Stats 102C - Lecture 2-2: Monte Carlo Integration

Miles Chen, acknowledgements: Michael Tsiang

Week 2 Wednesday

Section 1

Lecture 2-2

Section 2

Applications of Monte Carlo Integration

Estimate an integral of any function

Let's say you want to estimate the integral of a function $h(x)$ on a closed interval (a, b):

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

Perhaps $h(x)$ is difficult to integrate directly.

Keep in mind the average value of $h(x)$ on the interval (a, b) is

$$h_{avg} = \frac{1}{b-a}I = \frac{1}{b-a}\int_a^b h(x)dx$$

Estimate an integral of any function

The average value of $h(x)$ on the interval (a, b) is

$$h_{avg} = \frac{1}{b-a} I = \frac{1}{b-a} \int_a^b h(x)dx$$

We can approximate this average with a Monte Carlo method:

$$h_{avg} = \frac{1}{b-a} \int_a^b h(x)dx = \int_a^b h(x) \frac{1}{b-a} dx = \int_a^b h(x)f(x)dx = \mathbb{E}_f[h(X)] \approx \bar{h}_n$$

The trick is to let $f(x) = \frac{1}{b-a}$, which is the PDF of a uniform distribution on the interval (a,b).

We can use this result to approximate the value of the original integral:

$$I = (b-a)h_{avg} \approx (b-a)\bar{h}_n$$

Estimate an integral of any function

We estimate $\mathbb{E}_f[h(X)]$ as follows:

1. Generate $x_1, x_2, \ldots, x_n \overset{\text{iid}}{\sim} Unif(a, b)$.
2. Compute $h(x_1), h(x_2), \ldots, h(x_n)$.
3. Estimate $\mathbb{E}_f[h(X)]$ by

$$\bar{h}_n = \frac{1}{n} \sum_{j=1}^{n} h(x_j)$$

4. Estimate I with

$$\hat{I} = (b-a)\bar{h}_n = \frac{(b-a)}{n} \sum_{j=1}^{n} h(x_j)$$

Example

$$h(x) = x^2$$

Estimate the following integral via Monte Carlo integration:

$$\int_2^5 h(x)dx = \int_2^5 x^2 dx$$

Answer using calculus

$$\int_2^5 x^2 dx = \frac{x^3}{3}\Big|_2^5 = \frac{125}{3} - \frac{8}{3} = \frac{117}{3} = 39$$

Estimate via Monte Carlo

```r
h <- function(x){x ^ 2}
set.seed(1)
n <- 10^5 # Specify the number of points to generate
# Generate n points from Unif(2,5)
X <- runif(n, 2, 5)
# Compute h(X)
h_X <- h(X)
# Compute mean(h(X))
mean(h_X)
```

```
## [1] 12.99609
```

$$\bar{h}_n = 12.9960949$$

$$\hat{I} = (b-a)\bar{h}_n = (5-2)\bar{h}_n = 3 \times \bar{h}_n = 38.9882848$$

Another Example

$$h(x) = [\cos(50x) + \sin(20x)]^2$$

Estimate the following integral via Monte Carlo integration:

$$\int_0^1 h(x)dx = \int_0^1 [\cos(50x) + \sin(20x)]^2 dx$$

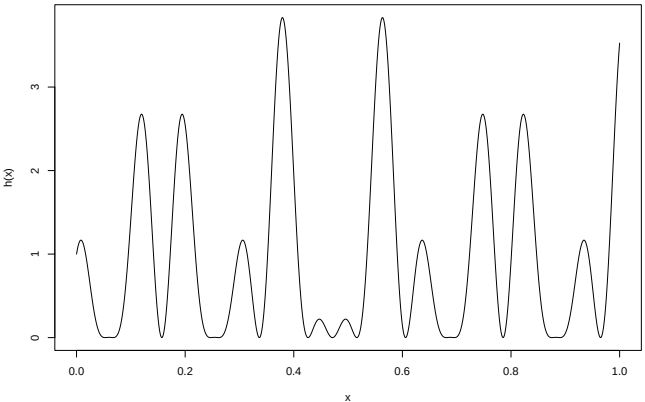
Wolfram alpha

https://www.wolframalpha.com/input/?i=%5Cint_0%5E1+%5B%5Ccos%2850x%29+%2B+%5Csin%2820x%29%5D%5E2+dx

