Monte Carlo Integration

Dr. Uma Ravat
PSTAT 194CS

Monte Carlo Methods

Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on **repeated random sampling** to obtain numerical results.

The underlying concept is to use randomness to solve problems that might be deterministic in principle.

Monte Carlo methods are mainly used in three problem classes:

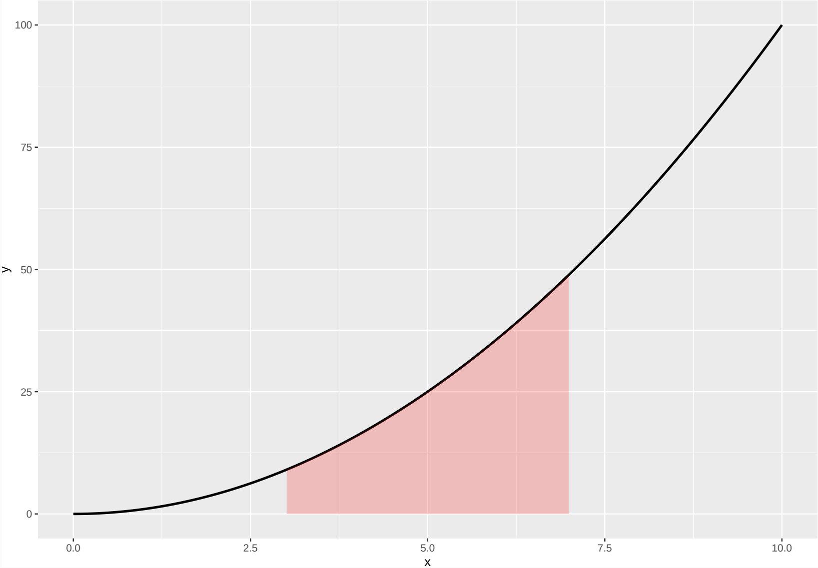
- Numerical integration
- Generating draws from a probability distribution
- Optimization

Steps in MC methods

Monte Carlo methods vary, but tend to follow a particular pattern:

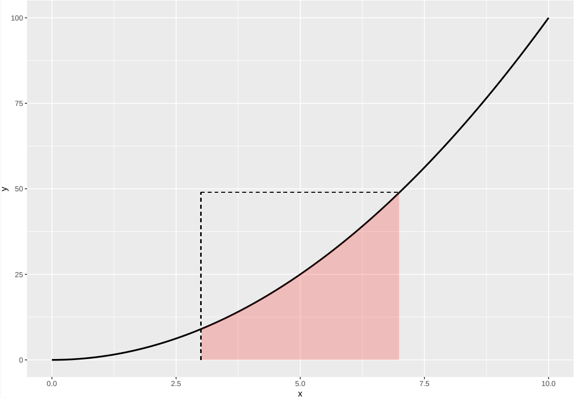
1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
4. Aggregate the results

Numerical integration: Estimate $\int_3^7 x^2 dx$



MC steps:

1. Define a domain of possible inputs: $x \in [3, 7]$ and $y \in [0, f(7)]$
2. Draw points uniformly at random inside the box $[3, 7] \times [0, 49]$
3. Determine how many points below the $f(x)$
4. Compute the ratio*(area of the rectangle) as an estimate of the area under the curve.



R code

1. Generating random points inside the box:

```
numTrials <- 100000
# The function runif() samples from a "uniform" distribution
x <- runif(numTrials, min = 3, max = 7) # Values of x between 3 and 7
y <- runif(numTrials, min = 0, max = 7^2) # Values of y between 0 and 7^2
```

2. Estimating the area under the curve based on simulated observations:

```
# Determine which y points are less than or equal to x^2:
belowCurve <- y <= x^2 # This will create a vector of TRUE and FALSE values
# Compute the ratio of points below the curve:
ratio <- sum(belowCurve) / numTrials
totalArea <- (7 - 3) * 7^2 # Rectangle length x height = 196
areaUnderCurve <- ratio * totalArea
areaUnderCurve
```

```
## [1] 105.2226
```

We know the correct solution is

$$\int_3^7 x^2 dx = (\frac{x^3}{3})|_3^7 = \frac{7^3}{3} - \frac{3^3}{3} \approx 105.33$$

Monte Carlo Integration

Monte Carlo integration is a powerful method for computing the value of complex integrals using probabilistic techniques.

