com/help/matlab/random-number-generation.html.

Python has a lot of resources for generating random numbers and sampling from commonly used distributions.

- The module `random`: https://docs.python.org/3/library/random.html.

- Pseudorandom numbers in NumPy: https://numpy.org/doc/stable/reference/random/index.html.
- statistical tools in SciPy: https://docs.scipy.org/doc/scipy/reference/stats.html.

3. Sampling random variables with given distribution

Typically, the goal of sampling is to evaluate an expected value of a random variable (RV) η with a given pdf $f_\eta(x)$, i.e., the integral of the form

$$E[\eta] = \int x f_\eta(x) dx.$$

If η is a vector random variable belonging to $\mathbb{R}^d$ with $d > 3$, the direct integration using quadrature rules can be too expensive and too memory-demanding. Then Monte Carlo integration becomes a reasonable choice. If we are able to generate N independent samples of η, and η has a finite variance σ^2, we can estimate $E[\eta]$ as follows:

$$E[\eta] = \int x f_\eta(x) dx \approx \frac{1}{N} \sum_{i=1}^{N} \eta_i.$$

Recall that by the Central Limit Theorem,

$$\frac{1}{N} \sum_{i=1}^{N} \eta_i \sim \mathcal{N}\left(E[\eta], \frac{\sigma^2}{N}\right).$$

Most programming languages have tools for generating a uniformly distributed random variable ξ on the interval $[0, 1]$. Below we will discuss three approaches for sampling other kinds of RVs.

3.1. Inversion.

Suppose we need to sample a random variable η with a pdf $f(x)$. Assume that we can integrate $f(x)$ analytically, i.e., have an analytic expression for the cumulative distribution function (CDF) $F(x)$. Assume that $f(x) \equiv 0$ for $x < 0$. We observe that

$$\int_0^x f_\eta(y) dy = F_\eta(x) = \xi. \quad \text{Hence} \quad x = F_\eta^{-1}(\xi),$$

where $F_\eta^{-1}(\xi)$ is the inverse function of F_η. It exists if $F_\eta(x)$ is strictly increasing. If $\xi \sim \mathcal{U}(0, 1)$ is uniformly distributed on $[0, 1]$ then $F_\eta^{-1}(\xi)$ has CDF $F_\eta(x)$. Indeed, the CDF of $\xi \sim \mathcal{U}(0, 1)$ is

$$F_\xi(x) = \mathbb{P}(\xi \le x) = \begin{cases} 1, & x \ge 1 \\ x, & 0 \le x < 1 \\ 0, & x < 0 \end{cases}.$$

By definition of the CDF,

$$F_\eta(x) = \mathbb{P}(\eta \le x).$$

On the other hand, if we replace η with $F_\eta^{-1}(\xi)$ where $\xi \sim U(0, 1)$, we get

$$\mathbb{P}(\eta \le x) = \mathbb{P}(F_\eta^{-1}(\xi) \le x) = \mathbb{P}(\xi \le F_\eta(x)) = F_\xi(F_\eta(x)) = F_\eta(x).$$

Here we used the fact that $F_\xi(x) = x$ for $0 \le x \le 1$.

Example 1 Suppose we need to generate an exponentially distributed random variable η with pdf

$$f_\eta(x) = \begin{cases} ae^{-ax}, & x \ge 0 \\ 0, & x < 0, \end{cases} \quad \text{where } a > 0 \text{ is a constant.} \tag{1}$$

The CDF of η is given by

$$F_\eta(x) = \mathbb{P}(\eta \le x) = \int_0^x ae^{-ay}dy = 1 - e^{-ax}.$$

Let $\xi \sim \mathcal{U}(0,1)$. Then η can be generated from ξ by

$$\eta = F_\eta^{-1}(\xi) = -\tfrac{1}{a}\log(1-\xi).$$

Observing that $1 - \xi$ is also a random variable uniformly distributed on $[0,1]$, we can choose to generate η by

$$\eta = F_\eta^{-1}(1-\xi) = -\tfrac{1}{a}\log(\xi).$$

The limitation of the inversion method is due to the fact that not all pdfs are analytically integrable to CDFs.

3.2. **The change of variables. The Box-Muller algorithm.** The Box-Muller algorithm for generating Gaussian random variables exemplifies the change of variables approach.

Suppose we need to generate a Gaussian random variable (RV) η with mean 0 and variance σ^2 (i.e., $\eta \sim \mathcal{N}(0,\sigma^2)$) while we have a built-in function for generating a random variable ξ uniformly distributed on $[0,1]$. Unfortunately, the pdf of η

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{2}$$

is not analytically integrable. Therefore, we cannot use the inversion method directly. However, we can generate pairs of independent jointly Gaussian RVs (η_1, η_2) given a pair of independent RVs (ξ_1, ξ_2) uniformly distributed on $[0,1]$.

Let us elaborate this point. The joint pdf of two i.i.d. Gaussian RVs $\eta_1, \eta_2 \sim \mathcal{N}(0,\sigma^2)$ is

$$f_{\eta_1,\eta_2}(x,y) = f_{\eta_1}(x) f_{\eta_2}(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3}$$

Now let us to switch to the polar coordinates

$$r = \sqrt{x^2+y^2} \quad \text{and} \quad \theta = \begin{cases} \arctan\frac{y}{x}, & x \ge 0, \\ \pi + \arctan\frac{y}{x}, & x < 0. \end{cases}.$$

Respectively, $x = r\cos\theta$, $y = r\sin\theta$.

Let us recall the general formula for the variable change in a joint pdf. Suppose we have RVs

$$(X_1, \dots X_n) \in \Omega_X \quad \text{with the joint pdf} \quad f_{X_1,\dots,X_n}(x_1,\dots,x_n).$$

The desired RVs $(Y_1, \ldots, Y_n) \in \Omega_Y$ are functions of X_1, ..., X_n:

$$Y_i = g_i(X_1, \ldots, X_n), \quad i = 1, \ldots, n.$$

To be able to compute the joint pdf for the RVs Y_1, ..., Y_n, we make the following assumptions.

- The system of equations $y_i = g_i(x_1, \ldots, x_n)$, $i = 1, \ldots, n$ is uniquely solvable for x_1, ..., x_n for any given $(y_1, \ldots, y_n) \in \Omega_Y$, and the solution is given by $x_i = h_i(y_1, \ldots, y_n)$, $i = 1, \ldots, n$.
- The Jacobian

$$J(x_1, \ldots, x_1) = \det \left[\left(\frac{\partial g_i(x_1, \ldots, x_n)}{\partial x_j} \right)_{i,j=1}^{n} \right] \neq 0 \quad \text{for all} \quad (x_1, \ldots, x_n) \in \Omega_X.$$

Then the joint pdf for the RVs Y_1, ..., Y_n is given by

$$f_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n) = f_{X_1,\ldots,X_n}(x_1, \ldots, x_n)|J(x_1, \ldots, x_n)|^{-1}, \tag{4}$$

where $x_i = h_i(y_1, \ldots, y_n)$, $i = 1, \ldots, n$.

