Introduction to Monte Carlo Methods

Arun Nayak[1]

Online workshop on
Software Tools and Techniques used in EHEP and its Applications
MNIT, Jaipur

12 July 2021

[1] Institute of Physics, Bhubaneswar

References

1. G. H. Givens, J. A. Hoeting, Computational Statistics
2. M. Hjorth-Jensen, Computational Physics (Lecture Notes)
3. C. M. Bishop, Pattern Recognition and Machine Learning
4. T. Pang, An introduction to Computational Physics
5. https://martin-haugh.github.io/teaching/monte-carlo/
6. P. R. Bevington and D. K. Robinson, Data Reduction and Error Analysis
7. G. Cowan, Statistical Data Analysis

Monte Carlo Techniques – Introduction

It is a numerical technique for calculating probabilities and related quantities by using sequences of random numbers.

The usual procedure:

1. Generate a sequence of random values $r_1, r_2, ..., r_n$ according to uniform distribution, $0 < r < 1$.
2. Use this to determine another sequence $x_1, x_2, ...$, distributed according to some pdf $f(x)$ in which one is interested.
3. The values of x are treated as simulated measurements, and used to compute probabilities for x to be in a certain region, e.g. $P(a < x < b) = \int_a^b f(x)dx$.

MC calculation ⇔ integration

The technique is most useful when other methods are not feasible to do the integration, e.g. integration of a joint pdf $f(\vec{x})$ over a complicated region.

Basic Ingredients

At least four crucial ingredients needed to understand the basic Monte-Carlo Strategy:

1. Random variables,
2. probability distribution functions (PDF),
3. moments of a PDF
4. and its pertinent variance σ^2.

Example-1

Consider tossing of two dice:
What are the outcomes and their corresponding probabilities?

It will yield following possible outcomes:
$\{2,3,4,5,6,7,8,9,10,11,12\} \rightarrow$ These values are called the "domain".

The corresponding probabilities are:
$\{1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36\}$

One cannot tell beforehand whether the outcome of a toss will be 3 or 5 or any other number in this domain $\implies$ **Randomness** of the outcome. The only thing we can tell beforehand is that an outcome has a certain probability.

Hence, Random variables are characterized by a domain which contains all possible values that the random value may take. This domain has a corresponding PDF.

Example-2

Consider the radioactive decay of an α-particle from a certain nucleus.
Suppose we have a GM counter that registers every 10 ms whether an α-particle reaches the counter. 1 observation $=$ 1 hit, no observation $=$ 0.
If we repeat this experiment for a long time the outcomes of the experiment will be truly random $\implies$ can not form a specific pattern from the above observations.
The only possibility to say something about the outcome is given by the PDF. In this case it is well-known exponential function:

$$\frac{1}{\mu} exp\left(-\frac{x}{\mu}\right)$$

μ is the half-life of the nucleus that decays.

Random Numbers, PDF, CDF, ...

- Random numbers are numerical approximations to the statistical concept of stochastic variables, sometimes just called random variables.
- A stochastic variable can be either continuous or discrete. Let's denote it here as, X, Y etc...
- The domain is the set $D = \{x\}$ of all accessible values that the variable can have, so that $X \in D$.
- The *probability distribution function* (PDF) is a function $p(x)$ on the domain, such that,
  in descrete case $p(x) = Prob(X = x)$ and
  in case of continuous, $Prob(a \leq X \leq b) \ = \ \int_a^b p(x)dx$.
  Must be Positive: $0 \ \leq \ p(x) \ \leq \ 1$
  and Normalized: $\int_{x \in D} p(x)dx \ = \ 1$.

Random Numbers, PDF, CDF, ...

- Also interest to us is the *cumulative distribution function* (CDF), P(x), given by
  $P(x) \;=\; Prob(X \leq x) \;=\; \int_{-\infty}^{x} p(x')dx'$
  $\implies \;\; p(x) \;=\; \frac{d}{dx}P(x)$
- The n-th moment of the PDF p is defined as
  $\langle x^n \rangle \;\equiv\; \int x^n p(x)dx$
  The first moment $\langle x \rangle = \mu$ is called the "mean": $\langle x \rangle \;=\; \mu$.
- Similarly, the n-th central moment is defined as:
  $\langle (x - \langle x \rangle)^n \rangle \;\equiv\; \int (x - \langle x \rangle)^n \, p(x)dx$
  The 2nd central-moment (variance) is:
  $\sigma_X^2 \;=\; Var(X) \;=\; \left\langle (x - \langle x \rangle)^2 \right\rangle \;=\; \langle x^2 \rangle \;-\; \langle x \rangle^2.$

PDF of a function

Let $Y = h(X)$ be a function of X.
If $p_X(x)$ is pdf of X, What is pdf of Y, $p_Y(y)$?

Consider cases when $h(X)$ is invertible, so that it has to be strictly monotonous.
Construct CDF of Y, considering only the case where h increases:

$$\begin{aligned} P_Y(y) &= Prob(Y \leq y) = Prob(h(x) \leq y) \\ &= Prob(X \leq h^{-1}(y)) = P_X(h^{-1}(y)) \end{aligned}$$

So, PDF of Y:

$$p_Y(y) = \frac{d}{dy} P_Y(y) = \frac{d}{dy} P_X(h^{-1}(y))$$

Similarly, for decreasing h,

$$p_Y(y) = p_X(h^{-1}(y)) | \frac{d}{dy} h^{-1}(y) |$$

Random Number Generation

Goal is to generate a sequence of numbers which are distributed randomly according to a uniform probability distribution.

Desired Property:

- Long Periodicity: e.g. if a 32-bit interger is used the period should be close to $2^{31} - 1 = 2147483647$.
- Best Randomness: The correlation among the generated numbers should be small, i.e. $< x_i x_{i+l} >$ should have a uniform distribution for $l \neq 0$.

Two types of generators:

- **True Random number generator:** Based on physical phenomenon, such as radioactive decay, atmospheric noise etc..
- **Pseudorandom number generator (PRNG):** Computational algorithms producing long sequences of apparently random results.