- When an analytic solution **is** available, simulation techniques can often be used for verification and validation purposes.
- When an analytic solution **is not** available, simulation techniques can often be used for this purpose.

Monte Carlo integration : $\theta = \int_a^b g(x)dx$

$$
\begin{aligned}
\theta = \int_a^b g(x)dx &= (b-a)\int_a^b g(x)\frac{1}{b-a}dx \\
&= (b-a)\int_a^b g(x)f(x)dx, \quad f \sim \text{Uniform}(a,b) \\
&= (b-a)E_f[g(X)]
\end{aligned}
$$

Draw a random sample x_i from the uniform distribution $f(x)$, then

$$\hat{\theta} \approx (b-a)\frac{1}{n}\sum_{i=1}^{n} g(x_i) = (b-a)\bar{g}_n$$

The Strong Law of Large Numbers tells us that $\hat{\theta}$ converges to θ with probability 1 (almost surely) as $n \to \infty$.

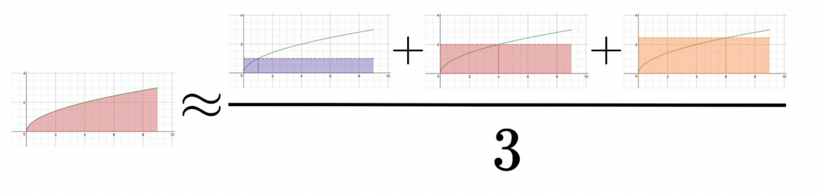
Monte-Carlo Integration

(not a formal proof) As $n \to \infty$, we can expect the sample of random draws $\{x_1, x_2, \cdots x_n\}$ to be more and more representative of the distribution made by the density function f.

The empirical cumulative distribution function based on the sample of size n converges almost surely to the actual cumulative distribution function as n gets large.

Visualizing Monte Carlo Integration

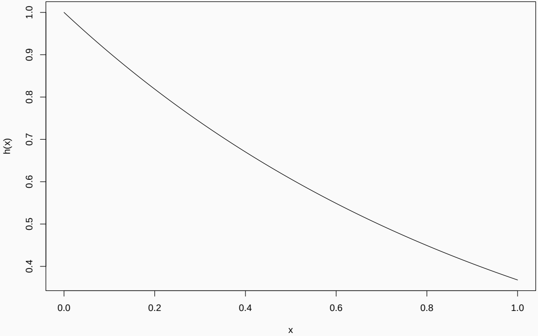
$$\hat{\theta} \approx (b-a)\frac{1}{n}\sum_{i=1}^{n} g(x_i) = \frac{1}{n}\sum_{i=1}^{n}(b-a)g(x_i) = \text{average of n rectangular areas}$$



Example

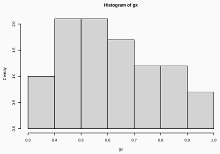
Let $\theta = \int_0^1 e^{-x} dx = 1 - 1/e = 0.63212$.

```
x <- seq(0,1, by =0.001)
h <- function(x){exp(-x)}
plot(x,h(x),type ="l")
```



Sample of size $m = 100$

```
m=100
x=runif(m,0,1)
gx=exp(-x)
hist(gx,prob=TRUE) # Does it look like exp?
```



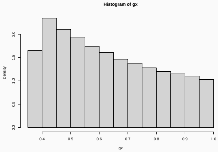

```
thetahat=mean(gx)
thetahat
```

```
## [1] 0.6138951
```

We can see how close we can come to this with samples of $m = 100$ and $m = 100000$ using Monte Carlo integration with respect to a uniform density.