Wolfram Alpha says the answer is

$$1 + \cos(30)/30 - \cos(70)/70 - \sin(40)/80 + \sin(100)/200 - 2/105 \approx 0.96520$$

```r
x <- seq(0, 1, by = .001)
h <- function(x) {(cos(50 * x) + sin(20 * x)) ^ 2}
plot(x, h(x), type = "l")
```



Estimate via Monte Carlo

```r
set.seed(1)
n <- 10^5 # Specify the number of points to generate
# Generate n points from Unif(0,1)
X <- runif(n, 0, 1)
# Compute h(X)
h_X <- h(X)
# Compute mean(h(X))
mean(h_X)
```

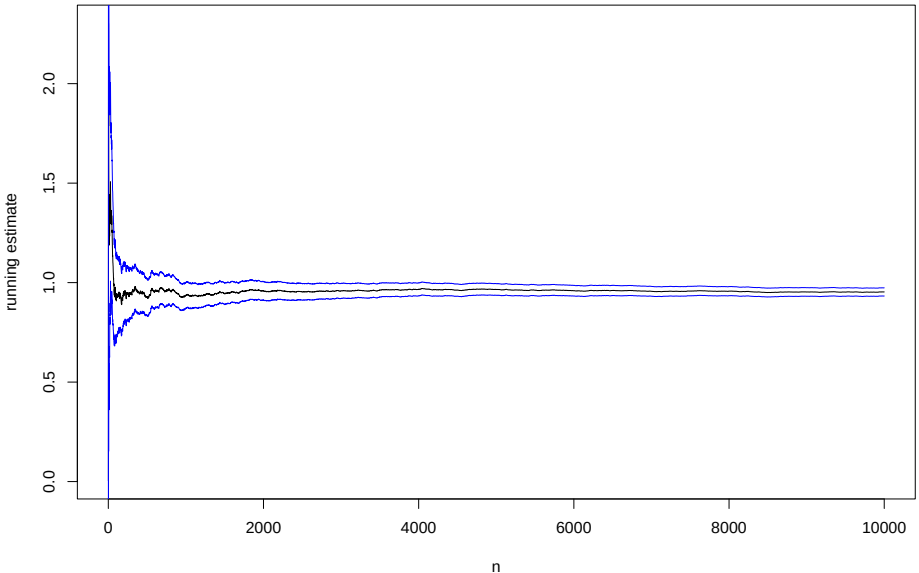
```
## [1] 0.966863
```

$$\bar{h}_n = 0.966863$$

$$\hat{I} = (b-a)\bar{h}_n = (1-0)\bar{h}_n = \bar{h}_n = 0.966863$$

```r
n <- 10^4 # graph just the first 10000 values

# Compute cumulative mean(h(X))
hbar_n <- cumsum(h_X[1:n])/(1:n)

# Function to estimate Var(hbar_n) (see slide 27)
var_m <- function(m) {
# Estimate Var(hbar_m) for any given m
sum((h_X[1:m] - hbar_n[m]) ^ 2) / m ^ 2
}

# running estimates of the variance
v_n <- apply(t(1:n), 2, var_m)

# Compute standard error
se_n <- sqrt(v_n)
```

```r
# Plot cumulative mean against iterations
plot(1:n, hbar_n, type = "l" , xlab = "n", ylab = "running estimate")

# Add approximate 95% confidence band
lines(hbar_n + 1.96 * se_n, col = "blue")
lines(hbar_n - 1.96 * se_n, col = "blue")
```



Another Example: Normal CDF

So this is a little bit of a silly example. We have a function to estimate the standard normal CDF already: `pnorm()`. The function `pnorm()` is technically an approximation. (As an approximation, it is accurate to 16 digits, so it is adequate for a computer using floating point numbers.)

That said, it is still an interesting Monte Carlo problem.

The Standard Normal CDF is:

$$F(x) = \Pr(Z \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

There is no closed form expression for $F(x)$.

We can use Monte Carlo integration for this.

Example: Normal CDF

Let $I(\cdot)$ denote the **indicator function**, so:

$$I(Z \leq x) = \begin{cases} 1 & \text{if } Z \leq x \\ 0 & \text{if } Z > x \end{cases}$$

The expected value of $I(Z \leq x)$ is

$$\begin{aligned} \mathbb{E}[I(Z \leq x)] &= 1 \cdot P(Z \leq x) + 0 \cdot P(Z > 0) \\ &= P(Z \leq x) \\ &= F(x) \end{aligned}$$

Now that the integral of interest is expressed as an expectation, we can use a Monte Carlo estimator to estimate $F(x)$. Let $h(x) = I(Z \leq x)$

Example: Normal CDF

We have $F(x) = P(Z \leq x) = \mathbb{E}_f[I(Z \leq x)]$, so we let $h(x) = I(Z \leq x)$.