Using formula (4), we obtain the joint pdf for the random variables R and Θ, the polar radius and polar angle, from the joint pdf (3) for $\eta_1, \eta_2 \sim \mathcal{N}(0, \sigma^2)$:

$$f_{R,\Theta}(r, \theta) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{r^2}{2\sigma^2} \right\} r, \tag{5}$$

where the factor r come from the calculation

$$|J(x, y)|^{-1} = \left| \frac{\partial(r, \theta)}{\partial(x, y)} \right|^{-1} = \left| \frac{\partial(x, y)}{\partial(r, \theta)} \right| = r.$$

We observe that the pdf $f_{R,\Theta}(r, \theta)$ is independent of θ, hence the marginal pdf of θ is uniform on $[0, 2\pi]$. Therefore, if $\xi_2 \sim U(0, 1)$ then Θ can be set to $2\pi\xi_2$. The marginal pdf for R is found by integrating out Θ:

$$f_R(r) = \int_0^{2\pi} f_{R,\Theta}(r, \theta) d\theta = \frac{1}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} r.$$

We observe that the pdf $f_R(r)$ is analytically integrable:

$$F_R(a) = \mathbb{P}(r \leq a) = \frac{1}{\sigma^2} \int_{\sqrt{r^2} \leq a} e^{-\frac{r^2}{2\sigma^2}} r dr = \frac{1}{\sigma^2} \sigma^2 \int_0^{a^2/(2\sigma^2)} e^{-t} dt = 1 - e^{-a^2/(2\sigma^2)}.$$

Then we set:

$$F_R(a) = 1 - e^{-a^2/(2\sigma^2)} = 1 - \xi_1,$$

where $\xi_1 \sim U(0, 1)$, and use the fact that if ξ_1 is uniformly distributed on $[0, 1]$, then so is $1 - \xi_1$. Inverting F_R, we find:

$$a = \sqrt{-2\sigma^2 \log \xi_1}. \tag{6}$$

In summary, to obtain $\eta_1, \eta_2 \sim \mathcal{N}(0, \sigma^2)$, i.i.d., we generate $\xi_1, \xi_2 \sim U(0,1)$, i.i.d. and set

$$\begin{cases} \eta_1 = a\cos\theta = \sqrt{-2\sigma^2 \log \xi_1}\cos(2\pi\xi_2) \\ \eta_2 = a\sin\theta = \sqrt{-2\sigma^2 \log \xi_1}\sin(2\pi\xi_2) \end{cases} \tag{7}$$

Equation (7) is the Box-Muller formula for generating pairs of independent jointly Gaussian random variables with mean 0 and variance σ^2.

The method of variable change can be used in some other situations provided that we can find a variable change such that the new variables have analytically integrable marginals and hence can be obtained by the inversion method. The method of variable change is not universal since there is no general recipe for finding such a variable change. However, it does work beautifully for generating i.i.d. Gaussian RVs.

3.3. **Acceptance-rejection method.** Reference: note by Prof. K. Sigman (Columbia University):
http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ARM.pdf.

Contrary to the inversion method and the method of variable change, the acceptance-rejection method is universal in the following sense. Suppose we *want* to sample an RV with pdf $f(x)$. Suppose we *can* find an RV with pdf $g(x)$ that we can sample and the pdf $g(x)$ is such that

$$\sup_{x\in\mathbb{R}} \frac{f(x)}{g(x)} =: c < \infty. \tag{8}$$

Then the rejection method will work. However, if $c \gg 1$, the rejection method will be wasteful.

Algorithm 1: The Acceptance-Rejection algorithm

Initialization:
Set `accept = False`
The main body:
while `accept == False` **do**
  1: Generate a sample ξ of a random variable with pdf $g(x)$
  2: Generate a sample of $u \sim \mathcal{U}(0,1)$
  3: **if**

$$u \le \frac{f(\xi)}{cg(\xi)}$$

  then
    4: Set $\eta = \xi$
    5: Set `accept = True`

Algorithm 1 generates a single sample of a random variable η with pdf $f(x)$. If you need n samples of η, put the while-loop inside a for-loop.

The number of calls of step 1 in Algorithm 1 is a geometric random variable with the probability of success

$$p = \mathbb{P}\left(u \leq \frac{f(\xi)}{cg(\xi)}\right).$$

Therefore, the probability that the number of calls of step 1 is n is

$$\mathbb{P}(N = n) = (1-p)^{n-1}p.$$

The expected number of calls of step 1 required to generate a sample of η is

$$\mathbb{E}[N] = \sum_{n=1}^{\infty} n(1-p)^{n-1}p = p\sum_{n=1}^{\infty} n(1-p)^{n-1} = \frac{1}{p}.$$

This sum was calculated using the observation that

$$\sum_{n=1}^{\infty} n(1-p)^{n-1} = -\frac{d}{dp}\sum_{n=0}^{\infty}(1-p)^n = -\frac{d}{dp}\left(\frac{1}{1-(1-p)}\right) = -\frac{d}{dp}\left(\frac{1}{p}\right) = \frac{1}{p^2}.$$

Proposition 1. *The expected number of calls of step 1 in Algorithm 1 is c.*

Proof. Using the law of total probability we get

$$\begin{aligned} p = \mathbb{P}\left(u \leq \frac{f(\xi)}{cg(\xi)}\right) &= \int_{-\infty}^{\infty} \mathbb{P}\left(u \leq \frac{f(\xi)}{cg(\xi)} \,\middle|\, \xi = y\right) g(y)dy \\ &= \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y)dy = \frac{1}{c}\int_{-\infty}^{\infty} f(y)dy = \frac{1}{c}. \end{aligned}$$

□

The resulting random variable η has a conditional distribution of ξ conditioned on the event

$$E := \left\{u \leq \frac{f(\xi)}{cg(\xi)}\right\}.$$

Now we will prove that the pdf of the random variable η generated by Algorithm 1 is $f(y)$.

Proposition 2. *The pdf of the random variable η generated by Algorithm 1 is $f(y)$.*

Proof. We will prove that the cumulative distribution function (CDF) $F(y)$ of η is equal to

$$F(y) = \int_{-\infty}^{y} f(t)dt.$$

By construction,

$$F_\eta(y) = \mathbb{P}(\eta \leq y) = \mathbb{P}\left(\xi \leq y \,\middle|\, u \leq \frac{f(\xi)}{cg(\xi)}\right).$$

Let A be the event that $\{\xi \leq y\}$. Let B be the event that $u \leq \frac{f(\xi)}{cg(\xi)}$. We will use Bayes' theorem saying that

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}.$$

Therefore,

$$F_\eta(y) = \mathbb{P}\left(\xi \le y \,\middle|\, u \le \frac{f(\xi)}{cg(\xi)}\right) = \frac{\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)} \,\middle|\, \xi \le y\right)\mathbb{P}(\xi \le y)}{\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)}\right)}.$$

To compute $F(y)$, we will take into account the following facts. First, $\mathbb{P}(\xi \le y)$ is he CDF of ξ: $G(y) = \mathbb{P}(\xi \le y)$. Second, $\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)}\right) = \frac{1}{c}$ by Proposition 1. Hence, we find:

$$\begin{aligned} F_\eta(y) &= \mathbb{P}\left(\xi \le y \,\middle|\, u \le \frac{f(\xi)}{cg(\xi)}\right) = \frac{\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)} \,\middle|\, \xi \le y\right)\mathbb{P}(\xi \le y)}{\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)}\right)} \\ &= \frac{\mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)} \;\&\; \xi \le y\right)}{\frac{1}{c}} = c\int_{-\infty}^{y} \mathbb{P}\left(u \le \frac{f(\xi)}{cg(\xi)} \,\middle|\, \xi = t\right) g(t)dt \\ &= c\int_{-\infty}^{y} \frac{f(t)}{cg(t)} g(t)dt = \int_{-\infty}^{y} f(t)dt, \end{aligned}$$

as desired. □

Example 2 The Acceptance-Rejection method can be used to generate the standard Gaussian random variable from the exponential random variable. Let $\eta \sim \mathcal{N}(0,1)$ be the Gaussian RV with mean 0 and variance 1. The pdf of η is

$$f_\eta(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}. \tag{9}$$

Let ξ be the exponential RV with pdf $g(y) = e^{-y}$. We can readily obtain samples of ξ using the inversion method. Let $u \sim \mathcal{U}(0,1)$. then $\xi = -\log(u)$.

