Lab Session-5: Introduction to Monte Carlo Methods

MATH350 – Statistical Inference

Dr. Tanujit Chakraborty
Assistant Professor in Statistics at Sorbonne University
tanujit.chakraborty@sorbonne.ae

Course Webpage: https://www.ctanujit.org/SI.html

R Code: https://github.com/tanujit123/MATH350

Let's start

Topics for Today's Session

- Monte Carlo: Introduction
- Monte Carlo: A Brief History
- Motivating Examples & Monte Carlo Solutions
- Computing Expectations
- Monte Carlo Algorithm
- Statistical Properties
- What's next?
- References

Monte Carlo Simulation: Introduction

- A broad class of numerical algorithm depends on repeated random sampling. If it is not possible to obtain the exact analytical solution often Monte Carlo method can be used to provide a very good approximate solution.

- Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational tools for the simulation of random variables and the approximation of integrals/expectations.

- Monte Carlo methods, are used in many fields including statistical physics, statistical inference, genetics, finance, etc.

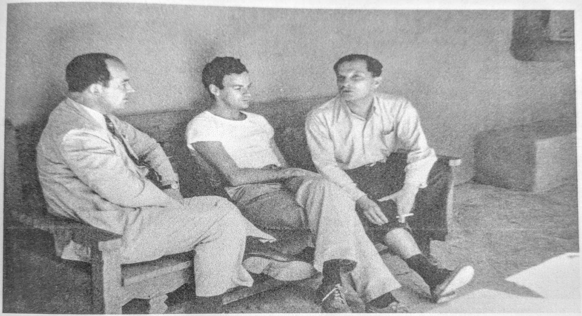
- The idea is use the computer to simulate or mimic the premises of the question and based on what the computer returns, guess the answer.

- A technique of numerical analysis that uses random sampling to simulate the real-world phenomena.

Monte Carlo Methods: Introduction

- Two essential classes of problems in the arena of Statistical Inference: Optimization & Integration Problems.
- It is sometimes infeasible to analytically compute the estimators using standard inferential methods: Maximum Likelihood Estimation (MLE), Method of Moments (MOM), Bayesian Estimation.
- Therefore, we need to rely on numerical solutions to solve statistical inferential problems.
- To provide a general solution, we need to simulate, of either the true or substitute distributions, to calculate the quantities of interest.
- In decision theory (classical or Bayesian), this solution is natural since risks and Bayes estimators involve integrals with respect to probability models.

People behind it



John von Neumann, Richard Feynman, and Stanislaw Ulam, at the lodge at Bandelier National Monument (near Los Alamos), 1949. "We used to go for walks . . . in the canyons . . . and von Neumann gave me an interesting idea: that you don't have to be responsible for the world that you're in," says Feynman. "So I have developed a very powerful sense of social irresponsibility as a result of Von Neumann's advice." *(Photograph by Nicholas Metropolis; courtesy of Claire and Françoise Ulam)*

Reference: `https://www.ias.edu/ideas/adventures-mathematician`

Monte Carlo Method: A brief history

- Manhattan project (1940's): Simulations during the initial development of thermonuclear weapons. During this, Monte Carlo method was invented by Stanislaw Ulam and John von Neumann.

- John von Neumann (may be together with his wife Klára Dán von Neumann) first wrote the computer code to perform Monte Carlo simulations.

- Being secret, the work of von Neumann and Ulam required a code name. Nicholas Metropolis suggested using the name Monte Carlo which refers to the Monte Carlo Casino in Monaco where Ulam's uncle would borrow money from relatives to gamble.

- The Metropolis algorithm (first actual Monte Carlo calculations) was named the top algorithm of the 20th century by a committee of mathematicians, computer scientists & physicists. Metropolis (1948) and Berger (1963) contributed significantly.

- Exponential growth since the 1980's with the availability of digital computers.

Monte Carlo Method: Usage

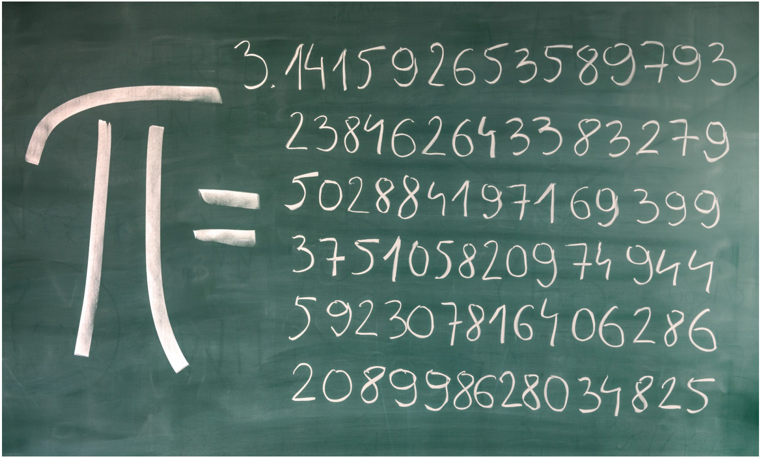
1. Calculating the area below a curve.
2. Calculating multidimensional integration.
3. Solving optimization problems.
4. Analyzing any complicated stochastic system (model).
5. Applications: With the dramatic increase in computational power, Monte Carlo methods are increasingly used in Particle physics, Quantum field theory, Traffic flow simulations, weather sciences, Financial market simulations, etc.

