Monte Carlo

Solve problems using random numbers in combination with known probabilities of potential outcomes.

Formulated as a method by Stanislaw Ulam, named “Monte Carlo” by Nicholas Metropolis after the Monte Carlo Casino where Ulam’s uncle liked to gamble. First numerically implemented: John von Neumann → Manhattan Project

A first simple example: MC integration by hit or miss method

\[ I = \int_{a}^{b} f(x) \, dx \]

\[ \text{with: } I = \sum_{i=1}^{N} \mathbb{1}(g(Y_i) < F(x_i)) \]

where \( g(Y_i) \) is a well sampled in \([a, b]\)

(Ask later: What does this mean?)

1. Domain: \([a, b] \rightarrow \mathbb{R}[f(x), f(b)]\) space
2. Two areas: \( A_1 = (b-a) \cdot f(b) \)
   \( A_2 = (b-a) \cdot (f(b) - f(a)) \)
3. \((x_i, y_i)\) uniformly distributed in \( A_2 \)
4. Loop over all drawn points: 1 if \( f(x_i) \geq y_i \)
   \( \text{then increment count to } n \) by one: \( n = n + 1 \)
5. Result: \( I = \int_{a}^{b} f(x) \, dx \approx \frac{A_2 \cdot n}{N} + (A_1 - A_2) \)
Key for such MC "experiments": Random Numbers

(a) True random #s → radioactive

(b) Pseudo-random #s → generated from a deterministic algorithm (sequences of such #s eventually repeat themselves)

(c) Sub-random (also called quasi-random) #s

- numbers constructed to be statistically uniformly distributed, but that are not completely independent of each other

Generating Pseudo-Random #s

Often can use random # generator (RNGs) as black boxes, but still useful to understand the general principle

Requirements:

* Uniform distribution of random #s over interval \([a, b]\).
  - \(F(x) = \frac{x-a}{b-a}\)
  - \(f(x) = \frac{1}{b-a}\)
  So that \(F(x) = \int_a^x f(x)\ dx = \frac{x-a}{b-a}\)

* Generated #s must be statistically independent

* Sequences of generated #s should, if they are pseudo, have a very long period

* Sequence of generated #s must be reproducible
- Algorithm must be machine independent (portable)
- Algorithm must be fast

Simple example: Linear Congruential RNG

\[ \text{SEED} = (A \times \text{SEED} + C) \mod M \]
\[ x = \frac{\text{SEED}}{M} \quad (5 \mod 4 = 1) \]

- \( M, A, C, \text{SEED} \rightarrow \text{integers} \)
- \( x \in IR \)
- For any given SEED value, a new SEED value is calculated. From this new seed, a new value \( x \) is created and so on.

Since SEED is always an integer mod \( M \), we have \( 0 \leq \text{SEED} < M \rightarrow x \in [0, 1) \).

- Max. val of SEED is \( M-1 \).

A new cycle starts whenever \( \text{SEED} = \text{SEED}(\text{initial}) \) → want to make \( M \) very large → limited by datatype / machine dependent.

Example: \( A = 5, C = 1, M = 2^n, n = 5 \), \( \text{SEED} = 9 \)

\[ 5 \times 9 + 1 = 46 \]
\[ 9, 14, 7, 4, \ldots \]
Convergence of MC Approaches

- random variable \( X \)
- expect. val. \( E(X) = \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{prob. for realization} x_i \)

- variance \( \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 \)
- "mean quadratic deviation from the mean"

\[ \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 = \frac{1}{N} \sum_{i=1}^{N} x_i^2 - \frac{1}{N} \bar{x}^2 \]

standard deviation \( \sigma (X) = \sqrt{\sigma^2} \)

In a MC experiment, we typically draw \( N \) random variables \( x_i \) and then evaluate a deterministic expression with them:

\[ f(x_i) \quad \text{(which itself is then a random variable)} \]

and we will be interested in the mean

\[ \bar{g} = \frac{1}{N} \sum_{i=1}^{N} g(x_i) \quad \text{which itself is a random variable,} \]

\[ \text{expect val.:} \quad E(g) = \bar{g} = \frac{1}{N} \sum_{i=1}^{N} g(x_i) = \frac{1}{N} \sum_{i=1}^{N} \bar{g}(x_i) = \frac{1}{N} \cdot \bar{x} \cdot \bar{g} = \bar{g} = \bar{g} \]

\[ \text{variance:} \quad V(g) = \left( \frac{1}{N} \sum_{i=1}^{N} g(x_i) \right) \]

for statistically independent random vars:

\[ V(a_1 \cdot x_1 + a_2 \cdot x_2) = a_1^2 V(x_1) + a_2^2 V(x_2) \]

\[ \Rightarrow V(g) = \frac{1}{N^2} \sum_{i=1}^{N} V(g(x_i)) = \frac{1}{N^2} \sum_{i=1}^{N} V(g(x_i)) = \frac{1}{N} \cdot V(g) 
\]
Theorem: \( g \), a random var, is the mean of the random variables \( g + \epsilon \to \bar{g} = \bar{g} \) (expect values)

So for large enough \( N \): \( \bar{g} \approx \bar{g}_0 \) converging with \( \sigma = TV \sqrt{1/N} \)

This will only work in this way if the RV \( \epsilon \) delivers statistically useful random values!

Monte Carlo Simulation

1. Define domain of possible inputs
2. Generate inputs randomly from the density on the basis of a PDF
3. Perform a deterministic calculation using the inputs
4. Aggregate results of the individual calculations into a final result

Example: Estimating \( \pi \)

- Domain: \( [0, R] \times [0, R] \)

- PDF: uniform, draw inputs \( (x_i, y_i) \)

- Calc: if \( x_i^2 + y_i^2 \leq R^2 \), then increment

- Result: \( \pi \approx \frac{4 \times \text{#incircle}}{N} \)

- Error should decrease \( O(N^{-1/2}) \)