MATH858D

SAMPLING AND MONTE-CARLO INTEGRATION

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 with the purpose of just looking at them. Typically, the error in Monte-Carlo methods decays as $n^{-1/2}$ where n is the number of samples which is worse that the error decay rate in most of deterministic methods (it is usually at least as good as n^{-1}). So, why bother? The reason is that in some important situations deterministic methods simply cannot be used due to such things as the "curse of dimensionality" or largeness 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 need 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 $[0 \ldots \text{RAND\_MAX}]$, where `RAND_MAX` is a constant defined in the library `stdlib.h`. It is platform-dependent. In most modern computers,

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

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 $\text{RAND\_MAX}] = 2147483647 \sim 2 \cdot 10^9$ as above, do not use `rand()` if you you need more that 10^7 samples.

```c
#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 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.

3. Sampling random variables with given distribution

The goal of sampling is typically 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 hard and Monte-Carlo methods come in handy. 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 with a pdf $f(x)$. Assume that we can integrate $f(x)$ analytically, i.e., have an analytic expression for the probability distribution function $F(x)$ (a.k.a. the cumulative distfiputiion function or CDF). Assume that $f(x) \equiv 0$ for $x, 0$. We observe that

$$\int_0^\eta f(x) dx = F(\eta) = \xi. \quad \text{Hence} \quad \eta = F^{-1}(\xi),$$

where $F^{-1}(\xi)$ is the inverse function of F. It exists if $F(x)$ is strictly increasing. If $\xi \sim \mathcal{U}(0,1)$ is uniformly distributed on $[0,1]$, i.e., its probability distribution function is

$$F_\xi(x) = P(\xi \leq x) = \begin{cases} 1, & x \geq 1, \\ x, & 0 \leq x < 1, \\ 0, & x < 0, \end{cases}$$

then the probability distribution of η is $F(x)$. Indeed,

$$P(\eta \leq x) = P(F^{-1}(\xi) \leq x) = P(\xi \leq F(x)) = F(x).$$

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

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

The probability distribution of ξ is given by

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

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

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

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

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

The limitation of the inversion method is due to the fact that not all pdf's are analytically integrable to CDF's.

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

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

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

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

Now let us to switch to the polar coordinates

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

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 $(X_1, \ldots X_n) \in \Omega_X$ with the joint pdf $f_{X_1,\ldots,X_n}(x_1,\ldots,x_n)$. The desired RVs $(Y_1, \ldots, Y_n) \in \Omega_Y$ are functions of X_1, ..., X_n: $Y_i = g_i(X_1, \ldots, X_n)$, $i = 1, \ldots, n$. 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 solved 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

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

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:

$$(5) \qquad f_{R,\Theta}(r,\theta) = \frac{1}{2\pi\sigma^2}e^{-\frac{r^2}{2\sigma^2}}r,$$

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:

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

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:

$$(6) \qquad a = \sqrt{-2\sigma^2 \log \xi_1}.$$

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

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

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 are able to find a variable change such that the new variables have analytically integrable marginal pdf's and hence can be obtained e.g. using the inversion method. This method 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.** Read a very nice note on it by Prof. K. Sigman (Columbia university):
http://www.columbia.edu/ ks20/4703-Sigman/4703-07-Notes-ARM.pdf.
I will only add a few remarks.

- Contrary to the inversion method and the method of variable change, the acceptance-rejection method is universal method in the sense that, as soon as we can find a RV with pdf $g(x)$ which we can sample and

$$\sup_{x\in\mathbb{R}} \frac{f(x)}{g(x)} \quad \text{is bounded,}$$

which is typically not hard, it will work. However, if this bound is much larger that 1, the method will be wasteful.
- The following fact was used in the note. If we need to find the probability of the event that one RV η with pdf $f_\eta(x)$ is less or equal to another RV ξ with pdf $f_\xi(x)$, i.e., $P(\eta \le \xi)$, we can do it using conditioning. If ξ takes a discrete set of values ξ_i, $i \in I$, where I some not more than countable index set, then we have

$$(8) \qquad P(\eta \le \xi) = \sum_{i\in I} P(\eta \le \xi | \xi = \xi_i) P(\xi = \xi_i).$$

The analogous formula for continuous RVs is

$$(9) \qquad P(\eta \le \xi) = \int_{-\infty}^{\infty} P(\eta \le \xi | \xi = x) f_\xi(x) dx.$$

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 from Sigman's note. Figure 1 ins generated by the code.