First, we note that the Gaussian RV can have either sign while the exponential RV is nonnegative. Hence, for negative y, the ratio $f_\eta(y)/g(y)$ is infinite. To overcome this difficulty, we will generate the RV $|\eta|$ whose pdf is

$$f_{|\eta|}(y) = \sqrt{\frac{2}{\pi}} e^{-\frac{y^2}{2}}. \tag{10}$$

Since η is symmetric w.r.t $y = 0$, we can generate a Bernoulli RV s that takes values ± 1 with probability $p = 1/2$. This can be done by generating $v \sim \mathcal{U}(0,1)$ and setting $s = -1$ if $v \le 1/2$ and $s = 1$ otherwise.

The ratio of pdfs of $|\eta|$ and ξ is

$$r(y) = \frac{f_{|\eta|}(y)}{g(y)} = \sqrt{\frac{2}{\pi}} e^{-\frac{y^2}{2}+y}, \quad y \ge 0.$$

Taking its derivative and setting it to zero we find that the maximum of $r(y)$ is achieved at $y = 1$ and is equal to

$$c = \sqrt{\frac{2}{\pi}} e^{\frac{1}{2}}.$$

Hence, the ratio that is compared with a sample of $u \sim \mathcal{U}(0,1)$ at step 2 of Algorithm 1 is

$$\frac{f_{|\eta|}(y)}{cg(y)} = e^{-\frac{y^2}{2}+y-\frac{1}{2}} = e^{-\frac{1}{2}(y-1)^2}.$$

Finally, we note that the inequality

$$u \le e^{-\frac{1}{2}(y-1)^2} \quad \text{is equivalent to} \quad \log u \le -\frac{1}{2}(y-1)^2.$$

Below is a vectorized MATLAB code implementing the algorithm for sampling $\mathcal{N}(0,1)$ using exponential RV with pdf $f(x) = e^{-x}$, $x \ge 0$, and $f(x) = 0$ otherwise. Figure 1 is generated by this code.

```
function AcceptRejectMethod()
%% Generate eta ~ N(0,1) using exp(-xi)
% see http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ARM.pdf
N = 1e8; % the number of samples
v = rand(N,1);
xi = -log(v);
u = rand(N,1);
log_u = log(u);
% generate signs for eta
b = rand(N,1);
ind = b < 0.5;
s = ones(N,1);
s(ind) = -1;
%
log_ratio = -0.5*(xi - 1).^2;
ind = find(log_u <= log_ratio);
Na = length(ind); % the number of accepted RVs
eta = xi(ind).*s(ind);
fprintf('N/Na = %d, C = sqrt(2*e/pi) = %d\n',N/Na,sqrt(2*exp(1)/pi));
%
%% plot a histogram to test the distribution
nbins = 500; % the number of bins
etamax = max(eta);
etamin = min(eta);
nb1 = nbins + 1;
x = linspace(etamin,etamax,nb1);
h = x(2) - x(1); % bin width
```

```
% xc = centers of bins
xc = linspace(etamin + 0.5*h,etamax - 0.5*h,nbins);
hh = zeros(nbins,1); % heights of the bins
for i = 1 : nbins
    ind = find(eta >= x(i) & eta < x(i + 1));
    hh(i) = length(ind);
end
hh = hh/(Na*h); % scale the histogram
f = exp(-0.5*x.^2)/sqrt(2*pi);
figure;
plot(x,f,'r','Linewidth',2);
hold on;
plot(xc,hh,'b','Linewidth',2);
grid;
set(gca,'Fontsize',20);
xlabel('x','Fontsize',20);
ylabel('f(x)','Fontsize',20);
legend('True N(0,1)','Generated N(0,1)');
end
```

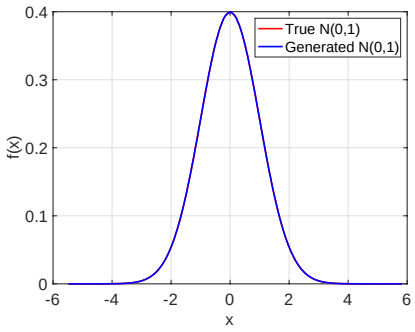

FIGURE 1. Figure generated by the code in Section 3.3.

4. MONTE-CARLO INTEGRATION

Monte-Carlo integration means the use of sampling for evaluating integrals. It can be used beyond the context of finding expected values of random variables.

Suppose we need to calculate an integral of the form

$$I = \int_a^b g(x)f(x)dx, \quad \text{where}$$

$$f(x) \geq 0, \ x \in [a,b], \quad \text{and} \quad \int_a^b f(x)dx = 1.$$

Such integral can be interpreted as the expected value of the function g of the RV η with the pdf $f(x)$, i.e.,

$$I = \int_a^b g(x)f(x)dx = \int_a^b g(x)f(x)dx = \mathbb{E}_f[g(\eta)].$$

Suppose we can sample i.i.d. RVs η_i each of which has the pdf $f(x)$ and a finite variance. According to the strong law of large numbers,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} g(\eta_i) = \mathbb{E}_f[g(\eta)] \quad a.s.$$

The integral I is called the estimand, the random variable $g(\eta)$ is called the estimator, and the quantity

$$\frac{1}{n}\sum_{i=1}^{n} g(\eta_i) \tag{11}$$

is the estimate. This method of evaluating integrals is called the *Monte Carlo integration*. According to the central limit theorem,

$$\frac{1}{n}\sum_{i=1}^{n} g(\eta_i) \;\longrightarrow\; \mathcal{N}\left(\mathbb{E}_f[g(\eta)], \frac{\mathsf{Var}(g(\eta))}{n}\right) \quad \text{in distribution.}$$

Therefore, the error of the estimate (11) is of the order of

$$\mathsf{err} \sim \frac{\sqrt{\mathsf{Var}(g(\eta))}}{\sqrt{n}}. \tag{12}$$

Eq. (12) suggests two ways to reduce the error of the Monte-Carlo integration: (i) to increase the number of samples n, and (ii) to reduce the variance in the numerator. Increasing the number of samples is simple but costly, as the error decays as $n^{-1/2}$. A better idea is to reduce the variance of $g(\eta)$. One approach to the variance reduction is called the *importance sampling*.

4.1. **Importance sampling.** Suppose we need to calculate the integral

$$I = \int_a^b g(x)dx.$$

In order to make the $\mathsf{Var}(g(\eta))$ small we need to find a function $h(x)$ with the following properties:

(1) The integral

$$I_1 = \int_a^b h(x)dx$$

is easy to evaluate;

(2) $h(x) \geq 0$;

(3) We can sample a random variable with the pdf

$$\frac{h(x)}{I_1} \quad \text{easily;}$$

(4) $g(x)/h(x)$ varies little.

Then we have

$$I = \int_a^b g(x)dx = I_1 \int_a^b \frac{g(x)}{h(x)} \frac{h(x)}{I_1} dx$$
$$= I_1 \mathbb{E}_h \left[\frac{g}{h}(\eta) \right] \approx \frac{I_1}{n} \sum_{i=1}^{n} \frac{g(\eta_i)}{h(\eta_i)},$$

where η has the pdf $h(x)/I_1$. See the example with

$$I = \int_0^1 \cos(x/5) e^{-5x} dx$$

in [2].

4.2. **Monte Carlo integration in higher dimensions.** Suppose we would like to evaluate the integral

$$I = \int_\Omega g(x)dx \tag{13}$$

where $\Omega \subset \mathbb{R}^n$. We proceed as we did in 1D. Let us generate a random variable η whose pdf $f_\eta(x)$ is nonzero in Ω and zero elsewhere and rewrite Eq. (13) as

$$I = \int_\Omega \frac{g(x)}{f_\eta(x)} f_\eta(x)dx \tag{14}$$

By the strong law of large numbers,

$$I = \int_\Omega \frac{g(x)}{f_\eta(x)} f_\eta(x)dx = \mathbb{E}_\eta \left[\frac{g(x)}{f_\eta(x)} \right] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{g(\eta_i)}{f_\eta(\eta_i)}, \tag{15}$$

where η_i, $1 \le i \le N$, are samples of the random variable η with pdf $f_\eta(x)$.