Monte Carlo Method: Applications

- Engineering, biology, climate science, neuroscience, ecology, finance, etc.
- Essentially, there are two main places where Monte Carlo is used

    1. If a complicated model determines the quantity, like climate models, bridge breaking points, stock markets etc.
    2. If it is not possible to analytically determine the quantity of interest. In statistics, this is where it is often used.

Motivating Example - 1 : Estimating π



Computing π (Mathematical Constant)

There exists various ways of Finding π

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}}}}}$$

$$\frac{1}{\pi} = \frac{12}{640320^{3/2}} \sum_{k=0}^{\infty} \frac{(6k)!(13591409 + 545140134k)}{(3k)!(k!)^3 (-640320)^{3k}}$$

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdots$$

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{k!^4 \left(396^{4k}\right)}.$$

$$\frac{\pi}{2} = \left(\frac{2}{1}\cdot\frac{2}{3}\right)\cdot\left(\frac{4}{3}\cdot\frac{4}{5}\right)\cdot\left(\frac{6}{5}\cdot\frac{6}{7}\right)\cdot\left(\frac{8}{7}\cdot\frac{8}{9}\right)\cdots$$

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots$$

Computing π : Let's play Dart game

Throwing darts (randomly) at a target and counting the fraction of darts that falls inside the circle:

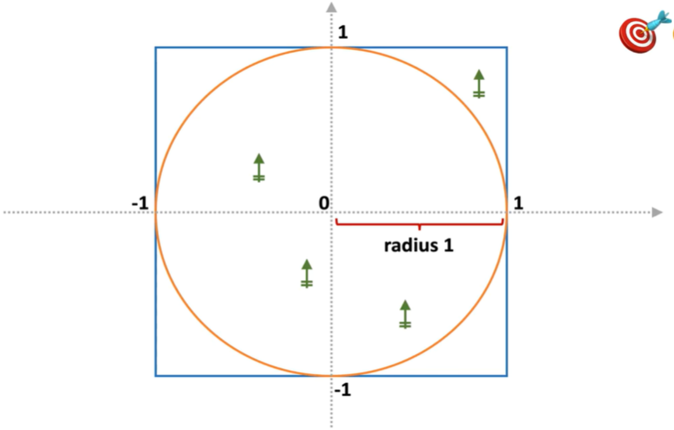
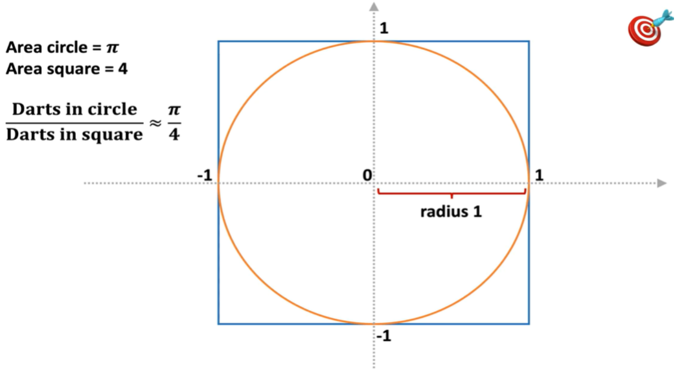


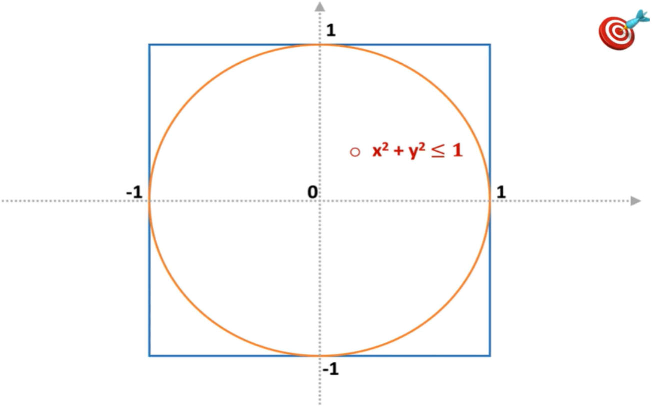
Image Credits: https://www.youtube.com/c/codeRtime

Computing π : Let's play Dart game

We can simulate this process using a simple R program and try estimating π:



Points inside the circle satisfies the following:



Consider the 2×2 square, say $S \subseteq \mathbb{R}^2$ with inscribed disk D of radius 1.

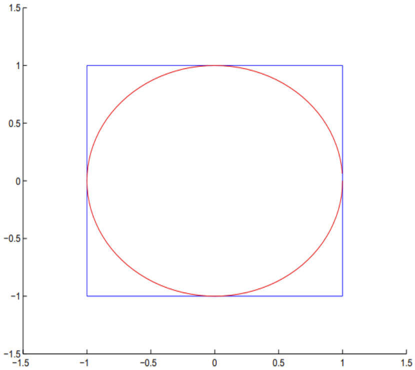


Figure: A 2×2 square $S \subseteq \mathbb{R}^2$ with inscribed disk D of radius 1.

