Lecture on parameter estimation and stochastic mode reduction:
Thursday May 17, 2-3:50 PM in Sage 2112.

In a pure sense, Monte Carlo methods refer to computational techniques in which a deterministic problem is formulated in terms of a stochastic problem, and then numerical simulations are performed to sample the stochastic problem.

Examples of stochastic representations of deterministic problems:

\[ \mathcal{K}(x) = \nabla^2 u + \nabla \cdot (\mathbf{c}(x)) \nabla u = r \]

for \( x \in D \)

\[ u(x) = f(x) \text{ for } x \in \partial D \]

Dynkin's formula:

\[ u(x) = \mathbb{E}\left[ \int_0^{\xi(x)} r(\mathbf{x}(t'), t') dt' \mid \mathbf{x}(0) = x \right] \]

\[ + \mathbb{E}\left[ f(\mathbf{x}(\xi(x))) \mid \mathbf{x}(0) = x \right] \]

where \( \mathbf{x} \) satisfies SDE

\[ d\mathbf{x}(t) = \mathbf{a}(\mathbf{x}(t)) \, dt + \sigma(\mathbf{x}(t)) \, d\mathbf{w}(t) \]

\[ a^2 = \nabla \mathbf{c} \quad \frac{1}{2} \sigma \sigma^T = \mathbf{K} \]

killed at rule C.
**Parabolic problems:**

\[
\frac{\partial u(x,t)}{\partial t} = \chi(x,t): \nabla \nabla u + \nabla (x,t) \cdot \nabla u
\]

\[
- c(x,t) u + r(x,t)
\]

for \( x \in D \)

\[
u(x, t=0) = u_{in}(x)
\]

\[
u(x, t) = f(x) \text{ for } x \in \partial D
\]

Solve by random method of characteristics running backward in time.

---

**Associated SDE:**

\[
d\overline{x} = \alpha(\overline{x}(t'), t') \, dt' + \sigma(\overline{x}(t'), t') \, dw(t')
\]

\[
\sigma = \nabla \chi, \quad \frac{1}{2} \sigma^T \sigma = \chi
\]

Kill at rate \( c \)

\[
u(x, t) = \mathbb{E} \left[ \int_{T}^{t} r(\overline{x}(t'), t') \, dt' \mid \overline{x}(t' = t) = x \right]
\]

\[
+ \mathbb{E} \left[ u_{in}(\overline{x}(0)) \, I \{ T = 0 \} \mid \overline{x}(t' = t) = x \right]
\]

\[
+ \mathbb{E} \left[ f(\overline{x}(T)) \, I \{ T = T_0 \} \mid \overline{x}(t' = t) = x \right]
\]

\[
T = \max \left[ T_0, 0 \right] - \chi(x, t), \chi
\]
Calculate the average of some physical quantity against some ensemble (canonical, microcanonical, etc.) of possible configurations with prescribed probability distribution.

\[ \langle A \rangle = \int A(\mathbf{x}) \, e^{-\mathcal{H}(\mathbf{x})/k_b T} \, d\mathbf{x} \]

\[ Z = \int e^{-\mathcal{H}(\mathbf{x})/k_b T} \, d\mathbf{x} \]

Suppose \( A \) (energy, end-to-end distance, force on walls of container)

The high dimensionality creates challenges for traditional deterministic integration methods. So one instead usually attempts a Monte Carlo sampling of the probability distribution of configurations which has a dynamic character (go from a starting configuration, randomly make some changes, accept or reject changes, go to new configuration, repeat...Metropolis algorithm)

This is mostly a Markov chain procedure. But sometimes one includes an SDE component: hybrid Monte Carlo/molecular dynamics simulations. The idea behind behind these procedures is that the probability distribution we want to sample (canonical distribution) is the stationary limit distribution of the following SDE system:

\[ \frac{d\mathbf{q}}{dt} = \mathbf{p} \]

\[ \frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{p}} \mathcal{H} + \mathbf{\Sigma}^{-1} \mathbf{f} \]
In practice, MD trajectories are expensive to simulate. So usually combine with gross Monte Carlo methods.

Simulated annealing
(stochastic optimization is 5d MC?)

Goal: Find global minimum at $U(x)$

where $x$ is high dimensional,

$U$ is complicated.

