Monte Carlo Methods: An Introduction

Parthapratim Biswas

Physics and Astronomy

The University of Southern Mississippi

OUTLINE

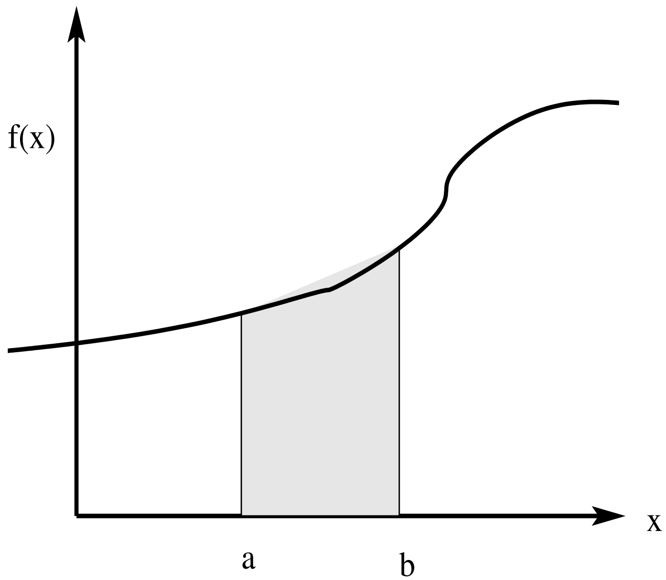
- Monte Carlo integration
- Direct and importance sampling
- Random numbers and probability distributions
- Function optimization in high-dimensional spaces
- Metropolis and related algorithms
- Monte Carlo simulation of (disordered) solids

OUTLINE

- Monte Carlo integration
- Direct and importance sampling
- Random numbers and probability distributions
- Function optimization in high-dimensional spaces
- Metropolis and related algorithms
- Monte Carlo simulation of (disordered) solids

Monte Carlo, Casino, and random numbers are all related!

Monte Carlo integration



$$\int_a^b f(x)\,dx = (b-a)\,\langle f \rangle$$

Estimator of average $\langle f \rangle$:

$$\langle f \rangle_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \qquad \text{(sampled at N points)}$$

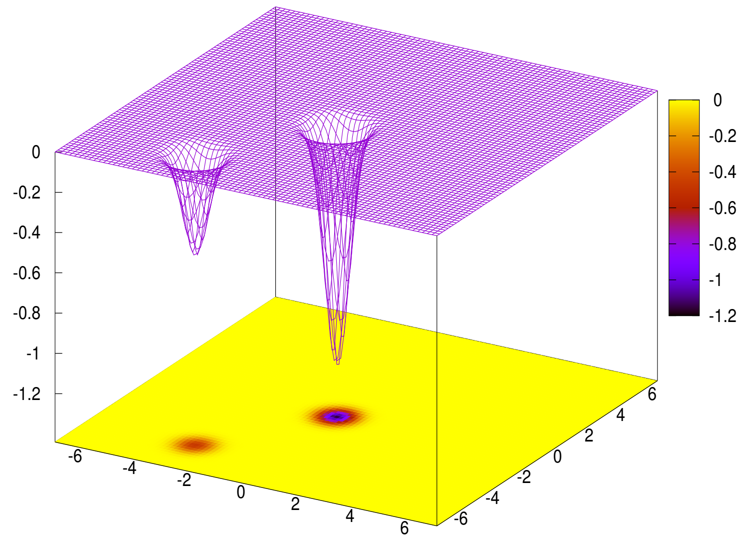
$$\int_a^b f(x)\,dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

Key problem: How to choose of x_i from $[a, b]$?

Observarions

- In one- and two-dimensional spaces, efficent numerical schemes exist.
- Think of Newton-Cotes (Traphezoidal and Simpson rules) and Gaussian quadrature
- Curse of high dimensions (except for mean-field theorists!)