Computing π with Monte Carlo Methods

- We have
$$\frac{\int\int_D dx_1dx_2}{\int\int_S dx_1dx_2} = \frac{\pi}{4}.$$
- How could you estimate this quantity through simulation?

$$\frac{\int\int_D dx_1dx_2}{\int\int_S dx_1dx_2} = \int\int_S \mathbb{I}\left((x_1,x_2)\in D\right)\frac{1}{4}dx_1dx_2 = \mathbb{E}[\phi(X_1,X_2)] = \theta$$

where the expectation is w.r.t. the uniform distribution on S and

$$\phi(X_1,X_2) = \mathbb{I}\left((x_1,x_2)\in D\right).$$

- To sample uniformly on $S = (-1,1)\times(-1,1)$, we simply use `runif` function in R.

Computing π: Monte Carlo Solutions

We simulate 1000 dart throws and obtain the approximate value of π. runif is a r-function to generate uniform random variables within a specified range [(-1, 1) here].

```
num_darts <- 1000
num_darts_in_circle <- 0 # We have not thrown any darts now
x <- runif(n=1,min=-1,max=1) # random number between -1 and 1
y <- runif(n=1,min=-1,max=1)
?runif # R Help
# We need Thousand Throws
for(i in 1: num_darts){
  x <- runif(n=1,min=-1,max=1)
  y <- runif(n=1,min=-1,max=1)
  # 1000 values are assigned
  if(x^2+y^2 <= 1){ # condition for the dart inside the circle
    num_darts_in_circle <- num_darts_in_circle + 1
    # counting that one dart inside the circle
  }
}
print(4 * num_darts_in_circle / num_darts)
# Values will change in every time we run it. Can be stable if we set num_dart very high.
# Let us try to optimize the code.
```

Computing π: Code

Here, we write the code in a different manner. We simulate 100000 dart throws and try to write the code efficiently so that the running time is optimized.

```
### Fast Implementation and Optimized Code ###
num_darts <- 100000
num_darts_in_circle <- 0 # We have not thrown any darts now
x <- runif(n=num_darts, min=-1, max=1)
y <- runif(n=num_darts, min=-1, max=1)
sum_square <- x^2 + y^2
# x, y, sum_square are vectors of length 100000
indexes_darts_in_circle <- which(sum_square <= 1) # counter
num_darts_in_circle <- length(indexes_darts_in_circle)
# check ?length and ?which
print(4 * num_darts_in_circle / num_darts)

### Visualization darts ###
plot(x,y)
```

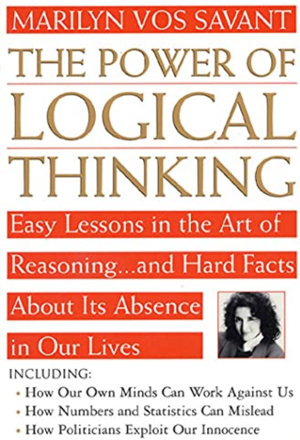
All the Codes are made available at https://github.com/tanujit123/MATH350.

Motivating Eg. - 2 : Monty Hall Problem

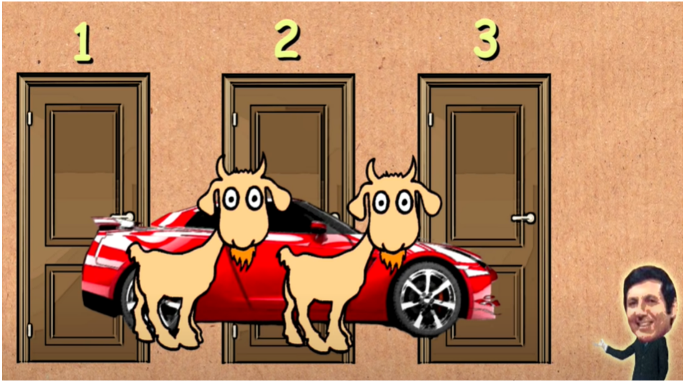
Let's Make a Deal Show (1960s - 1990s):

- You are on a game show, being asked to choose between three doors. One door has a car, and the other two have goats.
- Once you choose a door, Monty Hall (host) opens one of the other doors, which he knows has a goat behind it.
- Monty then asks whether you would like to switch your choice of door to the other remaining door.

Do you choose to switch or not?



Monty Hall Problem



References:
1. https://brilliant.org/wiki/monty-hall-problem/
2. Monte Carlo Simulation Talk by Dootika Vats:
https://www.isibang.ac.in/~athreya/Teaching/statistics1/dootika.pdf

Monty Hall Problem: Switch or Not?

- Of course, whether you will switch or not depends on which action has the largest probability of winning the car.
- The answer is : You should always switch!