Pseudorandom number generator (PRNG)

- Not truly random → Always produces same sequence of numbers if input is same.
- The series of numbers generated by these computational algorithms is generally determined by a fixed number, called a **seed**.
- Can be used as random numbers if the sequence of numbers have good random properties
- Advantage: Speed and reproducibility
- Hence, are central in applications such as Monte Carlo simulations

Most common: **multiplicative linear congruential generator (MLCG)**

$$x_{i+1} = (ax_i + b)\ mod\ c.$$

$a,\ b,\ c$ are large integers, determine the quality of generator.

Exercise: Write a program to generate uniform random number in $[0, 1]$, with $a = 7^5,\ b = 0,\ c = 2^{31} - 1$

Uniform random number generator - contd..

For the generator to produce different sequence of random numbers everytime, the initial seed can be changed.
There can be a systematic way of obtaining different seeds.
e.g. We can use the current time to get a seed.
$i_s = t_6 + 70(t_5 + 12\{t_4 + 31[t_3 + 23(t_2 + 59t_1)]\})$
where,
$0 \leq t_1 \leq 59$ for second
$0 \leq t_2 \leq 59$ for minute
$0 \leq t_3 \leq 23$ for the hour
$0 \leq t_4 \leq 31$ for the day
$0 \leq t_5 \leq 12$ for the month
t_6 is the current year
It is roughly in the region $[0, 2^{31} - 1]$, and the result would be different if the time is even one second apart.

Mersenne Twister generator

- It is a pseudorandom number generator
- Most widely used currently
- The period length is chosen to be a Mersenne prime.
- The common algorithm is based on the Mersenne prime $2^{19937}-1 \implies$ a long period of $2^{19937}-1$
- It is also relatively faster

Reference:

1) Matsumoto, M., Nishimura, T. “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator”, ACM Transactions on Modeling and Computer Simulation. 8 (1): 3–30 (1998)

2) Harase, S., “Conversion of Mersenne Twister to double-precision floating-point numbers”, Mathematics and Computers in Simulation, 161: 76–83 (2019), arXiv:1708.06018.

Random number generator - contd..

There are several classes in ROOT to generate random numbers. e.g.,

TRandom: A Linear Congruential Generator. Periodicity is only 2^{31}.

TRandom1: Based on the “RANLUX algorithm”. Much slower than others.

TRandom2: Based on the “Tausworthe generator of L’Ecuyer”. Fast, periodicity is about 10^{26}.

TRandom3: Based on the “Mersenne Twister generator”. Fast and long period of about 10^{6000}.

.....

Use of MC method

Consider the evaluation of an integral using numerical methods

$$I = \int_0^1 f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i)$$

where, w_i are weights at grid points x_i (to be evaluated by, say, Simpson's method)

In the simple midpoint or rectangle method, $w_i = 1$,

$$I = \int_0^1 f(x)dx \approx h \sum_{i=1}^{N} f(x_{i-1/2})$$

Since, $h = (b-a)/N = 1/N$,

$$I = \int_0^1 f(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} f(x_{i-1/2})$$

$x_{i-1/2}$ are midpoint values of x.

Use of MC method

The average of a function $f(x)$ for a given pdf $p(x)$

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} p(x_i) f(x_i)$$

If we consider a uniform pdf, i.e. $p(x) = 1$ for $x \in [0, 1]$,

$$I = \int_0^1 f(x) dx \approx \langle f \rangle$$

So, the integral is nothing but the average $\langle f \rangle$ evaluated using random numbers x_i distributed uniformly between 0 and 1.
This approach is often called 'crude' or 'Brute-Force' Monte-Carlo method.

Accuracy of MC method

Let's calculate the variance σ^2:

$$\sigma_f^2 \;=\; \left\langle (f \;-\; \langle f \rangle)^2 \right\rangle \;=\; \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - \langle f \rangle)^2 p(x_i)$$

for uniform distribution

$$\sigma_f^2 \;=\; \frac{1}{N}\sum_{i=1}^{N}(f(x_i))^2 \;-\; \left(\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right)^2$$

or

$$\sigma_f^2 \;=\; \left(\left\langle f^2 \right\rangle \;-\; \langle f \rangle^2\right)$$

which is nothing but a measure of the extent to which f deviates from its average over the region of integration.

Accuracy of MC method - contd..

Let's consider the previous result for a fixed value of N as "one measurement".

Suppose we recalculate the average and variance for a series of M different measurements. Then, we can write the integral as the average of M such averages:

$$\langle I \rangle_M \ = \ \frac{1}{M} \sum_{l=1}^{M} \langle f \rangle_l$$

Considering the probability of correlated events to be zero, the variance of these series of measurements will be

$$\sigma_M^2 \ \approx \ \frac{1}{M} \left(\langle f^2 \rangle \ - \ \langle f \rangle^2 \right) \ = \ \frac{\sigma_f^2}{M}$$

i.e., the standard deviation

$$\sigma_M \sim \frac{1}{\sqrt{M}}$$

Thus, the aim of Monte Carlo calculations is to have σ_M as small as possible after M samples.

Accuracy of MC method - contd..

Let's compare it to any numerical integration based on Taylor expansion, e.g., trapezoidal rule.
The error goes as $O(h^k)$, $k = 2$ for trapezoidal rule.
where, $h = (b-a)/N$, the step size.
That means, the error goes as $\sim N^{-k}$.

Consider integration in higher dimension: Suppose integration volume is a hypercube with side L and dimension d.
The number of points in the cube: $N = (L/h)^d$. $\implies$ the error $\sim N^{-k/d}$.

If we perform the same integration using MC method, the error $\sigma \sim 1/\sqrt{N}$.

Thus, for dimention $d > 2k$ MC method has better accuracy than the traditional numerical integration methods.