Suppose η is uniformly distributed in Ω. Then its pdf is given by

$$f_\eta(x) = \begin{cases} \frac{1}{|\Omega|}, & x \in \Omega \\ 0, & x \notin \Omega, \end{cases} \tag{16}$$

where $|\Omega|$ is the volume of Ω. In this case, Eq. (17) becomes:

$$I = \int_\Omega \frac{g(x)}{f_\eta(x)} f_\eta(x)dx \approx \frac{|\Omega|}{N} \sum_{i=1}^{N} g(\eta_i). \tag{17}$$

Similarly we proceed when we need to calculate an integral over a k-dimensional hypersurface S embedded into $\mathbb{R}^n$:

$$I = \int_S g(x) d\sigma, \tag{18}$$

where $d\sigma$ is a surface element. Let η be a random variable whose pdf is supported at the hypersurface S, i.e. $f_\eta(x) > 0$ if and only if $x \in S$. Then the integral is approximated by

$$I = \int_S g(x) d\sigma \approx \frac{1}{N} \sum_{i=1}^{N} \frac{g(x_i)}{f_\eta(x_i)}, \tag{19}$$

where x_i, $1 \le i \le N$ are samples of the random variable η. If η is uniformly distributed on the hypersurface S, then

$$I = \int_S g(x) d\sigma \approx \frac{|S|}{N} \sum_{i=1}^{N} g(x_i), \tag{20}$$

where $|S|$ is the measure (k-dimensional area) of S:

$$|S| = \int_S d\sigma. \tag{21}$$

Example 17 Consider the integral

$$I = \int_{S_{n-1}} g(x) d\sigma, \tag{22}$$

where S_{n-1} is the unit $n-1$-dimensional sphere (n-sphere) embedded into $\mathbb{R}^n$:

$$S_{n-1} = \{x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_1^2 + \ldots + x_n^2 = 1\}.$$

Let us generate N samples of random variable η uniformly distributed on S_n. This can be done as follows. First we generate an array $N \times n$ of independent Gaussian random variables with mean 0 and variance 1. It is well-known that n independent Gaussian random variables with mean zero and variance 1 have the joint pdf

$$f_{\eta_1,\ldots,\eta_n}(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{x_1^2+\ldots+x_n^2}{2}} \equiv \frac{1}{(2\pi)^{n/2}} e^{-\frac{r^2}{2}}, \tag{23}$$

where $r := \sqrt{x_1^2 + \ldots + x_n^2}$. Let us treat each row of our array as a sample of a vector random variable ξ with pdf given by Eq. (23). The distribution of ξ is spherically symmetric. Hence, we can obtain the desired random variable η uniformly distributed on the unit sphere by normalizing the radius of ξ:

$$\eta = \frac{\xi}{\sqrt{\xi_1^2 + \ldots + \xi_n^2}}. \tag{24}$$

In matlab, N samples of a random variable η uniformly distributed on the unit n-sphere can be generated by the following set of commands:

```
xi = randn(N, n);
aux = sqrt(sum(xi.^2, 2))*ones(1, n);
eta = xi./aux;
```

The surface area of the unit sphere S_{n-1} is given by

$$|S_{n-1}| = \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})}, \tag{25}$$

where

$$\Gamma(x) := \int_0^\infty t^{x-1}e^{-t}dt$$

is the Gamma-function. Thus, the integral (22) can be estimated as

```
S = 2*pi^(n/2)/gamma(n/2);
I = sum(g(eta))*S/N;
```

where n, N, and the function $y = g(x)$ must be provided. A table of exact integrals of some functions over unit hypersphere are found here.

- For $n = 4$ and $g(x) = x_1^2x_2^2$, the exact integral (22) is
$$I = \int_{S_3} x_1^2x_2^2 d\sigma = \frac{\pi^2}{12} = 0.8224670\ldots,$$
while its estimate using 10^6 samples is 0.8227420, and its error estimate is 10^{-3}.
- For $n = 10$ and $g(x) = x_1^2$, the exact integral (22) is
$$I = \int_{S_9} x_1^2 d\sigma = \frac{\pi^5}{120} = 2.550164\ldots,$$
while its estimate using 10^6 samples is 2.548990, and its error estimate is $3 \cdot 10^{-3}$.

5. Motivating example. The Ising model

References:

- A. Chorin and O. Hald, "Stochastic Tools in Mathematics and Science", Second Edition, Springer, 2009.
- Harvey Gould and Jan Tobochnik, "Magnetic systems". Lecture notes.

We will consider the Ising model in 2D. It is straightforward to promote it in 3D. For simplicity, we consider a 2D $N \times N$ mesh with periodic boundary conditions, i.e., each site (i, j), $0 \le i, j \le N-1$, has four nearest neighbors:

$$(i-1 \mod N, j),\ (i+1 \mod N, j),\ (i, j-1 \mod N),\ (i, j+1 \mod N).$$

The spin at each site (i, j) can be up, $s_{i,j} = +1$, or down, $s_{i,j} = -1$. The total energy of the spin system is the sum of energies of the magnetic interactions of the spins $s_{i,j}$ at the nearest neighbors of the mesh:

$$H(\{s_{i,j}\}) = -\sum_{i,j=0}^{N-1} s_{i,j}\left[s_{i+1,j} + s_{i,j+1}\right], \tag{26}$$

where the addition in indices is modulo N.

The probability of each of the 2^{N^2} spin microstates is

$$\mathbb{P}(\{s_{i,j}\}) = \frac{1}{Z} e^{-\beta H(\{s_{i,j}\})}, \tag{27}$$

where β^{-1} is the temperature times Boltzmann's constant and Z is the normalization constant:

$$Z = \sum_{\{s_{i,j}\}} e^{-\beta H(\{s_{i,j}\})}.$$

A physical question of interest is finding the mean magnetization at the given β. The magnetization of a microstate is the sum of the spins divided by the number of sites:

$$m(\{s_{i,j}\}) = \frac{1}{N^2} \sum_{i,j=0}^{N} s_{i,j}. \tag{28}$$

The mean magnetization is

$$\mu(\beta) = \mathbb{E}\left[m(\{s_{i,j}\})\right] = \sum_{\{s_{i,j}\}} m(\{s_{i,j}\}) \mathbb{P}(\{s_{i,j}\}), \tag{29}$$

where m and $\mathbb{P}$ are given by equations (28) and (27) respectively.

The 2D Ising model is interesting because it exhibits a phase transition. If the temperature β^{-1} is low, almost all spins are aligned up or down. Hence, the mean magnetization μ is close to 1 or to -1. If $\beta^{-1} > \beta_c^{-1} = 2.269$, ($\beta_c = 0.4407$) the spins relatively easily flip up and down and the resulting mean magnetization is zero. The exact formula for the mean magnetization is due to C. N. Yang (1952):

$$\mu(\beta) = \begin{cases} \pm \left(1 - [\sinh(2\beta)]^{-4}\right)^{1/8}, & \beta > \beta_c \\ 0, & \beta < \beta_c \end{cases}. \tag{30}$$

The direct calculation of the mean magnetization by formula (29) is infeasible because the number of terms in the sum is extremely large, 2^{N^2}. Moreover, the normalization constant Z is unknown.

However, despite the number of terms the sum in (29) is huge, the overwhelming majority of them is very small and do not contribute much to the sum. This suggests that we should use some kind of importance sampling, i.e., an algorithm that samples the large terms in the sum in (29). The *Metropolis algorithm* described below is such an algorithm. It is an example of a Markov Chain Monte Carlo (MCMC).

6. Markov Chains

Reference: J. R. Norris "Markov Chains", Cambridge, UK ; New York : Cambridge University Press, 1998.

7. Discrete-time Markov chains

Think about the following problem.