Sample of size $m = 100000$

```
m=100000#Now try m=100000
x=runif(m,0,1)
gx=exp(-x)
hist(gx,prob=TRUE) # Does it look like exp?
```




```
thetahat=mean(gx)
thetahat
```

```
## [1] 0.632621
```

$m = 100$ vs $m = 100000$

- Of course, both are unbiased, but the standard error gets smaller as m increases.
- Monte Carlo integration not as efficient as numerical methods for 1d integrals
- Monte Carlo integration real efficiency comes into play in higher dimensional integrals

Classical Monte-Carlo integration algorithm

If we are able to sample directly from the density function $f(x)$ of X, we can estimate $E_f(h(X))$ for any function $h(x)$

1. Generate $x_1, x_2, \cdots, x_n$ iid from $\sim f(x)$
2. Compute $h(x_1), h(x_2), \cdots, h(x_n)$
3. Estimate $E_f[h(X)]$ by

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

By SLLN, $\bar{h}_n$ based on our sample will converge to $E_f[h(X)]$ as $n \to \infty$

Properties of the Monte-Carlo estimator $\bar{h}_n$

Properties of sample mean

Since $\bar{h}_n$ is a sample mean of $h(X)$, so by Central Limit Theorem, for large n, it's sampling distribution will

$$\bar{h}_n \approx N\left(\text{mean} = \mu = E_f(h(X)), \text{variance} = \sigma^2 = \frac{var(h(X))}{n}\right)$$

Knowing the distribution of $\bar{h}_n$ will allow us to create confidence intervals using normal quantiles.

Expected value of the estimate $\bar{h}_n$

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

$$\begin{aligned} E[\bar{h}_n] &= E\left[\frac{1}{n}\sum_{i=1}^{n} h(X_i)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} E[h(X_i)] \\ &= E[h(X)] \end{aligned}$$

- $\bar{h}_n$ is an unbaised estimator of $E[h(X)]$

Variance of the estimate $\bar{h}_n$

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

$$\begin{aligned} Var(\bar{h}_n) &= Var(\frac{1}{n}\sum_{i=1}^{n} h(X_i)) = \frac{1}{n^2}\sum_{i=1}^{n} Var(h(X_i)) \\ &= \frac{1}{n^2}(n.Var(h(X))) \\ &= \frac{1}{n}Var(h(X)) \end{aligned}$$

- If we do not know $Var(h(X))$, we can estimate it also using Monte Carlo integration since $Var(h(X)) = E_f[(h(X) - E[h(X)])^2]$ where X has density f.

Monte Carlo estimate of variance of MC estimate $\bar{h}_n$

$$
\begin{aligned}
Var(\bar{h}_n) &= \frac{1}{n} Var(h(X)) \\
&= \frac{1}{n} \int (h(x) - E_f[h(X)])^2 f(x) dx = \frac{1}{n} \int g(x) f(x) dx \\
&\approx \frac{1}{n} \frac{1}{n} \sum_{j=1}^{n} g(x_j) \\
v_n &\approx \frac{1}{n^2} \sum_{j=1}^{n} (h(x_j) - \bar{h}_n)^2
\end{aligned}
$$

Taking $g(x) = (h(x) - E_f[h(X)])^2$ and approximating $\int g(x) f(x) dx$ using MC integration with samples x_j from f and plugging into g

Distribution of $\bar{h}_n$

So, by central limit theorem, we have that

$$\bar{h}_n \approx N\left(\text{mean} = \mu = E_f(h(X)), \text{variance} = \sigma^2 = \frac{var(h(X))}{n}\right)$$

We may estimate

- μ by $\hat{\mu} = \bar{h}_n = \sum_{i=1}^{n} h(x_i)$ and
- σ^2 by $v_n = \frac{1}{n^2} \sum_{j=1}^{n} (h(x_j) - \bar{h}_n)^2$