  $\Pr$ (Winning if you don't switch) $= \frac{1}{3}$
  $\Pr$(Winning if you switch) $= \frac{2}{3}$

- If this phenomenon is hard to believe, you're not alone.

> *"even Nobel physicists systematically give the wrong answer, and that they insist on it, and they are ready to berate in print those who propose the right answer" – Vos Savant (1997)*

- Let's simulate this situation on a computer to repeat a Monty Hall experiment multiple times. And in each time, we will see whether switching or not switching would be more beneficial.

Monty Hall Problem: Switch or Not?

Now, if you switch are you guaranteed to win? Absolutely not. But if you play this game over and over again, you will win 2/3 of the time. So, switch is a better strategy.

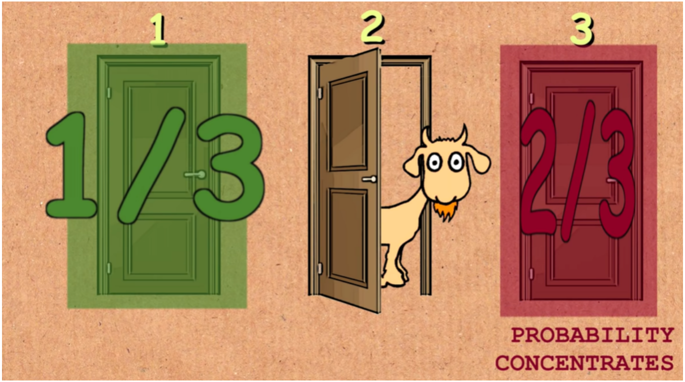


Image Credits: https://www.youtube.com/c/numberphile

Solving Monty Hall Problem: Code

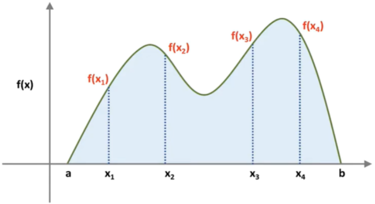
```r
set.seed (1) # To obtain reproducible results
repeats <- 1e4 # We will repeat the experiment 10000 times
win.no.switch <- numeric(length = repeats) # will save 0 / 1 based on winning no switch
win.switch <- numeric(length = repeats)   # will save 0 / 1 based on winning switch

for(r in 1:repeats) # Repeat process many times
{
 # The setup
 doors <- 1:3    # three doors
 prize <- sample(1:3, 1)  # randomly select the door which has the prize
 # Contestants are ready. Game starts
 chosen.door <- sample(1:3, 1)    # choose a door

 # reveal a door that is not the chosen door and not the door with a prize in it
 which.reveal <- rep((1:3)[-c(prize, chosen.door)], 2) # doing rep(., 2) because sample() is being annoying
 reveal <- sample(which.reveal, size = 1) # randomly choose which door to reveal
 win.no.switch[r] <- chosen.door == prize  #tracking win if don't change door
 chosen.door <- (1:3)[-c(reveal, chosen.door)] #change door
 win.switch[r] <- chosen.door == prize # tracking win if change door
}

### Monty Hall Problem Outputs ###

head(win.no.switch) # first few results for no switching
head(win.switch) # first few results of switching
mean(win.no.switch) #Prob of winning if you don't switch
mean(win.switch) # Prob of winning if you switch
```

Motivating Eg. - 3 : Monte Carlo Integration

Solving Definite Integrals:

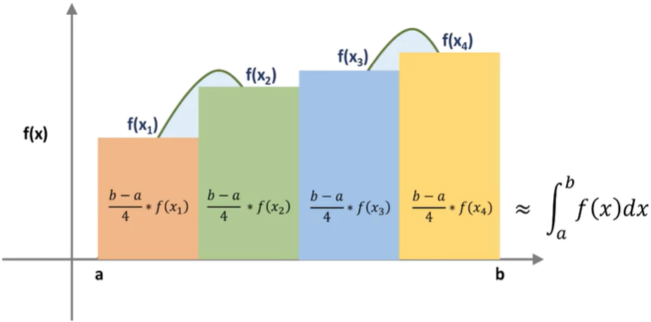
- Using Monte Carlo Integration, we can estimate single and multiple definite integrals.
- Imagine we have a function defined on (a, b) and you want to find the area under the curve (plot on the right).
- We can uniformly sample many random points and then compute the value of the function at each points.

An example with four point...



$$\int_a^b f(x)dx \approx (b-a) * \frac{f(x_1)+f(x_2)+f(x_3)+f(x_4)}{4}$$

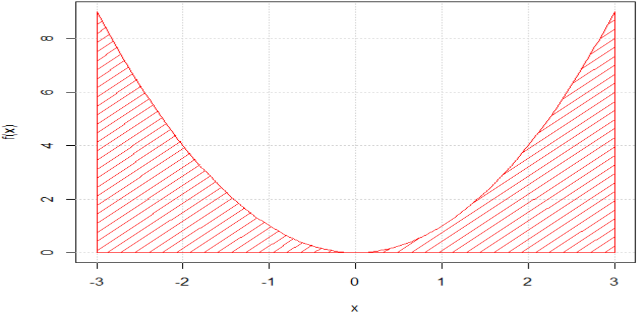
Solving Definite Integrals



Note:
1. This is a crude estimate of the actual function. Not perfect anyway.
2. To make the estimate better, we can increase the random points from 4 to a few thousands.