> **Example 3 (Gambler's ruin).** Imagine a gambler who has \$1 initially. At each discrete moment of time $t = 0, 1, \ldots$, the gambler can play \$1 if he has it and win one more \$1 with probability p or lose it with probability $q = 1 - p$. If the gambler runs out of money, he is ruined and cannot play anymore. What is the probability that the gambler will be ruined?

The gambling process described in this problem exemplifies a discrete-time Markov chain. In general, a discrete-time Markov chain is defined as a sequence of random variables $(X_n)_{n\geq 0}$ taking a finite or countable set of values and characterized by the Markov property: the probability distribution of X_{n+1} depends only of the probability distribution of X_n and does not depend on X_k for all $k \leq n-1$. We will denote this discrete set of values by S and call it *the set of states*.

Definition 1. *We say that a sequence of random variables $(X_n)_{n\geq 0}$, where*

$$X_n : \Omega \to S \subset \mathbb{Z},$$

is a Markov chain with initial distribution λ and transition matrix $P = (p_{ij})_{i,j\in S}$ if

(1) *X_0 has distribution $\lambda = \{\lambda_i \mid i \in S\}$ and*
(2) *the Markov property holds:*

$$\mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n, \ldots, X_0 = i_0) = \mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n) = p_{i_n i_{n+1}}.$$

We will denote the Markov chain by $\mathsf{Markov}(P, \lambda)$. Note that the ith row of $P = (p_{ij})$ is the probability distribution for X_{n+1} conditioned on the fact that $X_n = i$. Therefore, all entries of the matrix P are nonnegative, and the row sums are equal to one:

$$p_{ij} \geq 0, \quad \sum_{j\in S} \mathbb{P}(X_{n+1} = j \mid X_n = i) = \sum_{j\in S} p_{ij} = 1.$$

A matrix P satisfying these conditions is called *stochastic* or *Markov*.

We will address the following questions about Markov chains:

- What is the equilibrium probability distribution, i.e., the one that is preserved from step to step?
- Does the probability distribution of X_n tend to the equilibrium distribution?

Prior to addressing these question, we will go over some basic concepts.

7.1. Time evolution of the probability distribution.

If the set of states S is finite, i.e., if $|S| = N$, then P^n is merely the nth power of P. If S is infinite, we define P^n by

$$(P^n)_{ij} \equiv p_{ij}^{(n)} = \sum_{i_1\in S} \cdots \sum_{i_{n-1}\in S} p_{ii_1} p_{i_1 i_2} \ldots p_{i_{n-1} j}.$$

Notation. $\mathbb{P}_i(X_n = j)$ denotes the probability that the Markov process starting at i at time 0 will reach state j at time n:

$$\mathbb{P}_i(X_n = j) := \mathbb{P}(X_n = j \mid X_0 = i).$$

Theorem 1. *Let $(X_n)_{n\geq 0}$ be a Markov chain with initial distribution λ and transition matrix P. Then for all $n, m \geq 0$*

(1) $\mathbb{P}(X_n = j) = (\lambda P^n)_j$;
(2) $\mathbb{P}_i(X_n = j) = \mathbb{P}(X_{n+m} = j \mid X_m = i) = p_{ij}^{(n)}$.

Proof. (1)

$$\begin{aligned}\mathbb{P}(X_n = j) &= \sum_{i_0\in S} \dots \sum_{i_{n-1}\in S} \mathbb{P}(X_n = j, X_{n-1} = i_{n-1}, \dots, X_0 = i_0)\\ &= \sum_{i_0\in S} \dots \sum_{i_{n-1}\in S} \mathbb{P}(X_n = j \mid X_{n-1} = i_{n-1}, \dots, X_0 = i_0)\mathbb{P}(X_{n-1} = i_{n-1}, \dots, X_0 = i_0)\\ &= \sum_{i_0\in S} \dots \sum_{i_{n-1}\in S} \mathbb{P}(X_n = j \mid X_{n-1} = i_{n-1})\mathbb{P}(X_{n-1} = i_{n-1} \mid X_{n-2} = i_{n-1}) \dots \mathbb{P}(X_0 = i_0)\\ &= \sum_{i_0\in S} \dots \sum_{i_{n-1}\in S} \lambda_{i_0} p_{i_0 i_1} \dots p_{i_{n-1} j} = (\lambda P^n)_j.\end{aligned}$$

(2) The second statement is proven similarly.

□

7.2. **Communicating classes and irreducibility.** We say that state i *leads to* state j (denote it by $i \longrightarrow j$) if

$$\mathbb{P}_i(X_n = j \text{ for some } n \geq 0) > 0.$$

If i leads to j and j leads to i we say that i and j *communicate* and write $i \longleftrightarrow j$. Note that i leads to j if and only if one can find a finite sequence $i_1, \dots, i_{n-1}$ such that

$$p_{ii_1} > 0,\ p_{i_1 i_2} > 0,\ \dots,\ p_{i_{n-1} j} > 0.$$

This, in turn, is equivalent to the condition that $p_{ij}^{(n)} > 0$ for some n.

The relation $\longleftrightarrow$ is an equivalence relation as it is

(1) symmetric as if $i \longleftrightarrow j$ then $j \longleftrightarrow i$;
(2) reflective, i.e., $i \longleftrightarrow i$;
(3) transitive, as $i \longleftrightarrow j$ and $j \longleftrightarrow k$ imply $i \longleftrightarrow k$.

Therefore, the set of states is divided into equivalence classes with respect to the relation $\longleftrightarrow$ called *communicating classes.*

Definition 2. *We say that a communicating class C is closed if*

$$i \in C, i \longrightarrow j \text{ imply } j \in C.$$

Once the chain jumps into a closed class, it stays there forever.

A state i is called *absorbing* if $\{i\}$ is a closed class. In the corresponding network, the vertex i has either only incoming edges, or no incident edges at all.

Example 4 Let us identify the states in the Gambler's ruin Markov chain 3 with the number of dollars at each of them. It is easy to see that states $\{1, 2, \ldots\} =: C_1$ constitute a communication class. The class C_1 is not closed because state $1 \in C_1$ leads to state $0 \notin C_1$. State 0 is a closed communicating class $\{0\} =: C_0$ and an absorbing state.

Definition 3. *A Markov chain whose set of states S is a single communicating class is called irreducible.*

7.3. Invariant distributions and measures.

Definition 4. *A measure on a Markov chain is any vector $\lambda = \{\lambda_i \geq 0 \mid i \in S\}$. A measure is invariant (a. k. a stationary or equilibrium) if*

$$\lambda = \lambda P.$$

A measure is a distribution if, in addition, $\sum_{i \in S} \lambda_i = 1$.

Theorem 2. *Let the set of states S of a Markov chain $(X_n)_{n \geq 0}$ be finite. Suppose that for some $i \in S$*

$$\mathbb{P}_i(X_n = j) = p_{ij}^{(n)} \to \pi_j \text{ as } n \to \infty \text{ for all } j \in S.$$

Then $\pi = \{\pi_j \mid j \in S\}$ is an invariant distribution.

Proof. Since $p_{ij}^{(n)} \geq 0$ we have $\pi_j \geq 0$. Show that $\sum_{j \in S} \pi_j = 1$. Since S is finite, we can swap the order of taking limit and summation:

$$\sum_{j \in S} \pi_j = \sum_{i \in S} \lim_{n \to \infty} p_{ij}^{(n)} = \lim_{n \to \infty} \sum_{i \in S} p_{ij}^{(n)} = 1.$$

Show that $\pi = \pi P$:

$$\pi_j = \lim_{n \to \infty} p_{ij}^{(n)} = \lim_{n \to \infty} \sum_{k \in S} p_{ik}^{(n-1)} p_{kj} = \sum_{k \in S} \lim_{n \to \infty} p_{ik}^{(n-1)} p_{kj} = \sum_{k \in S} \pi_k p_{kj}.$$

□

Remark If the set of states is not finite, then the one cannot exchange summation and taking limit. For example, $\lim_{n \to \infty} p_{ij}^{(n)} = 0$ for all i, j for a simple symmetric random walk on $\mathbb{Z}$. $\{\pi_i = 0 \mid i \in \mathbb{Z}\}$ is certainly an invariant measure, but it is not a distribution.