```
function AcceptRejectMethod()
%% Generate eta ~ N(0,1) using exp(-xi)
N = 1e8; % the number of samples
xi = rand(N,2);
y1 = -log(xi(:,1));
y2 = -log(xi(:,2));
% generate signs for eta
u = rand(N,1);
ind = find(u > 0.5);
s = ones(N,1);
s(ind) = -1;
%
f = 0.5*(y1 - 1).^2;
ind = find(y2 >= f);
Na = length(ind); % the number of accepted RVs
eta = y1(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
```

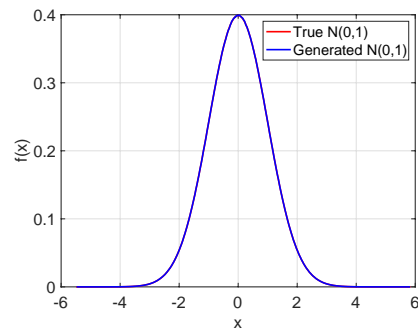

FIGURE 1. Figure generated by the code in Section 3.3.

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.,

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

Suppose we are able to 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,

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

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

$$(10) \qquad\qquad\qquad \frac{1}{n} \sum_{i=1}^n g(\eta_i)$$

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

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

Therefore, the error of the estimate (10) is of the order of

$$(11) \qquad\qquad\qquad err \sim \frac{\sqrt{\text{Var}(g(\eta))}}{\sqrt{n}}.$$

Eq. (11) 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 of $g(\eta)$. Note that increasing the number of samples is not very efficient approach, as the error decays as $n^{-1/2}$. A better idea is to try 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

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

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

In order to reduce $\text{Var}(g(\eta))$ we can try to find a function $h(x)$ with the following properties:

(1) The integral

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

is easy to evaluate;

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

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

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

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

Then we have

$$I = \int_a^b g(x)f(x)dx = \int_a^b \frac{g(x)}{h(x)}f(x)h(x)dx = I_1 \int_a^b \frac{g(x)}{h(x)} \frac{f(x)h(x)}{I_1}dx$$

$$= I_1 E\left[\frac{g}{h}(\eta)\right] \sim \frac{I_1}{n} \sum_{i=1}^n \frac{g(\eta_i)}{h(\eta_i)},$$

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

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

in [2].

4.2. **Finding normalization constants.** This section is based on notes by Prof. J. Weare (NYU) [4]. It is often the case in statistical mechanics and chemical physics that the invariant pdf $\pi(x)$ is known only up to the normalization constant, i.e.

(12) $$\pi(x) = \frac{p(x)}{Z_p}, \quad \text{where } p(x) \text{ is known and } Z_p = \int_{\mathbb{R}^d} p(x)dx \text{ is not.}$$

In 1D, one can find Z by numerical integration, e.g., using the composite trapezoidal rule. As the dimension grows, numerical quadrature becomes less and less feasible.

Alternatively, independent of the dimensionality of the system, one can find an estimate for Z using some other pdf $\gamma(x) = Z_g^{-1}g(x)$ where both $g(x)$ and Z_g are known and that one are able to sample from. We observe that

$$\int_{\mathbb{R}^d} \frac{p(x)}{g(x)}\gamma(x)dx = \int_{\mathbb{R}^d} \frac{p(x)}{g(x)}\frac{g(x)}{Z_g}dx = \frac{1}{Z_g}\int_{\mathbb{R}^d} p(x)dx = \frac{Z_p}{Z_g}.$$

On the other hand, this integral can be evaluated by sampling from the pdf $\gamma(x) = Z_g^{-1}g(x)$. Hence

$$Z_p = Z_g \int_{\mathbb{R}^d} \frac{p(x)}{g(x)}\gamma(x)dx \approx \frac{Z_g}{N} \sum_{i=1}^N \frac{p(\xi_i)}{g(\xi_i)}.$$

Note that in the case where Z_g is also unknown, but one still is able to sample from the pdf $\gamma(x) = Z_g^{-1}g(x)$, one can evaluate the ratio of normalization constants. In applications

coming from chemical physics, finding ratio of normalization constants is the basis for finding so-called *free energy* differences where *free energy* is defined by $F = -\log Z$. Suppose we are considering a family of pdf's depending on a parameter θ (e.g. temperature). Then $F(\theta) = -\log Z_{p(\theta)}$ and

$$F(\theta_2) - F(\theta_1) = -\log Z_{p(\theta_2)} + \log Z_{p(\theta_1)} = \log \frac{Z_{p(\theta_1)}}{Z_{p(\theta_2)}}.$$

4.3. **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,

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

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

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

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

$$(21) \qquad |S| = \int_S d\sigma.$$

Example 17 Consider the integral

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

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

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

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:

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

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

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

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 $g(x) = x_1^2 x_2^2$, the exact integral (22) is

$$I = \int_{S_3} x_1^2 x_2^2 d\sigma = \frac{\pi^2}{12} = 0.8224670\ldots,$$

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. Estimators, estimates and sampling distributions

Statistics is concerned with finding the distribution given a set of samples. Here we will go over basic concepts of statistics that we need obtaining integrals by Monte Carlo methods.