Solving Definite Integrals

- Solve $\int_{-3}^{3} x^2 dx$ by MC Simulations.



Code: `https://github.com/tanujit123/MATH350`

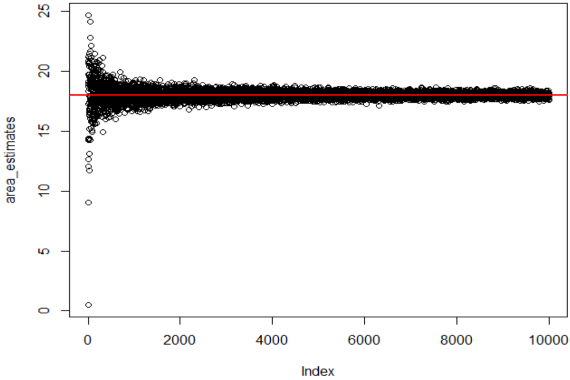
Solving Definite Integrals

A Simple Monte Carlo Solution to the problem using the approximation formula presented earlier.

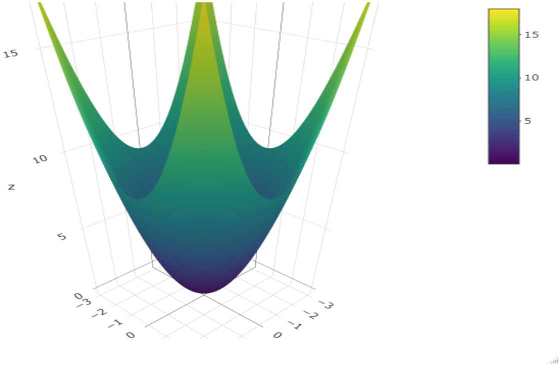
```
### Single Integral Problem ###

f <- function(x)x^2
curve(f, from = -3, to = 3) # plot the function
install.packages("DescTools")
library(DescTools)
Shade(f, breaks = c(-3,3), col = "red") # add background grid to the plot
grid()

# Estimate the shaded region by Monte Carlo Simulation
# 2000 random samples are generated uniformly between (-3, 3)

query_points <- runif(n = 2000, min=-3, max=3)
area_shaded_region <- 6 * mean(f(query_points))
```

Observation: Answer is close to the actual value 18

If we increase the number of points, we may get better estimates. Below we plot in the horizontal axis the number of random points used in the simulation and the vertical axis shows the values of the estimates. Red line defines the actual value. Accuracy of our estimates improves once we include more points and stabilizes after certain numbers.



- Solve $\int_{-3}^{3}\int_{-3}^{3}(x^2 + y^2)dxdy$ by MC Simulations.



Code: `https://github.com/tanujit123/MATH350`

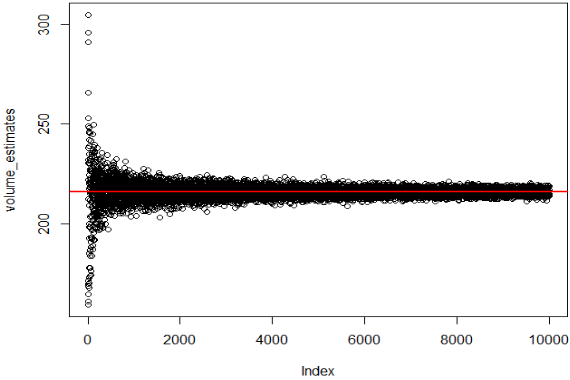
Our Goal: Estimate the volume under the surface of the last plot.

```
### Double Integral Problem ###

g <- function(x,y)x^2 + y^2
x <- y <- seq(-3, 3, length = 100) # generate 100 (x,y) values
z <- outer(x, y, g)
persp(x, y, z)
install.packages("plotly")
library(plotly)
plot_ly(x = x, y = y, z = z) %>% add_surface()

# 2000 random samples are generated uniformly between (-3, 3)

query_points_x <- runif(n = 1000, min=-3, max=3)
query_points_y <- runif(n = 1000, min=-3, max=3)
volume_shaded_region <- 36 * mean(g(query_points_x, query_points_y))
```

Observation: Answer is close to the actual value 216 (elementary calculus).

Solving Double Definite Integrals

If we increase the number of points, we may get better estimates. Below we plot in the horizontal axis the number of random points used in the simulation and the vertical axis shows the values of the estimates. Red line defines the actual value. Accuracy of our estimates improves once we include more points and stabilizes after certain numbers.