The existence of an invariant distribution does not guarantee convergence to it. For example, consider the two-state Markov chain with transition matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The distribution $\pi = (1/2, 1/2)$ is invariant as

$$(1/2, 1/2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = (1/2, 1/2).$$

However, for any initial distribution $\lambda = (q, 1-q)$ where $q \in [0, 1/2) \cup (1/2, 1]$, the limit

$$\lim_{n\to\infty} P^n$$

does not exist as

$$P^{2k} = I, \quad P^{2k+1} = P.$$

In order to eliminate such cases, we introduce the concept of aperiodic states.

Definition 5. *Let us call a state i aperiodic, if $p_{ii}^{(n)} > 0$ for all sufficiently large n.*

Theorem 3. *Suppose P is irreducible and has an aperiodic state i. Then for all states j and k, $p_{jk}^{(n)} > 0$ for all sufficiently large n. In particular, all states are aperiodic.*

Proof. Since the chain is irreducible, there exist such r and s that $p_{ji}^{(r)} > 0$ and $p_{ik}^{(s)} > 0$. Then for sufficiently large n we have

$$p_{jk}^{(r+n+s)} = \sum_{i_1,\ldots,i_n \in S} p_{ji_1}^{(r)} p_{i_1 i_2} \cdots p_{i_{n-1} i_n} p_{i_n k}^{(s)} \geq p_{ji}^{(r)} p_{ii}^{(n)} p_{ik}^{(s)} > 0.$$

□

Definition 6. *We will call a Markov chain aperiodic if all its states are aperiodic.*

Theorem 4. *Suppose that $(X_n)_{n\geq 0}$ is a Markov chain with transition matrix P and initial distribution λ. Let P be irreducible and aperiodic, and suppose that P has an invariant distribution π. Then*

$$\mathbb{P}(X_n = j) \to \pi_j \text{ as } n \to \infty \text{ for all } j.$$

In particular,

$$p_{ij}^{(n)} \to \pi_j \text{ as } n \to \infty \text{ for all } i, j.$$

A proof of this theorem is found in [5]. In the case where the set of states is finite, this result can be proven using linear algebra. A building block of this proof is the Perron-Frobenius theorem.

Theorem 5. *Let A be an $N \times N$ matrix with nonnegative entries such that all entries of A^m are strictly positive for all $m > M$. Then*

1. *A has a positive eigenvalue $\lambda_0 > 0$ with corresponding left eigenvector x_0 where all entries are positive;*
2. *if $\lambda \neq \lambda_0$ is any other eigenvalue, then $|\lambda| < \lambda_0$.*
3. *λ_0 has geometric and algebraic multiplicity one.*

Let P be the stochastic matrix for a Markov chain with N states. For sufficiently large n, all entries of P^n for stochastic irreducible aperiodic matrices P become positive. The proof of this fact is similar to the one of Theorem 3. Furthermore, the largest eigenvalue of a stochastic matrix is equal to 1. Indeed, since the row sums of P are ones, $\lambda_0 = 1$ is an eigenvalue with the right eigenvector $e = [1, \ldots, 1]^\top$.

Now let us show that the other eigenvalues are less than $\lambda_0 = 1$ in absolute value. Let (λ, v) be an eigenvalue and a corresponding right eigenvector of a stochastic matrix P^n. We normalize v so that

$$v_i = \max_{k\in S} |v_k| = 1.$$

Since

$$\lambda = \lambda v_i = \sum_{k\in S} p_{ik}^{(n)} v_k,$$

we have

$$|\lambda| = \left|\sum_{k\in S} p_{ik}^{(n)} v_k\right| \le \sum_{k\in S} p_{ik}^{(n)} |v_k| \le \sum_{k\in S} p_{ik}^{(n)} = 1.$$

Since all $p_{ik}^{(n)} > 0$ and $\sum_{k\in S} p_{ik}^{(n)} = 1$, the equality holds if and only if (i) all v_k are of the same sign and (ii) $|v_k| = 1$, $k \in S$. Hence, if the eigenvector $v \neq [1, \dots, 1]^\top$ then the corresponding eigenvalue λ has absolute value $|\lambda| < 1$. Finally, we observe that if (μ, v) is an eigenpair of P than (μ^n, v) is an eigenpair of P^n. Therefore, an irreducible and aperiodic P has a unique eigenvalue $\lambda_0 = 1$, and all other eigenvalues have absolute values less than 1.

Remark The fact that the eigenvalues of a stochastic matrix do not exceed 1 in absolute value is an instance of the Gershgorin Circle Theorem.

Theorem 6. *Every irreducible aperiodic Markov chain with a finite number of states N has a unique invariant distribution π. Moreover,*

$$\lim_{n\to\infty} qP^n = \pi \tag{31}$$

for any initial distribution q.

Proof. The Perron-Frobenius theorem applied to a finite stochastic irreducible aperiodic matrix P implies that the largest eigenvalue of P is $\lambda_0 = 1$ and all other eigenvalues are strictly less than 1 in absolute value. The left eigenvector π, corresponding to λ_0 has positive entries and can be normalized so that they sum up to 1. Hence,

$$\pi = \pi P, \quad \sum_{i=1}^{N} \pi_i = 1.$$

Now let us establish convergence. First we consider the case when P is diagonalizable:

$$P = V\Lambda U,$$

where Λ is the matrix with ordered eigenvalues along its diagonal:

$$\Lambda = \begin{pmatrix} 1 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{N-1} \end{pmatrix}, \quad 1 > |\lambda_1| \ge \ldots \ge |\lambda_{N-1}|,$$

V is the matrix of right eigenvectors of P: $PV = V\Lambda$, such that its first column is $e = [1, \dots, 1]^\top$. $U = V^{-1}$ is the matrix of left eigenvectors of P: $UP = \Lambda U$. The first row of U is $\pi = [\pi_1, \dots, \pi_N]$. One can check that if $UV = I_N$, these choices of the first column of V and the first row of U are consistent. Therefore, taking into account that $\sum_{i=1}^N q_i = 1$, we calculate:

$$\begin{aligned}
&\lim_{n\to\infty} qP^n \\
&= \lim_{n\to\infty} [q_1\ q_2\ \dots\ q_N] \begin{pmatrix} 1 & * & * & * \\ 1 & * & * & * \\ & \dots & & \\ 1 & * & * & * \end{pmatrix} \begin{pmatrix} 1 & & & \\ & \lambda_1^n & & \\ & & \ddots & \\ & & & \lambda_{N-1}^n \end{pmatrix} \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_N \\ * & * & * & * \\ & \dots & & \\ * & * & * & * \end{pmatrix} \\
&= [1\ 0\ \dots\ 0] \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix} \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_N \\ * & * & * & * \\ & \dots & & \\ * & * & * & * \end{pmatrix} \\
&= [\pi_1\ \pi_2\ \dots\ \pi_N].
\end{aligned}$$

In the case when P is not diagonalizable, the argument is almost identical, just a bit more tedious. We consider the Jordan decomposition of P

$$P = VJU$$

where $U = V^{-1}$ and J is the Jordan form of of P, i.e., a block-diagonal matrix of the form:

$$J = \begin{bmatrix} 1 & & & \\ & J_1 & & \\ & & \ddots & \\ & & & J_r \end{bmatrix},$$

with the first block being 1×1 matrix $J_0 \equiv 1$, and respectively, the first column of V being $[1, \dots, 1]^\top$, and the first row of U being π – the right and left eigenvectors corresponding to the eigenvalue 1, and the other blocks J_i of sizes $m_i \times m_i$, where $1 \le m_i \le N-1$ and $m_1 + \dots + m_r = N - 1$, of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & 1 & \\ & & \ddots & \ddots \\ & & & \lambda_i \end{bmatrix} =: \lambda_i I_{m_i \times m_i} + E. \tag{32}$$

Exercise (1) Check that the matrix E in Eq. (32) with ones right above the diagonal and all other entries zero is nilpotent. More precisely, $E^{m_i} = \mathbf{0}_{m_i \times m_i}$.