Exercises

1. Calculate the value of the integral and error using uniform sampling Monte Carlo program, for 10^6 sampling points:

$$I \;=\; \int_0^1 x^2 dx$$

2. Perform the following integration using a brute force Monte Carlo program and compare to the exact result:

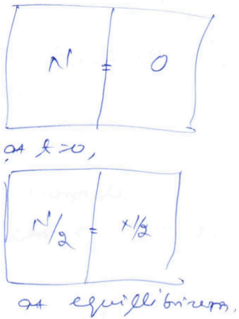
$$I \;=\; \int_0^1 \frac{dx}{1+x^2}$$

   Find the accuracy of the integral as function of number of MC samples N.

Example: Particles in a box

Let, time $t = 0$, there are N particles on the left side. One particle can pass through the hole per unit time. After some time the system reaches its equilibrium state with equally many particles in both halves, $N/2$.

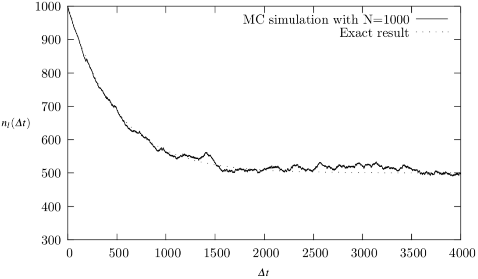
We can simulate this system. Assume, all particles in left have equal probability to go to right.
At a given t, n_l particles in left and $n_r = N - n_l$ particles in right.
For each time step, Δt, the probability to move right is n_l/N.



Example: Particles in a box (2)

Steps to simulate:

- loop over time steps, choose $t_{max} > N$
- for each Δt, generate an uniform r.n. $0 < x < 1$
- if $x < n_l/N$, move one particle from left to right, else move one from right to left.



Analytic solution: $n_l(t) \;=\; \frac{N}{2}(1 + e^{-2t/N})$ (derive it)

Exercise: The nucleus ^{210}Bi decays through β-decay to ^{210}Po. ^{210}Po further decays through emision of an α-particle to ^{206}Pb, which is a stable nucleus. ^{210}Bi has a mean lifetime of 7.2 days while ^{210}Po has a mean lifetime of 200 days. Suppose, at time t=0 we have 10^4 ^{210}Bi nuclei and zero ^{210}Po nuclei. Write a simulation program to compute the number ^{210}Po nuclei remaining as a function of time. Plot this distribution as a function of time.

If a nucleus X decays to a daughter nucleus Y which can also decay, we get the coupled equations

$$\begin{aligned} \frac{dN_X(t)}{dt} &= -\omega_X N_X(t) \\ \frac{dN_Y(t)}{dt} &= -\omega_Y N_Y(t) + \omega_X N_X(t) \end{aligned}$$

where $\omega = \frac{1}{\tau}$.
Analytic solution:
$N_Y(t) = \frac{\omega_X}{\omega_Y - \omega_X} N_X^0 (e^{-\omega_X t} - e^{-\omega_Y t}) + N_Y^0 e^{-\omega_Y t}$ (derive it)

Central Limit Theorem

Suppose we generate a series of random variables x_i from a pdf $p(x)$. The mean and standard deviation of $p(x)$ are μ and σ, respectively. Compute the mean of m such values

$$z = \frac{x_1 + x_2 + ... + x_m}{m}$$

In the limit $m \rightarrow \infty$, the pdf of new variable z will be a normal distribution:

$$\tilde{p}(z) = \frac{1}{\sqrt{2\pi}(\sigma/\sqrt{m})} exp\left(-\frac{(z-\mu)^2}{2(\sigma/\sqrt{m})^2}\right)$$

The mean of $\tilde{p}(z)$ is the mean of $p(x)$ and the variance of $\tilde{p}(z)$ is the variance of $p(x)$ divided by m. i.e.,

$$\sigma_m = \frac{\sigma}{\sqrt{m}}$$

Sum of two uniform random numbers

We know pdf $f(z)$ of the sum $z = x + y$ is

$$f(z) = \int_{-\infty}^{\infty} g(x)h(z-x)dx = \int_{-\infty}^{\infty} g(z-y)h(y)dy$$

If x and y are uniformly distributed between 0 and 1,

$$f(z) = \int_{0}^{1} h(z-x)dx$$

We know, $z_{min} = 0$ and $z_{max} = 2$.
$h(z-x) = 1$ for some range of z and 0 otherwise.
Let's consider two ranges, $0 < z \leq 1$ and $1 < z < 2$.

Case-1: $0 < z \leq 1$, $h(z-x) = 1$ when $z - x \geq 0$ or $x \leq z$.
$\implies \int_0^1 h(z-x)dx = \int_0^z dx = z$

Case-2: $1 < z < 2$, $h(z-x) = 1$ when $z - x \leq 1$ or $x \geq z - 1$
$\implies \int_0^1 h(z-x)dx = \int_{z-1}^1 dx = 2 - z$

Thus, $f(z)$ triangular with peak at 1.

Addition of random numbers

Exercise-1: Generate two independent uniform random numbers x and y, distributed between 0 and 1. Plot the distribution of $z = x + y$ and show that your distribution matches to the one derived in last slide.

Exercise-2: Generate $N (\geq 20)$ independent uniform random numbers, between 0 and 1. Then, plot the distribution of $z = \frac{\sum x_i}{N}$. What distribution do you get? Verify your result against central limit theorem.

Exercise-3: Repeat exercise-2 for a mixture of Exponential and Poission distributed random numbers.

Exercise-4: Repeat exercise-2 for Breit-Wigner and Landau (separately) distributed random numbers. Does it obey C.L.T? Why not? Find the reason.

Variable Transformation

Given uniform random numbers x_i in $[0,1]$, find y_i that are distributed according to some pdf $f(y)$ by finding a suitable transformation.
Using conservation of probabilities:

$$\begin{array}{rcl} p(x)dx & = & f(y)dy \implies dx = f(y)dy \\ \implies x(y) & = & F(y) = \int_{-\infty}^{y} f(y')dy' \\ \implies y & = & F^{-1}(x). \end{array}$$

Example: Transformed uniform distribution:
Suppose we need $f(y) = \frac{1}{b-a}, \ a \le y \le b$.

$$\begin{array}{rcl} p(y)dy & = & \frac{dy}{b-a} = dx \\ \implies x(y) & = & \int_a^y \frac{dy'}{b-a} \\ \implies y & = & a + (b-a)x \end{array}$$

Example of the transformation method

Exponential pdf

Assume $f(y) \;=\; e^{-y}$, and $p(x) = 1$ for $x \in [0, 1]$.

$$\begin{array}{rcl} \Longrightarrow \;\; dx & = & f(y)dy \;=\; e^{-y}dy \\ \Longrightarrow \;\; x(y) & = & \int_0^y e^{-y'}dy' \;=\; 1 - e^{-y} \\ \Longrightarrow \;\; y(x) & = & -\,ln(1-x) \end{array}$$

When x is generated uniformly, y will be exponentially distributed.