$\tilde{x} \xrightarrow{\text{down gradient}} x$

$\frac{d\tilde{x}}{dt} = -\nabla U(x(t)) dt + \sqrt{2 T(t)} \sigma dW(t)$

$T(t) \xrightarrow{\text{as } t \to \infty} 0$

Generic approach to Monte Carlo methods: Convert the deterministic problem to a stochastic representation, typically involving random trajectories (SDEs, Markov chains). This step: for some problems the representation is well known, but to develop new Monte Carlo applications, the general approach is to formulate the problem in terms of an integral equation, see if the kernel is nonnegative and normalizable, and if so, could it be interpreted as a probability distribution.
After this is done, the stochastic representation generically looks like:

\[
\mathbb{E} G(\overline{X}) \approx \frac{1}{N} \sum_{n=1}^{N} G(\overline{X}(n))
\]

Numerical approximation:

See Kloeden + Platen, Sec. 9.3

Numerical error:

i) \( N \) is finite (sampling/statistical error)

ii) Approximating the prob. dist. for the samples (discretization)

(Systematic error)

Systematic error creates bias when

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} G(\overline{X}(n)) \neq \mathbb{E} G(\overline{X})
\]

Sampling error (suppose zero bias)

\[
\frac{1}{N} \sum_{n=1}^{N} \left( G(\overline{X}(n)) - \mathbb{E} G(\overline{X}) \right)
\]

With independent samples, sampling error has mean 0 (no bias)

\[
\text{Variance} = \frac{1}{N^2} \sum_{n=1}^{N} \sigma^2_{\overline{X}_n} = \frac{\sigma^2_{\overline{X}}}{N}
\]

\[
\sigma^2_{\overline{X}} = \mathbb{E} \left( G(\overline{X}(n)) - \mathbb{E} G(\overline{X}) \right)^2
\]
\[
\sigma^2_G = \left\langle \left( G \left( X^{(m)} \right) - \left\langle G \left( X \right) \right\rangle \right)^2 \right\rangle
\]

**Standard deviation** = \( \frac{\sigma_G}{\sqrt{N}} \)

Half-order accurate. Hard to improve this.

Variance reduction methods: reduce the prefactor. Quasi-Monte Carlo methods.

When are Monte Carlo methods a good thing to try?

- Need the solution of PDE at a small number of points or if you just care about a few numbers (functionals) involving the solution.
- Complicated geometry (deterministic meshes may be hard to generate).
- High dimensions (too many grid points)
- Low accuracy good enough (couple of percent)

In short, Monte Carlo methods are a good way to get reasonable approximations to difficult computations with fairly simple algorithms. Mascagni though implemented a calculation of capacitance in a cube with Monte Carlo methods with 8 digits of accuracy (best ever).

Modern application to biomolecular electrostatics problem (Mascagni + Simonov 2004)

Modern application to biomolecular electrostatics problem (Mascagni + Simonov 2004)
\[ E_i, \varepsilon_{ex} : \text{electric permittivity} \]

Calculate electrostatic energy:
\[
\rho = \sum q_j \delta (x - x^{(j)})
\]
\[
E = \frac{1}{2} \sum q_j u(x^{(j)})
\]

where \( u \) is electrostatic potential

\[
\varepsilon_i \partial_n u_{in} = -\rho \quad \text{for} \quad x \in \Gamma
\]

\[
\Delta u_{ex} - K^2 u_{ex} = 0 \quad \text{for} \quad x \in \mathcal{D}
\]

\[
K^2 = \frac{2 J e^2}{\varepsilon_{ex} k_B T}
\]

\[ u_{ex} \to 0 \quad \text{as} \quad |x| \to \infty \]

\[ u_{in}(x) = u_{ex}(x) \quad \text{for} \quad x \in \partial \mathcal{D} \]

\[ \varepsilon_i \partial_n u_{in} = \varepsilon_{ex} \partial_n u_{ex} \quad \text{for} \quad x \in \partial \mathcal{D} \]

Key idea to improve solution to Laplace equation: "Walk-on-spheres" or "walk on subdomains" algorithms.

Suppose the domain \( \mathcal{D} \) were a ball

\[ \mathcal{D} = B(x_c, r) \]

Want to know \( P_D \) for \( \nabla \cdot \mathbf{j}(\mathbf{r}_0) \)

From absorption probabilities:
\[ p(\overline{y}; \overline{x}) \text{ PDF for exit position} \]
\[ \overline{y} \text{ for Brownian motion starting at } \overline{x} \]
\[ A_x p(\overline{y}; \overline{x}) = 0 \text{ for } \overline{x} \in D \]
\[ p(\overline{y}; \overline{x}) = \delta(\overline{y} - \overline{x}) \text{ for } \overline{x} \in \partial D \]

Brownian motion: \( A = \frac{1}{2} A \)