(2) Check that the matrices $\lambda_i I_{m_i \times m_i}$ and E commute.

(3) Check that
$$J_i^n = \sum_{k=0}^{m_i-1} \left(\begin{array}{c} n \\ k \end{array} \right) \lambda_i^{n-k} E^k.$$
(4) Argue that
$$\lim_{n\to\infty} J_i^n = \mathbf{0}_{m_i\times m_i}$$
provided that $|\lambda_i| < 1$.
(5) Now prove Eq. (31) for the case when P is not diagonalizable.

□

8. Time reversal and detailed balance

For Markov chains, the past and the future are independent given the present. This property is symmetric in time and suggests looking at Markov chains with time running backward. On the other hand, convergence to equilibrium shows that the behavior is asymmetric in time. Hence, to complete the symmetry in time, we need to start with the equilibrium distribution.

Theorem 7. *Let $(X_n)_{0\le n\le N}$ be* Markov(P,π)*, where P is irreducible and π is invariant. Define $Y_n = X_{N-n}$. Then $(Y_n)_{0\le n\le N}$ is* Markov$(\hat{P},\pi)$ *where the transition matrix $\hat{P} = (\hat{p}_{ij})$ defined by*
$$\pi_j p_{ji} = \pi_i \hat{p}_{ij} \text{ for all } i,j \in S.$$

Proof. Note that, since P is irreducible, all components of π are positive. We need to check the following three facts.

(1) Check that $\hat{P}$ is a stochastic matrix (i.e., all its entries are nonnegative and its row sums are equal to 1):
$$\hat{p}_{ij} = \frac{\pi_j}{\pi_i} p_{ji} \ge 0.$$
$$\sum_{j\in S} \hat{p}_{ij} = \frac{1}{\pi_i} \sum_{j\in S} \pi_j p_{ji} = \frac{\pi_i}{\pi_i} = 1.$$
In the last equation, we used the fact that π is invariant for P.
(2) Check that π is invariant for $\hat{P}$, i.e., that $\pi\hat{P} = \pi$:
$$\sum_{j\in S} \pi_j \hat{p}_{ji} = \sum_{j\in S} \pi_i p_{ij} = \pi_i \sum_{j\in S} p_{ij} = \pi_i \text{ for all } i \in S.$$
(3) Check that $(Y_n)_{0\le n\le N}$ satisfies Markov property.
$$\mathbb{P}(Y_0 = i_0, Y_1 = i_1, \dots, Y_N = i_N) = \mathbb{P}(X_0 = i_N, X_1 = i_{N-1}, \dots, X_N = i_0)$$
$$= \pi_{i_N} p_{i_N i_{N-1}} \dots p_{i_1 i_0} = \hat{p}_{i_N i_{N-1}} \pi_{i_{N-1}} p_{i_{N-1} i_{N-2}} \dots p_{i_1 i_0}$$
$$= \dots = \hat{p}_{i_{N-1} i_N} \dots \hat{p}_{i_0 i_1} \pi_{i_0}.$$
Therefore, $(Y_n)_{0\le n\le N}$ satisfies Markov property.

□

Definition 7. *The chain $(Y_n)_{0\le n\le N}$ is called the time-reversal of $(X_n)_{0\le n\le N}$.*

Definition 8. *A stochastic matrix P and a measure λ are in detailed balance if*

$$\lambda_i p_{ij} = \lambda_j p_{ji}.$$

Suppose the set of states S is finite, the matrix P is irreducible, and the system is distributed according to the invariant distribution π. The condition of detailed balance means the following. Let $N_{i\to j}(n)$ be the number of transitions from i to j observed by time n. Then for all $i, j \in S$,

$$\lim_{n\to\infty} \frac{N_{i\to j}(n)}{N_{j\to i}(n)} = 1,$$

if P is in detailed balance with π. In words, over large intervals of time, on average, one observes equal numbers of transitions from i to j and from j to i for all $i, j \in S$ given the detailed balance.

The detailed balance condition gives us another way to check whether a given measure λ is invariant.

Theorem 8. *Let P and λ be in detailed balance. Then λ is invariant for P.*

Proof.

$$(\lambda P)_i = \sum_{j\in S} \lambda_j p_{ji} = \lambda_i \sum_{j\in S} p_{ij} = \lambda_i.$$

Hence $\lambda P = \lambda$. □

Definition 9. *Let $(X_n)_{n\ge 0}$ be* $\mathsf{Markov}(P, \lambda)$ *where P is irreducible. We say that $(X_n)_{n\ge 0}$ is reversible if for all $N \ge 1$, $(X_{N-n})_{0\le n\le N}$ is* $\mathsf{Markov}(P, \lambda)$.

Theorem 9. *Let P be an irreducible stochastic matrix and let λ be a distribution. Suppose that $(X_n)_{n\ge 0}$ is* $\mathsf{Markov}(P, \lambda)$. *Then the following are equivalent:*

(1) *$(X_n)_{n\ge 0}$ is reversible;*
(2) *P and λ are in detailed balance.*

Proof. Both (1) and (2) imply that λ is invariant for P. Then both (1) and (2) are equivalent to the statement that $\hat{P} = P$. □

9. Metropolis and Metropolis-Hastings algorithms

Metropolis and Metropolis-Hastings algorithms exemplify the family of Markov Chain Monte Carlo (MCMC) algorithms. In both, importance sampling is done via running a Markov chain designed to have the stochastic matrix in detailed balance with the desired invariant probability measure. Note that while an invariant measure f can be known, the invariant distribution $\pi = Z^{-1}f$ can still be hard to find because it is hard to find the normalization constant Z.

We consider the problem of finding the expected value of $g(\eta)$ in the case where the set of outcomes Ω is finite but huge, i.e., $|\Omega| = N$ where N is huge. Let $\pi(\omega)$ be the probability distribution on Ω, then

$$E[g(\eta)] = \sum_{\omega\in\Omega} g(\eta(\omega))\pi(\omega), \tag{33}$$

9.1. **Metropolis algorithm.** The Metropolis algorithm approximates the sum (33) by finding a subset $\Omega' \subset \Omega$ such that the probabilities of the optcomes $\omega \in \Omega'$ are high and summing only over those outcomes. Finding such a subset of outcomes and sampling from it is achieved by constructing a discrete-time Markov chain $(X_n)_{n\geq 0}$, $X_n : \Omega \to \{1, \dots, N\}$ so that its invariant distribution is $\pi(\omega)$. In the Metropolis algorithm, such a Markov chain is found with the help of the detailed balance. The stochastic matrix P for such a Markov chain is designed using the claim of Theorem 8 that if P and π are in detailed balance then π is invariant for P. In addition, P should be irreducible and aperiodic so that the convergence of the probability distribution to the invariant distribution π is achieved starting from any initial distribution.

Typically, we do not know π. Instead, we know an invariant measure f such that $\pi = Z^{-1}f$ where Z is an unknown normalization constant.

The stochastic matrix P is constructed in two steps that A. Chorin describes as "we first do something stupid and then improve it".

(1) Suppose at time n, $X_n = k$. Propose a move from state k according to some irreducible aperiodic transition matrix $Q = (q_{ij})_{ij\in S}$ made-up by you. In the original Metropolis algorithm, the matrix Q must be symmetric, i.e., $q_{ij} = q_{ji}$. Suppose the proposed move is from state k to state l.