Example of the transformation method

Gaussian pdf

Difficult to find the inverse, since the cdf is error function:

$$F(x) \;=\; erf(x) \;=\; \tfrac{2}{\pi} \int_0^x e^{-t^2} dt$$

Solution: Generate uniform distribution $p(\phi) = 1$ for $\phi \in [0, 2\pi]$ and exponential $f(t) \;=\; e^{-t}$ for $t \in [0, \infty]$. From this we will obtain two gaussian distributions $g(x)$ and $g(y)$.

$$\begin{aligned} \tfrac{1}{2\pi}\, p(\phi) d\phi f(t) dt \;&=\; g(x) dx \; g(y) dy \\ \Longrightarrow \;\; e^{-t} dt d\phi \;&=\; e^{-(x^2+y^2)/2} dx dy \end{aligned}$$

This is a coordinate transformation from polar (ρ, ϕ) with $\rho = \sqrt{2t}$ to rectangular (x, y).

$$\Longrightarrow \;\; x \;=\; \sqrt{2t}\; cos(\phi) \quad y \;=\; \sqrt{2t}\; sin(\phi)$$

x and y will be distributed with Gaussian pdf.

Exercise: Consider a uniform 10-cm long rod. One end of the rod is held at 0^oC and the other at 100^oC so that the temperature along the rod is expected to vary lineraly from 0^oC to 100^oC. **Simulate the data** that would be obtained by measuring the temperature at regular intervals along the rod.

Assume that the parent population is described by the equation

$$T \;=\; a_0 \;+\; b_0 x$$

with $a_0 \;=\; 0^oC$ and $b_0 \;=\; 10^oC/cm$, and that 10 measurements are made at 1-cm intervals from $x \;=\; 0.5$ to $x \;=\; 9.5\; cm$, with negligible uncertainties in x_i and uniform measuring uncertainties in T_i of $\sigma_T \;=\; 1.0^oC/cm$.

Exercise: Consider an exponential pdf (e.g. representing the decay of a certain type of unstable particles).

$$f(t;\tau) = \frac{1}{\tau} e^{-t/\tau}$$

Suppose we have data for n decays and we want to estimate the value of τ. We can use the *Maximum Likelihood method* and maximize the **log-likelihood function** with respect to τ:

$$L(\tau) = \prod_{i=1}^{n} f(t_i;\tau) \implies logL(\tau) = \sum_{i=1}^{n} f(t_i;\tau) = \sum_{i=1}^{n} \left(log\frac{1}{\tau} - \frac{t_i}{\tau} \right)$$

Thus,

$$\frac{\partial logL(\tau)}{\partial \tau} = 0 \implies \hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} t_i$$

You can also show that $\langle \hat{\tau} \rangle = \tau$, i.e. $\hat{\tau}$ is an unbiased estimator for τ.

Q. Now generate a set of r.n. based on the above pdf (using a certain value of τ) and compute $\hat{\tau}$. Show that $\hat{\tau}$ approaches τ in the limit of large n.

Difficulties in the transformation method

Suppose we want to generate random numbers distributed according to the pdf:

$$p(y) \;=\; A(1+ay^2) \;\; \text{for} \; -1 \le y < 1$$

A is constantto normalize $p(y)$, i.e.

$$\int_{-1}^{1} p(y)dy \;=\; 1$$

So, we have,

$$x \;=\; \int_{-1}^{y} A(1+ay^2)dy, \quad x \in [0,1]$$

That gives,

$$x \;=\; A(y + ay^3/3 + 1 + a/3)$$

So, to get y,we must solve the third-degree equation. Analytic solution may not always be possible, **numerical calculations** are necessary.

Steps for transformation method

1. Decide on the range of y. If range is ∞, use some reasonable finite limits.
2. Normalize the Prob. density. If range has been adjusted renormalize it with that range using the same analytic or numerical integration routine which is used find the y.
3. Generate a random variable x drawn from the uniform distribution
4. Integrate the normalized prob. function $p(y)$ from lower limit to the value $y = y$, where y satisfies the equation $x \;=\; \int_{y_{low}}^{y} p(y)dy$

MC method needs large no. of events. Performing integration everytime is computing intesive. One can perform integrations only once and make tables of $y \; vs \; x$. If needed use interpolation methods to get points in-between.

Importance Sampling

Suppose we want to perform the integration:

$$I = \int_a^b f(y)dy$$

Assume $p(y)$ is a pdf whose behavior resembles that of function $f(y)$ in interval $[a, b]$. Normalization condition on $p(y)$ is

$$\int_a^b p(y)dy = 1$$

Rewriting the integral

$$I = \int_a^b f(y)dy = \int_a^b p(y)\frac{f(y)}{p(y)}dy$$

Random numbers are generated from a uniform distribution $p(x)$ with $x \in [0, 1]$. By performing change of variables, we get

$$x(y) = \int_a^y p(y')dy'$$

Inverting $x(y)$, we will get $y(x)$.

Importance Sampling – contd..

With this change of variables,

$$I = \int_a^b p(y)\frac{f(y)}{p(y)}dy = \int_{\tilde{a}}^{\tilde{b}} \frac{f(y(x))}{p(y(x))}dx$$

MC integration of this gives,

$$\int_{\tilde{a}}^{\tilde{b}} \frac{f(y(x))}{p(y(x))}dx = \frac{1}{N}\sum_{i=1}^{N}\frac{f(y(x_i))}{p(y(x_i))}$$

Note the change in integration limits from a and b to $\tilde{a}$ and $\tilde{b}$.