Computing Expectations

- Let X be either
  - a discrete random variable (r.v.) taking values in a countable or finite set Ω, with p.m.f. f_X
  - or a continuous r.v. taking values in $\Omega = \mathbb{R}^d$, with p.d.f. f_X
- Assume you are interested in computing

$$\theta = \mathbf{E}\left(\phi\left(X\right)\right) = \begin{cases} \sum_{x\in\Omega} \phi(x) f_X(x) & \text{if } X \text{ is discrete} \\ \int_{\Omega} \phi(x) f_X(x)\, dx & \text{if } X \text{ is continuous} \end{cases}$$

  where $\phi : \Omega \to \mathbb{R}$
- It is impossible to compute θ exactly in most realistic applications.
- Even if it is possible (for Ω finite) the number of elements may be so huge that it is practically impossible
- Example: $\Omega = \mathbb{R}^d,\ X \sim N(\mu, \Sigma)$ and $\phi(x) = \mathbb{I}\left(\sum_{k=1}^{d} x_k^2 \geq \alpha\right)$.
- Example: $\Omega = \mathbb{R}^d,\ X \sim N(\mu, \Sigma)$ and $\phi(x) = \mathbb{I}\left(x_1 < 0, \ldots, x_d < 0\right)$.

Example: Queuing Systems

- Customers arrive at a shop and queue to be served. Their requests require varying amounts of time.
- The manager cares about customer satisfaction and not excessively exceeding the 9 am-5 pm working day of his employees.
- Mathematically we could set up stochastic models for the arrival process of customers and for the service time based on past experience.
- **Question**: If the shop assistants continue to deal with all customers in the shop at 5 pm, what is the probability that they will have served all the customers by 5.30 pm?
- If we call $X \in \mathbb{N}$ the number of customers in the shop at 5.30pm then the probability of interest is

$$\mathbb{P}(X = 0) = \mathbb{E}(\mathbb{I}(X = 0)).$$

- For realistic models, we typically do not know analytically the distribution of X.

Example: Particle in a Random Medium

- A particle $(X_t)_{t=1,2,\dots}$ evolves according to a stochastic model on $\Omega = \mathbb{R}^d$.
- At each time step t, it is absorbed with probability $1 - G(X_t)$ where $G : \Omega \to [0, 1]$.
- **Question**: What is the probability that the particle has not yet been absorbed at time T?
- The probability of interest is

$$\mathbb{P}\left(\text{not absorbed at time } T\right) = \mathbb{E}[G(X_1)G(X_2)\dots G(X_T)].$$

- For realistic models, we cannot compute this probability.

Example: Ising Model

- The Ising model serves to model the behavior of a magnet and is the best-known/most researched model in statistical physics.
- The magnetism of material is modeled by the collective contribution of dipole moments of many atomic spins.
- Consider a simple 2D - Ising model on a finite lattice $\mathcal{G} = \{1, 2, ..., m\} \times \{1, 2, ..., m\}$ where each site $\sigma = (i, j)$ hosts a particle with a $+1$ or -1 spin modeled as a r.v. X_σ.
- The distribution of $X = \{X_\sigma\}_{\sigma \in \mathcal{G}}$ on $\{1, 1\}^{m^2}$ is given by

$$\pi(x) = \frac{exp\left(-\beta U(x)\right)}{Z_\beta}$$

where $\beta > 0$ is the inverse temperature and the potential energy is

$$U(x) = -J \sum_{\sigma \sim \sigma'} x_\sigma x_{\sigma'}$$

- Physicists are interested in computing $\mathbb{E}[U(X)]$ and Z_β.

Example: Ising Model

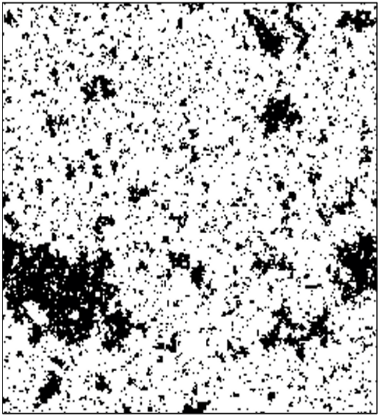


Figure: Sample from an Ising model for $m = 250$.

Example: Statistical Genetics

- At variable sites in the genome of a population, we can represent one chromosome as a haplotype as a vector of binary 0/1s. We humans are diploid so have two copies of each chromosome.
- We often acquire data as "reads", observing those 0/1s along the genome
- We may be interested in trying to determine the haplotypes of an individual given some set of observed sequencing reads where we observe some of the underlying haplotypes, from one of the individuals two haplotypes.
- Let $L_r \in \{1, 2\}$ represent whether a read came from the maternal or paternal haplotype.
- Then we might be interested in $P\left(H_i, H_j | O\right) \propto P\left(O | H_i, H_j\right) = \sum_{L_1, L_2, \ldots} P\left(O | H_i, H_j, L_1, L_2, \ldots\right) P\left(L_1, L_2, \ldots\right)$.
- Naively, for M sequencing reads, this has a computational cost of 2^M, which is unfeasible for a realistic M.
- Monte Carlo methods allow us to estimate $P\left(H_i, H_j | O\right)$ and similar calculations, and are used frequently in genetics.