(2) To guarantee that the condition $f_i p_{ij} = f_j p_{ji}$ holds, accept the proposed move with the probability

$$\alpha = \min\left\{\frac{f_l}{f_k}, 1\right\}. \tag{34}$$

I.e., if the proposed state l is more likely than the current state k, move to the new state. Otherwise, move there with probability f_l/f_k or stay at state k with probability $1 - f_l/f_k$.

As a result, the transition probabilities p_{ij} are given by

$$p_{ij} = q_{ij}\min\left\{\frac{f_j}{f_i}, 1\right\} \text{ if } i \neq j, \quad p_{ii} = 1 - \sum_{j\neq i} q_{ij}\min\left\{\frac{f_j}{f_i}, 1\right\}. \tag{35}$$

Let us check that P is in detailed balance with f. Assume $i \neq j$. Let $f_j/f_i \leq 1$. Then

$$f_i p_{ij} = f_i q_{ij}\frac{f_j}{f_i} = f_j q_{ij} = f_j q_{ji} = f_i p_{ji}.$$

If $f_j/f_i > 1$ then

$$f_i p_{ij} = f_i q_{ij} = f_i q_{ji} = f_i p_{ji}\frac{f_j}{f_i} = f_j p_{ij}.$$

Therefore, we have constructed a discrete-time Markov chain converging to the desired equilibrium distribution.

9.2. **Importance sampling for the Ising model via Metropolis algorithm.** For the 2D Ising model described in Section 5, the initial magnetic configuration can be set up arbitrarily. For example, all spins can be up, or each spin can be up or down with probability $1/2$ independently from the rest of the spins.

The matrix Q is usually chosen to make a flip of a random spin. It does not to be encoded explicitly. Instead, we pick a random spin and propose to flip it. Then we evaluate the energy difference between the current magnetic state and the proposed magnetic state. The energy of a spin state is given by (26).

Exercise Find the energy difference between the current spin state $\{s_{i,j}\}_{i,j=0}^{N-1}$ and the proposed spin state at which the spin at site (k,l) is flipped while the rest of the spins are the same.

The invariant measure for the Ising model is

$$f(\{s_{i,j}\}) = \exp\left(-\beta H(\{s_{i,j}\})\right).$$

Let H be the energy of the current state $\{s_{i,j}\}$ and H' be the energy of the proposed state $\{s'_{i,j}\}$. Then the acceptance probability α in (34) is given by

$$\alpha = \min\left\{\frac{f(\{s'_{i,j}\})}{f(\{s_{i,j}\})}, 1\right\} = \min\left\{\exp\left(-\beta(H'-H)\right), 1\right\}.$$

Therefore, if $\Delta H := H' - H > 0$, the proposed spin flip is accepted with probability $\alpha < 1$, while if $\Delta H := H' - H \leq 0$, the proposed spin flip is accepted for sure.

Thus, the mean magnetization $\mu(\beta)$ for the Ising model can be calculated using the Metropolis algorithm as outlined in the pseudo-code in Algorithm 2.

Algorithm 2: Calculation of the mean magnetization for the Ising model by the Metropolis algorithm

Initialization:
Set up an initial magnetic configuration.
Choose the number of MCMC steps `itermax`.
Choose β.
Set `iter` $= 0$.
Find the magnetization m by (28).
Set `mu` $= m$.
The main body:
for `iter` $= 1 :$ `itermax` **do**
  1: Randomly pick a site (k, l) and propose to flip the spin at it.
  2: Calculate the energy difference ΔH.
  if $\Delta H < 0$ **then**
    3: Set `accept = True`
  else
    4: Generate $u \sim \mathcal{U}(0, 1)$.
    if $u < \exp(-\beta \Delta H)$ **then**
      5: Set `accept = True`
    else
      6: Set `accept = False`
  if `accept == True` **then**
    6: Flip the spin that was proposed to flip.
    7: Calculate the magnetization m of the new state.
  8: Update the mean magnetization:

$$\texttt{mu} = \frac{\texttt{iter} * \texttt{mu} + m}{\texttt{iter} + 1}.$$

The mean magnetization for the 2D Ising model on a 30×30 lattice is displayed in Fig. 2. The number of iterations was set to 10^8.

9.3. **The Metropolis-Hastings algorithm.** The Metropolis-Hastings algorithm is a generalization of the Metropolis algorithms for the case where the matrix Q is not symmetric, i.e, $q_{ij} \neq q_{ij}$ for at least one pair of states (i, j). It differs from the Metropolis algorithm only by the definition of the acceptance probability α: in the Metropolis-Hastings, α is given by

$$\alpha = \min\left\{\frac{f_l}{f_k}\frac{q_{lk}}{q_{kl}}, 1\right\} \tag{36}$$

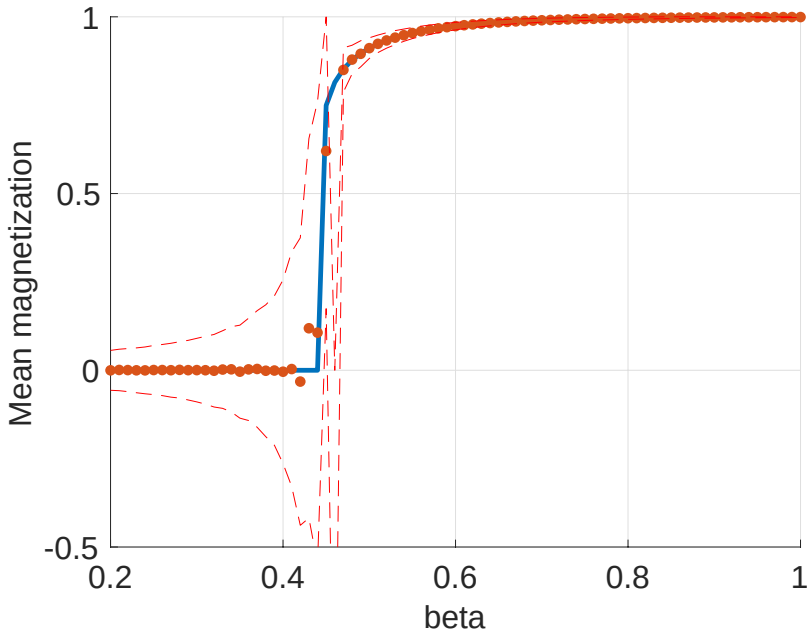

FIGURE 2. The mean magnetization $\mu(\beta)$ calculated using Algorithm 2. The blue curve is the analytical prediction by the formula (30). Red dots are the computed values of the mean magnetization. The dashed red curves are the computed magnetization $\pm$ standard deviation.

Therefore, the transition probabilities p_{ij} are

$$p_{ij} = q_{ij} \min\left\{\frac{f_j}{f_i}\frac{q_{ji}}{q_{ij}}, 1\right\}, \quad p_{ii} = 1 - \sum_{j\neq i} q_{ij} \min\left\{\frac{f_j}{f_i}\frac{q_{ji}}{q_{ij}}, 1\right\}. \tag{37}$$

Exercise Check that $P = (p_{ij})_{i,j\in S}$ and f are in detailed balance.

REFERENCES

[1] D. Bindel and J. Goodman, Principles of Scientific Computing, online book, 2009
[2] A. Chorin and O. Hald, Stochastic Tools in Mathematics and Science, 3rd edition, Springer 2013
[3] http://www.columbia.edu/ ks20/4703-Sigman/4703-07-Notes-ARM.pdf.
K. Sigman's lecture notes on the acceptance-rejection method for sampling random variables.

[4] J. Weare, Lecture notes for the course "Stochastic Simulation", Fall 2017. (*Not publicly available.*)
[5] J. R. Norris, "Markov Chains", Cambridge University Press, 1998