The advantage of this method is, if $p(y)$ follows closely f, the integrand becomes smooth (or close to constatnt) and we can sample over relevant values for the integrand.

$\Longrightarrow$ The accuracy would be much higher.

The variance of $\tilde{f} = f(y(x))/p(y(x))$ is given by,

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\tilde{f}\right)^2 - \left(\frac{1}{N}\sum_{i=1}^{N}\tilde{f}\right)^2$$

Exercise: Apply the method of importance sampling to compute the integral $\int_0^1 e^x\, dx$. Use $p(x) = \frac{2}{3}(1 + x)$ and generate random numbers with this probability distribution in the interval [0,1]. Using these random numbers, evaluate the integral and estimate the gain in efficiency with respect to the Brute-Force Monte Carlo method.

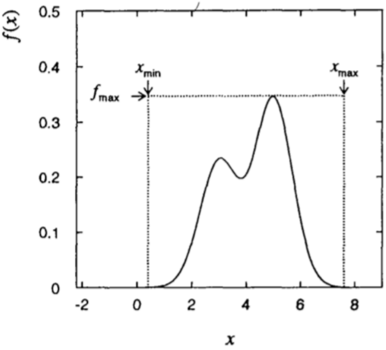
If y is distributed uniformly, by using variable transformation $y = \frac{2}{3}x + \frac{1}{3}x^2 \implies x^2 + 2x - 3y = 0$
$\implies x = -1 + \sqrt{1+3y}$

For x in [0,1] $\implies$ y in [0,1].

Acceptance-Rejection Method

Consider a pdf $f(x)$
with its maximum height f_{max}.
Enclose the pdf in a box

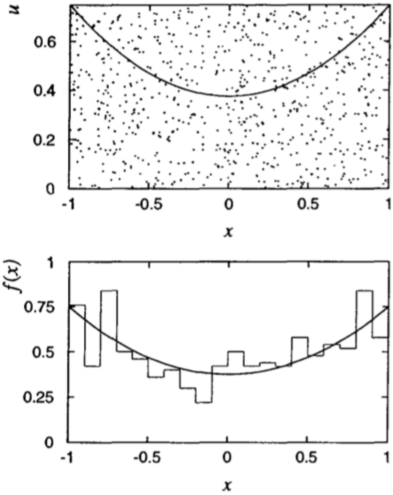
- Generate a uniform random number x, between $[x_{min}, x_{max}]$.
- Generate a 2nd independent uniform random number y, between $[0, f_{max}]$.
- If $y < f(x)$, accept x, otherwise reject x and repeat.



Example of Acceptance-Rejection Method

$$f(x) = \tfrac{3}{8}(1 + x^2)$$

$$(-1 \leq x \leq 1)$$

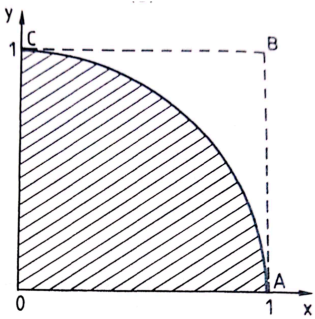


Normalize histogram of accepted$x$ values.

Example of Acceptance-Rejection Method

Example: Compute the value of π

Throw random numbers in to a unit square $x \in [0, 1]$, $y \in [0, 1]$. Compare the areas of the unit square and the quarter of the unit circle $(x^2 + y^2 < 1)$ centred around the region. The ratio $= \pi/4$.



Application of MC method in Particle Physics

Examples:

- Generate pdf $g(a)$ of a function $a(x_1, ...x_n)$ of random variables x_i that are distributed with pdfs $f_i(x_i)$. e.g. to estimate some properties of $g(a)$.
- MC event generator:
  e.g., Consider proton-proton scattering, $pp \rightarrow X$. The theory predicts the probability for a particular event to occur as a function of certain variables (say scattering angle, momenta of output particles etc..). MC program is constructed to generate values of those variables of the output particles.
  e.g. **PYTHIA, HERWIG** etc..
- Detector simulation program: Takes as input the momentum vectors of the generated particles. The response of a detector to the passage of the particles involves random processes such as production of ionization, multiple Coulomb scattering, electromagetic and hadronic showers, ...
  Programming package: **GEANT**

Markov Chain Monte Carlo

Markov Chains

A Markov chain is a stochastic process where given the present state, past and future states are independent.

- Consider a sequence of random variables $\{X_t\}$, $t = 0, 1, ...$
- X_t may have a finite or countably infinite number of possible values, called *states*. The *state space*, S, is the set of all possible values of X_t.
- The notation $X_t \ = \ j$ indicates that the process is in state j at instant t (consider descrete time steps).
- Let the probability for a state i at time t to change to a state j at time $t+1$ is $p_{ij}^{(t)}$.
- The sequence $\{X_t\}$, $t = 0, 1, ...$ is a *Markov chain* if

$$P(X_{t+1} = j | X_t = i, ...X_0 = x_0) \ = \ P(X_{t+1} = j | X_t = i)$$

i.e. the future, given past and present, only depends on the present.

contd..

- From the previous relation one can say that the probabilistic properties of the chain are completely determined by:
  - initial distribution for X_0, and
  - the transition distribution $p_{ij}^{(t)}$.

  If the transition probabilities do not depend on t, $p_{ij}^{(t)} = p_{ij}$, the Markov chain is called *homogeneous*.
- A Markov chain is governed by a *transition probability matrix*.
  - Assuming n states, all integer valued, the transition matrix $\mathbf{P}$ is a $n \times n$ matrix with elements p_{ij}.
  - $0 \leq p_{ij} \leq 1$ and $\sum_j p_{ij} = 1$.

