Monte Carlo for Optimization

- Two applications for Monte-Carlo Simulation

  Simulate biological processes

  (1) partial swarm optimization

  (2) genetic algorithm

Particle swarm

  a flock of birds, a school of fish, bees, ants...

  ✓ in this case, we'll look at birds
  find food in a field, each bird is a particle

  \( \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \)  d-dimensional coordinates.
  (e.g. d=2 in the birds' case)

Optimize Objective Function  \( f(\vec{x}) \)

\( \vec{x}^* \)  max (value)  1-Dim

Or
**Multi-modal**

In 2d, we can partition

\[ N^2 \]

\[ \times \]

\[ N \]

**d-dim**: \( N^d \)

e.g. \( d = 100 \)

Cannot search exhaustively when \( d \) is large.

Need more efficient search algorithm

\[ \rightarrow \text{exploitation} \]

\[ \rightarrow \text{exploitation} \]

We may not understand \( f(x) \) very much and do not have \( f'(x) \)

Exploration: multiple modes, randomness for exploration.

Back to Particle Swarm:

\( m \) particles, \( i = 1, 2, \ldots, m \)
\( \vec{x}_i(t) \) : position of particle \( i \) at time \( t \)  
\( \vec{v}_i(t) \) : velocity of particle \( i \) at time \( t \).

1. Randomly initialize \( (\vec{x}_i(t), \vec{v}_i(t), i = 1, ..., m) \)

\[
\vec{x}_{i(t+1)} = \vec{x}_i(t) + \vec{v}_i(t+1)
\]

Discretizing time \( \Delta t = 1 \)

\[
\vec{v}_{i(t+1)} = \vec{w} \vec{v}_i(t) + \vec{c}_1 \vec{r}_1 (\text{pbest}_i - \vec{x}_i(t)) + \vec{c}_2 \vec{r}_2 (\text{gbest} - \vec{x}_i(t))
\]

\( \vec{w} \) : inertia weight
\( \vec{c}_1 \) : cognitive coefficient
\( \vec{c}_2 \) : social coefficient

* \( \text{pbest}_i \) : the bird's best location
* \( \text{c}_1 \) : bird is aware of its best location
* \( \text{gbest} \) : out of all the birds, the best place so far
* \( \vec{c}_2 \) : birds are aware of group's best.

**Genetic Algorithm**

\[
\max f(\vec{x}) \rightarrow \text{fitness}
\]

\( \vec{x} \) : genetic representation

binary string (DNA sequence)
1. Initialization: Population of random strings

2. Selection: Use \( f(x) \) to select members of population to be parents

3. Cross-over and mutation:
   - To generate new generation then select again.

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \text{cross-over} \\
\end{array}
\]

- Random position then make swap

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & \text{mutation} \\
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \text{(the new generation)} \\
\end{array}
\]

4. Go back to "selection" step

Part 2. Monte Carlo Integration
Expectation

population of \((N \sim \infty)\) points \(\text{random sampling}\)

a random point \(X \sim f(x)\)

\(\text{e.g. heights}\)

\(\{x_1, x_2, \ldots, x_N\}\)

\[ E(X) = \text{population average} \]

\[ = \frac{1}{N} \sum_{i=1}^{N} x_i \]

\[ = \frac{1}{N} \sum_{\text{bins}} X \]

\[ = \sum_{\text{bins}} x \frac{N(x)}{N} \]

\[ = \sum_{\text{bins}} x f(x) \Delta x \]

\[ \frac{\Delta x}{0} = \int x \cdot f(x) \, dx \]

\[ E(\overline{X}) = \int x \cdot f(x) \, d(x) \]
More generally

\[ E(h(X)) = \int h(x) f(x) \, dx \]

\( \text{e.g. } h(x) = x^2 \text{ same} \)

\[ \mu = E(X) \]

**Variance**

\[ \text{Var}(X) = E((X - \mu)^2) = \sigma^2 \]

\( \text{population variance} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2 \)

\[ \text{Var}(h(X)) = E((h(X) - E(h(X))^2) \]

**Want to compute**

\[ I = E(h(X)) = \int h(x) f(x) \, dx \]

\( \text{e.g. } = \int \frac{1}{\sqrt{2\pi}} \, e^{-x^2/2} \, dx \)

\( \text{draw } X_1, X_2, X_3, X_4, \ldots X_n \sim f(x) \text{ independent} \)
Monte Carlo Approximation

\[ \bar{I} = \frac{1}{n} \sum_{i=1}^{n} h(x_i) \]

population vs. sample

population of \( N \approx \infty \) points \( \rightarrow \) random sample of \( n \) points

proportion

population average \( \bar{I} \) (fixed, not deterministic) \( \rightarrow \) estimate

Sample average \( \bar{I} \) (random, fluctuates) \( \rightarrow \) proportion

Special Case: avg of indicator function = proportion

\[ h(x) = \begin{cases} 
1 & x \in A \\
0 & x \notin A 
\end{cases} \]

\[ = 1_A(x) \text{ or } 1(x \in A) \]

population average of \( 1(x_i \in A) \)

\[ = \frac{1}{N} \sum_{i=1}^{n} 1(x_i \in A) \]

: population proportion \( = P(A) \)
Ex. 1.

Want \( I = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \, dx \)

\[ \downarrow \quad h(x) \quad \downarrow \quad f(x) \]

generate \( X_1, X_2, \ldots, X_n \sim \mathcal{N}(0, 1) \) indep

\[ \hat{I} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{|X_i|} \]

Ex. 2. Approximate \( \pi \)

Ex. 2. Approximate \( \pi \)

\[ P(A) = \frac{|A|}{|\Omega|} = \frac{\pi}{4} \]

Monte-Carlo Method

\[ \frac{m}{n} \approx \frac{\pi}{4} \]

frequency \( \approx \) probability

\[ \frac{\pi}{4} = \frac{4m}{n} \]
Can think of this as

\[ P(A) = E\left(1 \left(\frac{x^2 + y^2}{2} \leq 1\right)\right) \]

\[ = \int_0^1 \int_0^1 1 \left(\frac{x^2 + y^2}{2} \leq 1\right) \, dx \, dy \]

\(x, y \sim \text{Unif}[0, 1]\) indep

**Buffon's Needle**

Approximate \(\pi\) by dropping needles on a grid of parallel lines and calculating the probability that it will cross the line.

\(Y = \text{center of needle} \sim \text{Unif} \left[0, \frac{d}{2}\right]\)

\(\Theta = \text{angle} \sim \text{Unif} \left[0, \pi\right]\) :
\[ P \left( Y \leq \frac{l}{2} \sin(\theta) \right) = \frac{\frac{l}{2} \sin(\theta)}{\left| \Omega \right|} \]

\[ = \int_{0}^{\frac{\pi}{2}} \frac{\frac{l}{2} \sin(\theta)}{\frac{l}{2} \pi} d\theta \]

\[ = \frac{\frac{l}{2} \left( -\cos(\theta) \right) \bigg|_{0}^{\frac{\pi}{2}}}{\frac{d}{2} \pi} \]

\[ = \frac{l}{d \pi} \]

\[ = \frac{2l}{d \pi} \text{ probability} \]