Thus, a confidence interval for $\theta = \int h(x)f(x)dx$ of level $1 - \alpha$ is given by

$$(\bar{h}_n - z_{\alpha/2}\sqrt{v_n}, \bar{h}_n + z_{\alpha/2}\sqrt{v_n})$$

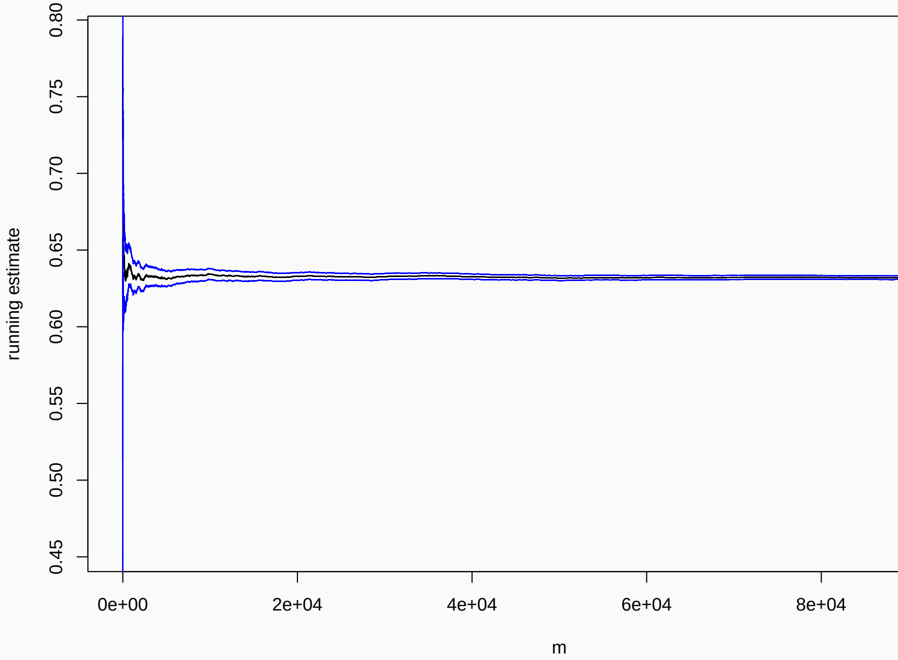
where $z_{\alpha/2}$ is defined by $P[Z > z_{\alpha/2}] = \alpha/2$ when $Z \sim N(0, 1)$.

MC integration example

$\theta = \int_0^1 e^{-x} dx = 1 - 1/e = 0.63212.$

```
m=100000#Now try m=100000
x=runif(m,0,1) # generate uniforms
gx=exp(-x) # compute g(X)

# compute cumulative mean(g(X))
thetahat_m = cumsum(gx[1:m])/(1:m)

# Function to estimate variance of thetahat_m = \bar{g}_m
var_k = function(k){
  # Estimate Var(thetahat_m) for any m from 1:n
  sum((gx[1:k] - thetahat_m[k])^2)/k^2
}
```

```r
# collect running estimates of
#the variance of the estimate
v_m = apply(t(1:m), 2, var_k)

# compute standard error
se_m = sqrt(v_m)
# Plot cumulative mean as iterations increase
plot(1:m, thetahat_m, type = "l", xlab = "m", ylab = "runni

# Add approximate 95% confidence band
# qnorm(1-.025,0,1)=1.959964
lines(thetahat_m +1.96*se_m, col="blue")
lines(thetahat_m -1.96*se_m, col="blue")
```


running estimate
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0e+00
2e+04
4e+04
6e+04
8e+04
m

Comparing efficiency of two estimators

Suppose we have two estimators of a parameter θ, $\hat{\theta}_1$ and $\hat{\theta}_2$. A common way to compare the relative efficiency of the two estimators is to look at their ratio of mean squared error

$$\frac{MSE(\hat{\theta}_1)}{MSE(\hat{\theta}_2)}$$

where MSE if defined for an estimator $\hat{\theta}$ by

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] = bias^2(\hat{\theta}) + Var(\hat{\theta})$$

For two unbiased estimators, this just amounts to comparing the ratio of their variances $\frac{Var(\hat{\theta}_1)}{Var(\hat{\theta}_2)}$.

Improving efficiency of Monte Carlo estimator

All of the Monte Carlo integral estimators are unbiased, so we only need to compare the variances.

To improve on the efficiency of our definite integral estimator, we look at techniques to reduce variance of the estimator

- Increase sample size - computationally costly, higher dimensions??
- Importance sampling