Example: Random walk

Consider a random walker in one dimension. The probability for moving left or right on the line from its current position is governed by a probability function f and x_n represents its current position at instant n, $n \in N$. Suppose the initial position x_0 is distributed according to some distribution. The positions can be related as

$$x_n = x_{n-1} + w_n = w_1 + w_2 + ... + w_n$$

where w_i are independent random variables with probability function f. So, $\{x_n : n \in N\}$ **is a Markov chain in** Z.
The position of the chain at instant n is described probabilistically by the distribution of $w_1 + w_2 + ... + w_n$.
If $f(+1) = p$, $f(-1) = q$ and $f(0) = r$, with $p + q + r = 1$, the transition probabilities are

$$P(x_n, x_{n+1}) = \begin{cases} p \ , if \ x_{n+1} = x_n + 1 \\ q \ , if \ x_{n+1} = x_n - 1 \\ r \ , if \ x_{n+1} = x_n \\ 0 \ , otherwise \end{cases}$$

Definitions

- A state is called a **recurrent state** if, the chain eventually returns to the same state with probability 1. If the expected time for return is finite, the state is called **nonnull**. For finite state spaces, the recurrent states are nonnull.
- A Markov chain is **irreducible** if, starting from a state i any other state j can be reached in a finite number of steps, for all i and j. i.e. for each i and j there must exist $m > 0$ such that $P[X_{m+n} = j | X_n = i] > 0$.
- A Markov chain is **periodic** if it can visit certain portions of the state space only at certain regularly spaced intervals. e.g. a state j has period d if the prob. of going from i to j in n steps is 0 for all n not divisible by d. If every state in a Markov chain has period 1, the chain is called **aperiodic**.
- An irreducible, aperiodic Markov chain with all its states nonnull and recurrent is called **ergodic**.

Limiting theory of Markov chains

- Let π is a vector of probabilities, $\pi_i(\text{marginal probability}) = P[X_t = i]$, and $\sum \pi_i = 1$.
- Then the marginal distribution of X_{t+1} must be $\pi^{\mathbf{T}}\mathbf{P}$.
- $\pi^{\mathbf{T}}\mathbf{P} = \pi^{\mathbf{T}}$ $(or\ \sum_i p_{ij}\pi_i = \pi_j) \implies \pi$ is **stationary** distribution for the Markov chain having transition probability matrix $\mathbf{P}$.
- If X_t follows a stationary distribution, then the marginal distributions of X_t and X_{t+1} are identical.
- If $\pi_i p_{ij} = \pi_j p_{ji}$ for a time-homogeneous Markov chain, for all $i, j \in S$, π is a stationary distribution for the chain. And the chain is called **reversible**, because the joint distribution of a sequence of observations is the same whether the chain is run forwards or backwards → Condition of **Detailed Balance**.

contd..

- In addition, if the Markov chain is *ergodic*, then

$$\lim_{n\to\infty} P[X_{t+n} = j | X_t = i] = \pi_j$$

- Extending further, if $X_1, X_2, ...$ are realizations from an irreducible and aperiodic Markov chain with stationary distribution π, then X_n converges in distribution to the distribution given by π.
- And, for any function f, and if f is integrable

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} f\left(X_t\right) \to \int f(x)\pi(x)dx$$

This is another form of the *ergodic theorem*

Motivation for Markov Chain Monte Carlo

- In application of Monte Carlo methods, we want to generate a sample of random variables with a target distribution $f(\mathbf{x})$.
- The methods discussed before may be able to provide an approximate or exact sample.
- The strategy of MCMC sampling is to construct an *irreducible, aperiodic Markov chain* for which the *stationary distribution* equals the target distribution f.
- Asymptotically, the sample will resemble that of f.
- MCMC methods are easy to customize for very diverse and difficult problems and, also, increasing dimensionality usually does not slow convergence or make implementation more complex.
- A wide variety of algorithms have been proposed for the construction of a suitable chain.

Metropolis-Hastings algorithm

- Suppose $f(\mathbf{x})$ is our target distribution
- Generate an initial value X_0, drawn at random from some starting distribution g, with $f(X_0) > 0$.
- Given X_0, the algorithm produces a sequence of random variables, as follows:
  - Sample a candidate value from a *proposal distribution*, $X^*_{t+1} \; \sim \; g(X|X_t)$
  - Compute the Metropolis-Hastings ratio $R(X_t, X^*)$,

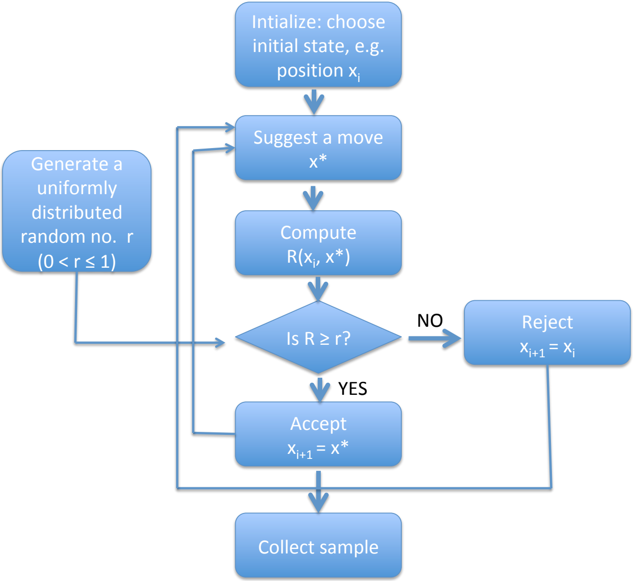
    $$R(X_t, X^*) \; = \frac{f(X^*_{t+1})}{f(X_t)} \frac{g(X_t|X^*_{t+1})}{g(X^*_{t+1}|X_t)}$$

    Note: $R(X_t, X^*)$ is always defined, since the proposal X^*_{t+1} can only occur if $f(X_t) > 0$ and $g(X^*_{t+1}|X_t) > 0$.
  - Sample a value for X_{t+1} according to

    $$X_{t+1} \; = \begin{cases} X^* & \text{with probability } \alpha(X_{t+1}|X_t) = min\{R(X_t, X^*), 1 \\ X_t & \text{otherwise} \end{cases}$$

  - Repeat the procedure for each t

Metropolis-Hastings algorithm



MH algorithm, contd..

