Monte Carlo Integration

Robert Winslow

GalvanizeU

June 20th, 2017

Warm-up discussion

What does it mean to *describe* a probability distribution?

What does it mean to *use* a probability distribution?

Learning objectives

- Know when you should consider approximation methods. (Lecture)
- Build an intuition for Monte Carlo integration in general. (Lecture + Lab)
- Understand MC integration for approximating expectations. (Lecture + Lab)
- Understand MC error. (Lab)
- Understand how naive MC integration can be impractical. (Lab)

Model computation

Recall the definition of a probability distribution P:

- Non-negativity: $P(A) \geq 0$
- Normalization: $P(\Omega) = 1$
- Finite additivity: $A \cap B = \emptyset \rightarrow P(A \cup B) = P(A) + P(B)$

What does the definition of P say about how to *compute* with P?

Nothing.

Model computation

How to compute with P, and with random variables in general, is up to us.

Models can be arbitrarily complicated.

Computation techniques will be specific to the implementation of a model.

Querying

In particular, we care about how to *query* our model.

Recall some query types for a random variable X:

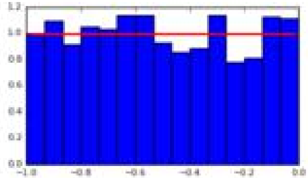
- Expectation: $E[X]$
- Draw: $a \sim X$
- PDF: $P(X = a)$

Sometimes, querying is easy.

But, in general, querying is difficult.

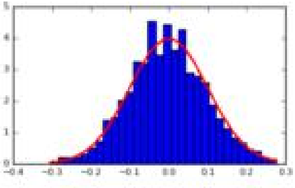
Computer science note: easy $\sim \mathcal{O}(1)$, difficult $> \mathcal{O}(1)$, very diffult $\geq \mathcal{O}(2^n)$.

Example: Continuous uniform distribution $\mathcal{U}(a, b)$



- $E[X]$: Closed form: $\frac{a+b}{2}$ (easy)
- $a \sim X$: Given by computing environment. (easy)
- $P(X = a)$: Closed form: $\frac{1}{b-a}$ (easy)

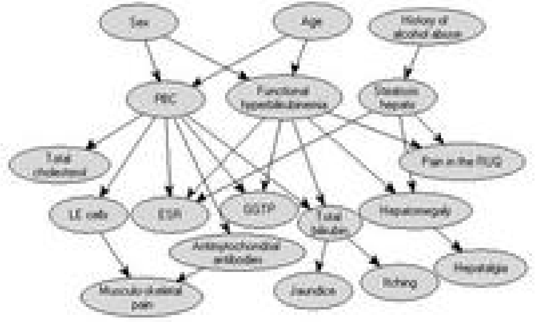
Example: Normal distribution $\mathcal{N}(\mu, \sigma^2)$



- $E[X]$: Closed form: μ (easy)
- $a \sim X$: Analytic: Box-Muller, Ziggurat, etc. (easy)
- $P(X = a)$: Closed form: $\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$ (easy)

N.B. Box-Muller, Ziggurat, and related methods are a special case for this distribution.

Example: Bayesian networks



- $E[X]$: Algorithmic: exponential in the treewidth (easy to very difficult)
- $a \sim X$: Algorthmic: depends on conditioned variables (easy to very difficult)
- $P(X = a)$: Algorithmic: chain rule on the network (medium)

N.B. Bayes models can be easy to very difficult.

Most distributions are not amenable to querying directly

N.B. Even the humble normal distribution requires a *specially-derived* approach for sampling.

Two big ideas:

- A few well-studied models have *easy* (closed-form or analytic) query procedures.
- Querying most models is *difficult* and requires approximations.

Questions so far?

Our approach

This week, we will examine several approximation techniques.

We start with *Monte Carlo Integration*.

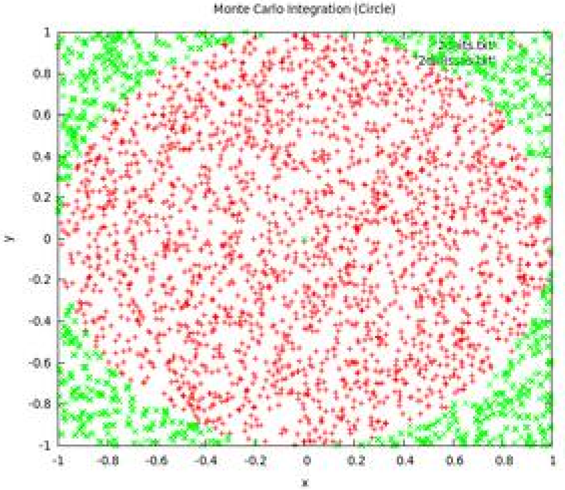
Monte Carlo Integration

MCI is a general technique to approximate the mean of an integral.

We use it when we can *draw* from X but we cannot calculate the *expectation* of X.

You have seen MCI before.

MCI: area of a circle



area of circle = volume of square * fraction of points that land in the circle

"Dartboard method"

MCI: History

The “Monte Carlo” method was developed by Ulam and von Neumann during the Manhattan Project.

Named after Monte Carlo, Monaco. Where there is supposedly a lot of gambling.

Used to create stochastic simulations to build “better” atomic bombs.

MCI: In general

MCI gives approximate integrals.

I is the expression to integrate:

Exact:

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

Approximate:

$$\widehat{I} = (b - a)MCI(f(x))$$

Monte Carlo Integration can be used when we want to compute the integral, but we can't do it analytically.

MCI: In statistics

We use MCI to approximate the expectation when exact computation is difficult:

$E[X]$	difficult
$a \sim X$	easy
$P(X = a)$	easy

Exact:

$$E[X] = \int pdf(X) X dx$$

Approximate:

$$\widehat{E[X]} = MCI(X)$$

MCI: algorithm

To approximately find the mean of the function $f(x)$ with MCI:

- Draw N samples from f.
- Average those samples.

Straightforward!

But, make sure you sample correctly.

MCI: worked example

$$I = \int_0^{10} x^3 dx$$

```
import numpy as np
import scipy.integrate as integrate
def f(x): return x**3

given_area = integrate.quad(f, 0, 10)[0]
given_mean = given_area / 10.

samples = f(np.random.uniform(0, 10, 1000))
approx_mean = np.mean(samples)
approx_area = approx_mean * 10
print(approx_mean, given_mean)
250.785410389 250.00000000000006
```