Bayesian Inference

- Suppose (X, Y) are both continuous r.v. with a joint density $f_{X,Y}(x, y)$.
- Think of Y as data, and X as unknown parameters of interest.
- We have

$$f_{X,Y}(x, y) = f_X(x)\, f_{Y|X}(y|x)$$

  where, in many statistics problems, $f_X(x)$ can be thought of as a prior and $f_{Y|X}(y|x)$ as a likelihood function for a given $Y = y$.
- Using Bayes' rule, we have

$$f_{X|Y}(x|y) = \frac{f_X(x)\, f_{Y|X}(y|x)}{f_Y(y)}.$$

- For most problems of interest, $f_{X|Y}(x|y)$ does not admit an analytic expression and we cannot compute

$$\mathbb{E}\left(\phi(X)|Y = y\right) = \int \phi(x)\, f_{X|Y}(x|y) dx.$$

Monte Carlo Integration

Definition (Monte Carlo method)
Let X be either a discrete r.v. taking values in a countable or finite set Ω, with p.m.f. f_X, or a continuous r.v. taking values in $\Omega = \mathbb{R}^d$, with p.d.f. f_X. Consider

$$\theta = \mathbf{E}\left(\phi\left(X\right)\right) = \begin{cases} \sum_{x\in\Omega} \phi(x) f_X(x) & \text{if } X \text{ is discrete} \\ \int_\Omega \phi(x) f_X(x)\, dx & \text{if } X \text{ is continuous} \end{cases}$$

where $\phi : \Omega \to \mathbb{R}$. Let $X_1, \ldots X_n$ be i.i.d. r.v. with p.d.f. (or p.m.f.) f_X. Then

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i),$$

is called the Monte Carlo estimator of the expectation θ.

Monte Carlo method can be thought of as a stochastic way to approximate integrals.

Pseudocode for Monte Carlo Algorithm

- Simulate independent $X_1, \dots, X_n$ with p.m.f. or p.d.f. f_X.
- Return $\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i)$.

Applications

- *Toy example*: simulate a large number n of independent r.v. $X_i \sim N(\mu, \Sigma)$ and

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left(\sum_{k=1}^{d} X_{k,i}^2 \geq \alpha \right)$$

- *Queuing:* simulate a large number n of days using your stochastic models for the arrival process of customers and for the service time and compute

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left(X_i = 0 \right)$$

  where X_i is the number of customers in the shop at 5.30 pm for i^{th} sample.

- *Particle in Random Medium*: simulate a large number n of particle paths $(X_{1,i}, X_{2,i}, \ldots, X_{T,i})$ where $i = 1, \ldots, n$ and compute

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} G(X_{1,i}) G(X_{2,i}) \ldots G(X_{T,i})$$

Monte Carlo Integration: Properties

- **Proposition**: Assume $\theta = \mathbb{E}(\phi(X))$ exists. Then the Monte Carlo estimator $\hat{\theta}_n$ has the following properties
  - **Unbiasedness**
    $$\mathbb{E}(\hat{\theta}_n) = \theta$$
  - **Strong consistency**
    $$\hat{\theta}_n \to \theta \text{ almost surely as } n \to \infty$$
- *Proof*: We have
  $$\mathbb{E}(\hat{\theta}_n) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(\phi(X_i)) = \theta$$
  Strong consistency is a consequence of the strong law of large numbers applied to $Y_i = \phi(X_i)$ which is applicable as $\theta = \mathbb{E}(\phi(X))$ is assumed to exist.

- **Proposition**: Assume $\theta = \mathbb{E}(\phi(X))$ and $\sigma^2 = \mathbb{V}(\phi(X))$ exist then

$$\mathbb{E}\left(\left(\hat{\theta}_n - \theta\right)^2\right) = \mathbb{V}\left(\hat{\theta}_n\right) = \frac{\sigma^2}{n}$$

and

$$\frac{\sqrt{n}}{\sigma}\left(\hat{\theta}_n - \theta\right) \xrightarrow{\mathrm{d}} N(0,1).$$

- *Proof.* We have $\mathbb{E}\left(\left(\hat{\theta}_n - \theta\right)^2\right) = \mathbb{V}\left(\hat{\theta}_n\right)$ as $\mathbb{E}\left(\hat{\theta}_n\right) = \theta$ and

$$\mathbb{V}\left(\hat{\theta}_n\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}\left(\phi(X_i)\right) = \frac{\sigma^2}{n}.$$

The CLT applied to $Y_i = \phi(X_i)$ tells us that

$$\frac{Y_1 + \ldots + Y_n - n\theta}{\sigma\sqrt{n}} \xrightarrow{\mathrm{d}} N(0,1)$$

so the result follows as $\hat{\theta}_n = \frac{1}{n}(Y_1 + \ldots + Y_n)$.