- When the proposal distribution is symmetric, $g(X^*|X_t) = g(X_t|X^*)$, the method is known as the **Metropolis algorithm**.
- A chain constructed via the MH algorithm is *Markov* as X_{t+1} depends only on X_t.
- Irreducible and aperiodic properties depends on the choice of proposal distribution.
  - If these properties are satisfied the sequence will converge to a stationary distribution.
  - Consequently, we can use the sample to compute useful quantities, e.g. mean can be approximated using sample averages.
- So, provided the simulation chain is run for *long enough time* it should get a good approximation of the desired distribution.

MH algorithm, contd..

- Choice of good proposal distributions can greatly enhance the performance of the MH algorithm
  - A well-chosen one can converge to stationary distribution in a reasonable number of iterations
  - Produces candidate values that are not accepted or rejected too frequently
- Both of these factors are related to the spread of the proposal distribution
  - If g is too diffuse relative to f, the candidate values will be rejected frequently $\implies$ more iterations
  - On the other hand, if it is too focused (small variance) the chain will remain in one small region of the target distribution for many iterations $\implies$ more iterations to cover whole region
- Two general strategies to chose proposal distribution

MH algorithm, contd..

- Independence chains: Consider the proposal distribution $g(X^*|X_t) = g(X^*)$ for some fixed density g.
  Each candidate value is drawn indepnedently of the past
  The Metropolis-Hastings ratio becomes

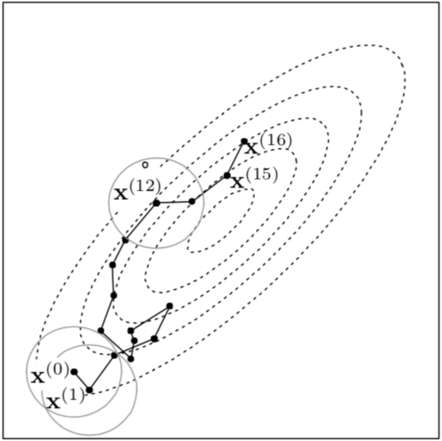
  $$R(X_t, X^*) \; = \frac{f(X^*_{t+1})}{f(X_t)} \frac{g(X_t)}{g(X^*)}$$

  The resulting Markov chain is irreducible and aperiodic if $g(X) > 0$ whenever $f(X) > 0$.
  Note: R can be expressed as $R = w^*/w_t$, where $w^* = f(X^*)/g(X^*)$ and $w_t = f(X_t)/g(X_t)$. If w_t is much larger than w^*, the chain will get stuck at the current value for long periods.

MH algorithm, contd..

- Random Walk Chains: $g(X^*|X_t) = h(X^* - X_t)$, for a symmetric distribution with pdf h.
  - Common choices for h: uniform distribution, scaled standard normal distribution, and scaled Student's t distribution
  - Example:

Hypothetical random walk chain for sampling a 2d target distribution using proposed increments sampled uniformly from a disk centered at the current value.

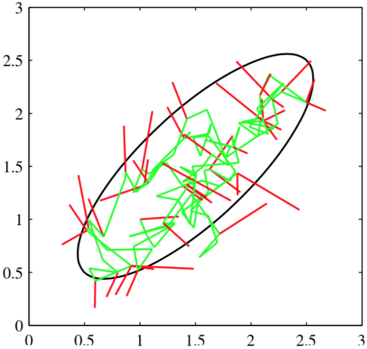


(Ref: Fig. 7.4, Givens & Hoeting)

MH algorithm, contd..

Example: Sampling from a two-dimensional Gaussian distribution using the Metropolis algorithm in which the proposal distribution is an isotropic Gaussian.

A simple illustration using Metropolis algorithm to sample from a Gaussian distribution whose one standard-deviation contour is shown by the ellipse. The proposal distribution is an isotropic Gaussian distribution whose standard deviation is 0.2. Steps that are accepted are shown as green lines, and rejected steps are shown in red. A total of 150 candidate samples are generated, of which 43 are rejected.

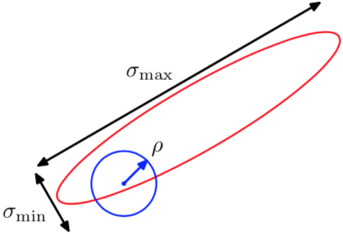


(Ref: Fig. 11.9, Bishop - Pattern Recognition and Machine Learning)

MH algorithm, contd..

Example: scale of the proposal distribution

Schematic illustration of the use of an isotropic Gaussian proposal distribution (blue circle) to sample from a correlated multivariate Gaussian distribution (red ellipse) having very different standard deviations in different directions, using the Metropolis-Hastings algorithm. In order to keep the rejection rate low, the scale ρ of the proposal distribution should be on the order of the smallest standard deviation $\sigma_{\min}$, which leads to random walk behaviour in which the number of steps separating states that are approximately independent is of order $(\sigma_{\max}/\sigma_{\min})^2$ where $\sigma_{\max}$ is the largest standard deviation.



(Ref: Fig. 11.10, Bishop - Pattern Recognition and Machine Learning)

Gibbs Sampling

- Consider multidimensional target distributions
- Metropolis-Hastings sampling can be applied, but there are challenges in constructing proposal distributions for multidimensions
- Goal is to construct a Markov chain whose stationary distribution approximates target distribution
- Idea is to sample one dimension at a time
- Gibbs sampler does this by sequentially sampling from univariate conditional distributions, which are often available in closed form

Gibbs Sampling, contd..

- Consider a trivariate target $f(x) = f(x_1, x_2, x_3)$
- Suppose, we are able to write down the conditional pdfs

  $$f(x_1|x_2, x_3), \quad f(x_2|x_1, x_3), \quad f(x_3|x_1, x_2)$$

  and these can be sampled from
- Gibbs sampler proceeds by generating a sequence $\{X^{(t)}\}$ iteratively, by sampling from conditionals

  $$\begin{aligned} X_1^{(t+1)} &\sim f(x_1|X_2^{(t)}, X_3^{(t)}) \\ X_2^{(t+1)} &\sim f(x_2|X_1^{(t+1)}, X_3^{(t)}) \\ X_3^{(t+1)} &\sim f(x_3|X_1^{(t+1)}, X_2^{(t+1)}) \end{aligned}$$

Gibbs Sampling, contd..