- Consider a fixed but unknown parameter $q \in \mathbb{R}^d$. A vector in $\mathbb{R}^d$ that represents q is called the point estimate. An interval estimate provides an interval that quantifies the plausible location of components of q.
- An estimator $g(\eta_1, \ldots, \eta_n)$ is a rule or procedure that specifies how to construct estimates for q based on random samples of random variable η. Hence the estimator is a random variable with an associated distribution which quantifies attributes of estimation process. An estimate is a realization of the estimator. Therefore, it is a function of realized values x_1, ..., x_n, i.e., a sample.

Often we need to estimate the expected value of a random variable. The standard estimator for the mean is the running average

$$(26) \qquad \bar{\eta}(\eta_1, \ldots, \eta_n) = \frac{1}{n} \sum_{i=1}^{n} \eta_i.$$

Often we also need to estimate the variance to assess the error. Below are three estimators each of which is optimal in some sense:

$$(27) \qquad S_1^2(\eta_1, \ldots, \eta_n) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\eta_i - \frac{1}{i} \sum_{k=1}^{i} \eta_k \right)^2,$$

$$(28) \qquad S_2^2(\eta_1, \ldots, \eta_n) = \frac{1}{n} \sum_{i=1}^{n} \left(\eta_i - \frac{1}{n} \sum_{k=1}^{n} \eta_k \right)^2,$$

$$(29) \qquad S_3^2(\eta_1, \ldots, \eta_n) = \frac{1}{n-1} \sum_{i=1}^{n} \left(\eta_i - \frac{1}{n} \sum_{k=1}^{n} \eta_k \right)^2.$$

The estimator S_1^2 is called the running variance. It has the advantage that it is not necessary to store samples in memory in order to calculate it. This is not so for S_2^2 and S_3^2. The estimator S_2^2 has the maximal likelihood (see below). The estimator S_3^2 is unbiased (see below).

5.1. **Unbiased estimators.** An estimator for q is said to be unbiased if

$$E[g(\eta_1, \ldots, \eta_n)] = q.$$

Example 2   Suppose $E[\eta] = m$. Show that the estimator (26) for the mean is unbiased:

$$E[\bar{\eta}] = \frac{1}{n} \sum_{i=1}^{n} E[\eta_i] = \frac{1}{n} \sum_{i=1}^{n} m = m.$$

Example 3   Suppose $\mathrm{Var}(\eta) = \sigma^2$. Consider the running variance as an estimator for the variance

$$S_1^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(\eta_i - \frac{1}{i} \sum_{k=1}^{i} \eta_k \right)^2.$$

This is a biased estimator as

$$E[S_1^2] = \frac{1}{n-1} \sum_{k=1}^{n} E\left[\left(\eta_k - \frac{1}{k}\sum_{i=1}^{k}\eta_i\right)^2\right]$$

$$= \frac{1}{n-1} \sum_{k=1}^{n} E\left[\left(\eta_k - m + m - \frac{1}{k}\sum_{i=1}^{k}\eta_i\right)^2\right]$$

$$= \frac{1}{n-1} \sum_{k=1}^{n} \left(E[(\eta_k - m)^2] - 2E\left[(\eta_k - m)\frac{1}{k}\sum_{i=1}^{k}(\eta_i - m)\right] + E\left[\left(\frac{1}{k}\sum_{i=1}^{k}(\eta_i - m)\right)^2\right]\right)$$

$$= \frac{1}{n-1} \sum_{k=1}^{n} \left(\sigma^2 - \frac{2}{k}\sigma^2 + \frac{1}{k}\sigma^2\right) = \frac{1}{n-1}\sum_{k=1}^{n}\sigma^2\left(1-\frac{1}{k}\right) = \sigma^2\frac{1}{n-1}\sum_{k=1}^{n}\left(1-\frac{1}{k}\right)$$

$$= \sigma^2\left(1 + \frac{1}{n-1} - \frac{1}{n-1}\sum_{k=1}^{n}\frac{1}{k}\right) \neq \sigma^2.$$

Note that

$$\lim_{n\to\infty} \sigma^2\left(1 + \frac{1}{n-1} - \frac{1}{n-1}\sum_{k=1}^{n}\frac{1}{k}\right) = \lim_{n\to\infty} \sigma^2\left(1 + \frac{1 - \log n}{n-1}\right) = \sigma^2.$$

Exercise Show that

$$S_3^2(\eta_1, \ldots, \eta_n) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\eta_i - \frac{1}{n}\sum_{k=1}^{n}\eta_k\right)^2$$

is an unbiased estimator of the variance.