Monte Carlo Integration: Variance Estimation

- **Proposition**: Assume $\sigma^2 = \mathbb{V}(\phi(X))$ exist then

$$S^2_{\phi(X)} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\phi(X_i) - \hat{\theta}_n \right)^2$$

  is an unbiased sample variance estimator of σ^2.
- *Proof.* Let $Y_i = \phi(X_i)$ then we have

$$\begin{aligned}
\mathbb{E}\left(S^2_{\phi(X)}\right) &= \frac{1}{n-1} \sum_{i=1}^{n} \mathbb{E}\left((Y_i - \bar{Y})^2 \right) \\
&= \frac{1}{n-1} \mathbb{E}\left(\sum_{i=1}^{n} Y_i^2 - n\bar{Y}^2 \right) \\
&= \frac{n\left(\mathbb{V}(Y) + \theta^2\right) - n\left(\mathbb{V}(\bar{Y}) + \theta^2\right)}{n-1} \\
&= \mathbb{V}(Y) \\
&= \mathbb{V}\left(\phi(X)\right)
\end{aligned}$$

  where $Y = \phi(X)$ and $\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i$.

How Good is The Estimator?

- Chebyshev's inequality yields the bound

$$\mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| > c\,\frac{\sigma}{\sqrt{n}}\right) \leq \frac{\mathbb{V}\left(\hat{\theta}_n\right)}{c^2\sigma^2 n} = \frac{1}{c^2}$$

- Another estimate follows from the CLT for large n

$$\frac{\sqrt{n}}{\sigma}\left(\hat{\theta}_n - \theta\right) \overset{\mathrm{d}}{\approx} N(0,1) \Rightarrow \mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| > c\,\frac{\sigma}{\sqrt{n}}\right) \approx 2\left(1 - \Phi(c)\right).$$

- Hence by choosing $c = c_\alpha$ s.t. $2\left(1 - \Phi(c_\alpha)\right) = \alpha$, an approximate $(1-\alpha)100\%$ CI for θ is

$$\left(\hat{\theta}_n \pm c_\alpha\,\frac{\alpha}{\sqrt{n}}\right) \approx \left(\hat{\theta}_n \pm c_\alpha\,\frac{S_{\phi(X)}}{\sqrt{n}}\right)$$

Monte Carlo Integration

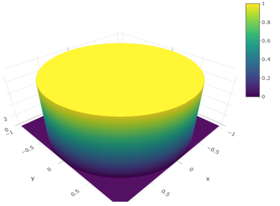
- Whatever being Ω; e.g. $\Omega = \mathbb{R}$ or $\Omega = \mathbb{R}^{1000}$, the error is still in $\sigma/\sqrt{n}$.
- This is in contrast with deterministic methods. The error in a product trapezoidal rule in d dimensions is $O(n^{-2/d})$ for twice continuously differentiable integrands.
- It is sometimes said erroneously that it beats the curse of dimensionality but this is generally not true as σ^2 typically depends on $\dim(\Omega)$.

Practice Problem - 1

- Consider the following function:

$$h(x,y) = \begin{cases} 1, & x^2+y^2 \leq 1 \quad -1 \leq x \leq 1 \\ 0, & x^2+y^2 > 1 \quad -1 \leq y \leq 1 \end{cases}$$

As given, h is a two-parameter function and its graphic is presented below:



Use the integral and the idea of Monte Carlo Simulation to estimate π.

Hints: A simple geometric argument tells us that the volume under this function is equal to π (since we are looking at the cylinder with base surface area π and height 1).

Practice Problem - 2

Consider the following Toy Collector Problem:

Children (and some adults) are frequently enticed to chips packets in an effort to collect toys found in these packets. Assume there are 15 different kinds of toys and each packet contains exactly one with each toy having a probability:

Fig.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Prob.	.2	.1	.1	.1	.1	.1	.05	.05	.05	.05	.02	.02	.02	.02	.02

What is the expected number of packets needed to collect all 15 action figures?

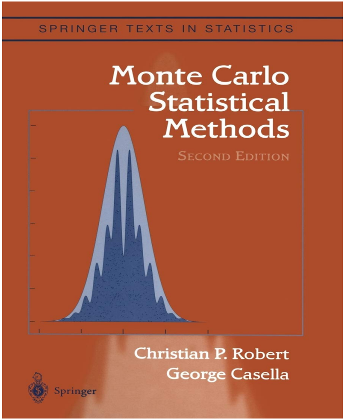
Hints: This question is of great practical importance since it allows us to know what number of chips packets we will need to buy. Surprisingly, the solution to this problem is mathematically quite complicated. A Monte Carlo solution is often the best bet.

Practice Problem - 3

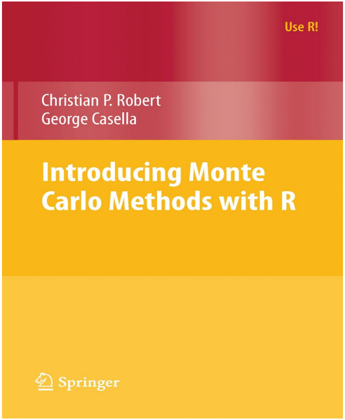
It's my 30th birthday, and my friends bought me a cake with 30 candles on it. I make a wish and try to blow them out. Every time, I blow out a random number of candles between one and the number that remains, including one and that other number. How many times, on average, do I blow before all the candles are extinguished?

Problem of Birthday Candles

Textbooks & Reference



Theoretical Book



Applied Book (Useful for Practicals)