- Gibbs sampling is a simple and widely applicable Markov chain Monte Carlo algorithm
- It can be seen as a special case of the Metropolis-Hastings algorithm:
  - Consider Gibbs sampler as a sequence that updates one component of X at a tme
  - The acceptance prob. is 1, that is why Gibbs sampler has no accept/reject step
- Since it is a special case of MH, the convergence of MH also applies to Gibbs

A Simple Example

Consider the distribution

$$f(x,y) = \frac{n!}{x!(n-x)!} y^{(x+\alpha-1)} (1-y)^{(n-x+\beta-1)}, \;\; x \in 0, ..., n, \;\; y \in [0,1]$$

Since it is hard to simulate directly from this $p(x,y)$ it can be easier to work with the conditional distributions.

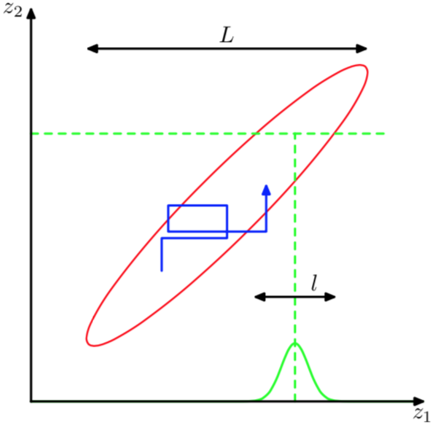
- $f(x|y) = f(x,y)/f_y(y) = f(x,y)/\int f(x,y)dx$
  $= f(x,y)/\left(y^{\alpha-1}(1-y)^{\beta-1}\right)$
  $\implies f(x|y) \sim Binomial(n,y)$
- $f(y|x) = f(x,y)/f_x(x) = f(x,y)/\int f(x,y)dy$
  $= f(x,y)/\frac{n!}{x!(n-x)!} B(x+\alpha, n-x+\beta) = \frac{y^{(x+\alpha-1)}(1-y)^{(n-x+\beta-1)}}{B(x+\alpha, n-x+\beta)}$
  $\implies f(y|x) \sim Beta(x+\alpha, n-x+\beta)$

Thus, it can be easy to use Gibbs sampler to simulate from the joint distributions

Difficulties with Gibbs sampling

In case of strongly correlated samples it might take long time to reach the stationary distribution

Illustration of Gibbs sampling by alternate updates of two variables whose distribution is a correlated Gaussian. The step size is governed by the standard deviation of the conditional distribution (green curve), and is $O(l)$, leading to slow progress in the direction of elongation of the joint distribution (red ellipse). The number of steps needed to obtain an independent sample from the distribution is $O((L/l)^2)$.



(Ref: Fig. 11.11, Bishop - Pattern Recognition and Machine Learning)

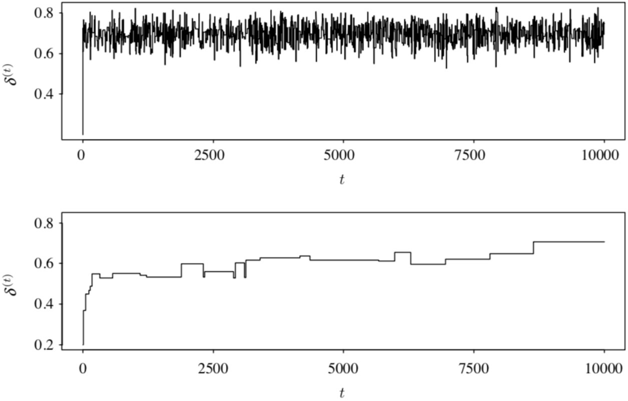
MCMC Convergence Diagnostics

- It is important to check the convergence and mixing properties of the chain before using it for any estimation
- Various methods to check whether the chain reached the stationarity distribution
- One way is to perform simple graphical diagnostics
  - *Sample path* (trace or history) plot: Plot of X_t vs t
    - If poor mixing: will remain at or near the same value for many iterations
    - Well mixing: quickly moves away from its starting value and wiggle about the region supported by f
  - *Autocorrelation plot*
    - Autocorrelation at lag ℓ is the correlation between iterates that are ℓ iterations apart

$$c_\ell = \frac{\langle A_{n+\ell} A_n \rangle - \langle A_n \rangle^2}{\langle A_n^2 \rangle - \langle A_n \rangle^2}$$

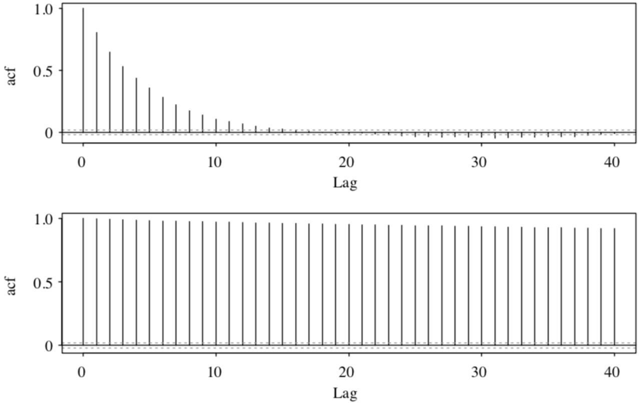
    - If poor mixing: exhibits slow decay of the autocorrelation as the lag between iterations increases

Example of sample path



(Ref: Fig. 7.2, Givens & Hoeting)

Example of Autocorrelation



(Ref: Fig. 7.8, Givens & Hoeting)

Summary

- MCMC is a powerful tool to solve many problems that are difficult using other numerical techniques
- This lecture was to just introduce you to some basic concepts of MCMC
- There are many packages readily available to perform MCMC simulation, implemening different sampling methods
- However, it is important to understand them before applying blindly to your problems
- It is also important to check the convergence of the sample before using it for any estimation or inference

END