1. Generate $Z_1, Z_2, \ldots, Z_n \sim \mathcal{N}(0, 1)$.
2. For each Z_i, compute:

$$h(Z_i) = I(Z_i \leq x) = \begin{cases} 1 & \text{if } Z_i \leq x \\ 0 & \text{if } Z_i > x \end{cases}$$

3. Estimate $F(x)$ by

$$\hat{F}(x) = \bar{h}_n = \frac{1}{n} \sum_{i=1}^{n} I(Z_i \leq x)$$

Note: This method generalizes to produce an estimator for the CDF of any random variable (if we can sample from its distribution).

It's silly and expensive, but it works

This example is silly. It's computationally expensive. At the same time it works at giving us a good estimate.

In summary, we're saying you can estimate the Normal CDF of a value like 1.96 by generating many many many random values from the normal distribution.

Then turn those results into 0s and 1s with a logical test. If the values are less than 1.96, it's 1. If not, it's 0.

Calculate the mean of the 0s and 1s to approximate the proportion of values in the normal distribution that are less than 1.96.

At the end of the day, this method depends on our ability to generate many random values from the normal distribution. Thankfully, we can do this.

Normal CDF

The random variable

$$I(Z_i \leq x) = \begin{cases} 1 & \text{if } Z_i \leq x \\ 0 & \text{if } Z_i > x \end{cases}$$

is a Bernoulli random variable with success probability $p = P(Z \leq x) = F(x)$.

The estimator

$$\hat{F}(x) = \bar{h}_n = \frac{1}{n} \sum_{i=1}^{n} I(Z_i \leq x)$$

is the sample proportion of successes in n trials.

Monte Carlo Estimate of Normal CDF

Producing a grid of Monte Carlo estimates of the normal CDF for some selected values. You'll see how the values improve as we increase the sample size.

```
set.seed(100)
x <- rnorm(10^8) # whole sample
bound <- qnorm(c(0.5, 0.8, 0.9, 0.95, 0.975, 0.99, 0.999, 0.9999))
res <- matrix(0, ncol = 8, nrow = 7)
for (i in 2:8) {
  for (j in 1:8) {
    res[i-1, j] = mean(x[1:10^i] < bound[j])
  }
}
```

Monte Carlo Estimate of Normal CDF

```
colnames(res) <- round(bound, 4)
res <- rbind(res, pnorm(bound))
rownames(res) <- c("100", "10^3", "10^4", "10^5", "10^6", "10^7", "10^8", "actual")
round(res, 6)
```

```
##                0   0.6745   0.8416   1.2816   1.6449   2.3263   3.0902    3.719
## 100    0.520000 0.750000 0.810000 0.870000 0.920000 0.980000 1.000000 1.000000
## 10^3   0.490000 0.738000 0.793000 0.890000 0.944000 0.986000 0.998000 1.000000
## 10^4   0.499200 0.748800 0.800400 0.902800 0.949700 0.989600 0.999400 0.999900
## 10^5   0.500820 0.750210 0.798950 0.899000 0.949420 0.990330 0.999210 0.999900
## 10^6   0.499268 0.749635 0.799824 0.900013 0.950231 0.990142 0.999028 0.999900
## 10^7   0.500060 0.750047 0.800067 0.899968 0.949982 0.990022 0.998997 0.999896
## 10^8   0.500006 0.750015 0.800018 0.900000 0.949981 0.990008 0.999001 0.999899
## actual 0.500000 0.750000 0.800000 0.900000 0.950000 0.990000 0.999000 0.999900
```

How many values do we need?

How many random values do we need to sample to achieve a certain level of accuracy in our estimates?

To answer this, we first need to know the variance of our estimate.

Our estimate, $\hat{F}(x)$ has been shown to be the sample proportion of successes in n Bernoulli trials.

The variance of the sample proportion is then $Var[\hat{F}(x)] = \frac{\hat{F}(x)(1-\hat{F}(x))}{n}$

Maximum variance occurrs when $\hat{F}(x) = 0.5$, so a conservative estimate of the variance is $Var[\hat{F}(x)] = \frac{1}{4n}$ and a conservative estimate of the standard error is $\frac{1}{2\sqrt{n}}$.

If we want to be 95% confident that our estimate $\hat{F}(x)$ is accurate to 4 decimal places, then we want to make sure that $2 \cdot SE \leq 10^{-4}$. Thus $n \geq 10^8$