Definition 1. *A statistic is a measurable function of one or more random variables that does not depend on unknown parameters.*

For example, the estimators for the mean and variance that we have mentioned are statistics.

5.2. **Maximum likelihood estimator.**

Definition 2. *Let $f_\eta(x; q)$ be a parameter-dependent joint pdf associated with the random vector $\eta = [\eta_1, \ldots, \eta_n]$ where $q \in Q$ is the unknown parameter vector, and $x = [x_1, \ldots, x_n]$ be a realization of η. The likelihood function $L : Q \to [0, \infty)$ is defined as*

$$L_x(q) = f_\eta(x; q),$$

where the observed sample $x = [x_1, \ldots, x_n]$ is fixed and q varies over all admissible parameter values.

The likelihood function is proportional to the probability to observe the given sample if the parameter vector is equal to q. Therefore, to find the maximal likelihood estimate we need to maximize the likelihood function.

For n i.i.d. random variables, the likelihood function becomes

$$L_x(q) = \prod_{i=1}^{n} f_\eta(x_i; q).$$

Often it is computationally advantageous to deal with $\log L_x(q)$ rather than with $L_x(q)$, because products are replaced with sums, and powers are replaced with multiplications. The log-likelihood function is denoted by $l_x(q)$ and defined by

$$l_x(q) = \log L_x(q).$$

Note that for n i.i.d. random variables

$$l_x(q) = \sum_{i=1}^{n} \log f_\eta(x_i; q).$$

Due to monotonicity of the logarithm, maximizing the likelihood function is equivalent to maximizing the log-likelihood function.

Example 4 Suppose we believe that η is a Gaussian random variable, and we wish to estimate its mean and variance using a sample of n independent trials $x = [x_1, \ldots, x_n]$. Then the parameter vector is $q = [m, \sigma^2]$. The likelihood function is given by

$$L_x(m, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - m)^2}{2\sigma^2}},$$

while the log-likelihood function is

$$l_x(m, \sigma^2) = \sum_{i=1}^{n} \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{(x_i - m)^2}{2\sigma^2} \right]$$

$$= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - m)^2.$$

To find the maximum likelihood estimate for m we compute $\frac{\partial l_x(m, \sigma^2)}{\partial m}$ and set it to zero:

$$\frac{\partial l_x(m, \sigma^2)}{\partial m} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} (m - x_i) = -\frac{1}{\sigma^2} \left(nm - \sum_{i=1}^{n} x_i \right) = 0.$$

Hence, the maximal likelihood estimate for the mean is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

To find the maximum likelihood estimate for σ^2 we compute $\frac{\partial l_x(m,\sigma^2)}{\partial \sigma^2}$ and set it to zero:

$$\frac{\partial l_x(m,\sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - m)^2 = \frac{1}{2\sigma^4}\left(n\sigma^2 - \sum_{i=1}^{n}(x_i - m)^2\right) = 0.$$

Hence, the maximal likelihood estimate for the variance is

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2.$$

Note that the maximal likelihood estimate for the variance is biased.

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] Wikipedia