Direct vs. Indirect sampling



$$\langle f \rangle \approx \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} f(x_i, y_j)$$

- Only two regions contribute to the integral above
- Direct sampling is inefficient (random or uniform grid) in higher dimensions
- Prior knowledge of the region(s) of *importance* can help significantly

Key idea

Instead of direct sampling, find a target density that largely defines over the region(s) of importance

Importance sampling

Elementary concepts

$$I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{g(x)}\,g(x)\,dx = \int_{y^{-1}(a)}^{y^{-1}(b)} \frac{f(x)}{g(x)}\,dy, \text{ where } y(x) = \int^x g(t)\,dt$$

- Choose a suitable $g(x)$ close to $f(x)$
- Sample y uniformly from $[y^{-1}(a), y^{-1}(b)]$
- Obtain x by inverting $y(x) = \int^x g(t)\,dt$
- Integrate $\frac{f(x)}{g(x)}$

Example

$$I_{exact} = \int_0^1 e^x\,dx = e - 1$$

$$\text{Assume, } g(x) = 1 + x, \quad I = \int_0^1 \frac{e^x}{1+x}(1+x)\,dx = \int_0^{\frac{3}{2}} \frac{e^{\sqrt{1+2y}-1}}{\sqrt{1+2y}}\,dy$$

$$\text{Here, } y = \int^x (1+t)\,dt = x + \frac{x^2}{2} \quad \rightarrow \quad x = \sqrt{1+2y} - 1$$

- Use direct sampling and importance sampling to compute the integral for a given number (say, 5000) of samples

Importance sampling

General algorithm

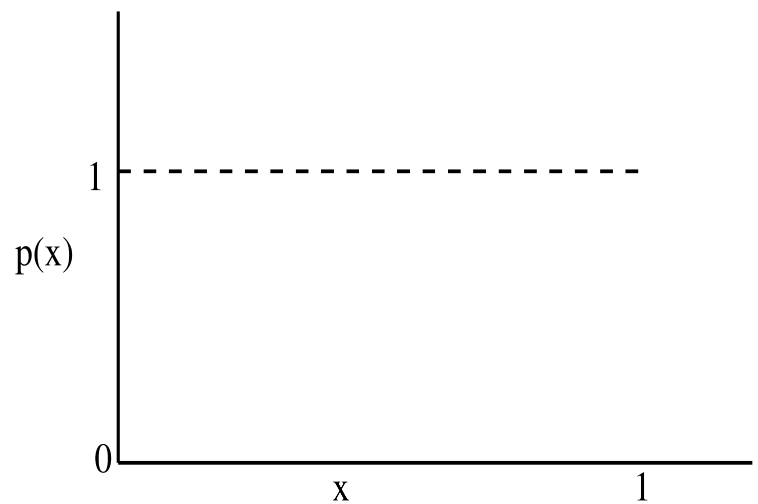
$$I = E_p\{f(x)\} = \int f(x)p(x)\,dx = \int f(x)\left[\frac{p(x)}{g(x)}\right] g(x)\,dx = \int f(x)\,\omega(x)\,g(x)\,dx$$

1. Draw $x_1, x_2, x_3, \cdots, x_j$ from a trial density g(.)
2. Compute the importance factor
$$\omega_j = \frac{p(x_j}{g(x_j)}$$
3. Approximate I by,
$$\begin{aligned}\hat{I} &= \frac{\omega_1 f(x_1) + \omega_2 f(x_2) + \cdots + \omega_j f(x_j)}{\omega_1 + \omega_2 + \cdots + \omega_j} \\ &= \frac{1}{W}\sum_{m=1}^{j} \omega_m f(x_m) \qquad (1)\end{aligned}$$
4. In Eq. (1), ω is needed up to a multiplicative factor. Also, this gives generally a small mean-squared error. The is is related with Markov Chain Monte Carlo algorithms.

Random numbers

Random Numbers

- Monte Carlo methods heavily rely on random numbers (RNs)
- Earlier RNs were produced manually – disc rolling, coin flipping, roulette spinning, etc.
- Physical processes, such as noises in PC, radioactivity, and universal background radiation can be used to generate RNs.
- Modern RNs are computer generated – they pass most of the statistical tests.
- Linear congruential generators are most common for generating pseudorandom sequences.
- We assume that a good uniform RNG, from 0 to 1, is available.



Random variates from a given probability density

How to generate different probability density?

- In MC simulations, one frequenty employs different probability densities: Uniform, Normal, Gamma, Exponential, etc.
- Generating some densities in higher dimension (≥ 4) can be nontrivial.
- Different methods exist for this purpose.
- Inverse-Transform and Acceptance-Rejection methods are two prominent examples.
- Generation of random vectors on the <u>surface</u> of unit hypersphere is often needed in MC simulations.

Inverse-transform method

Key idea

Let X be a random variable with cumulative distribution function (CDF) F, of a probability density $f(x)$, and that it is invertible,

$$F^{-1}(u) = \inf\{x : F(x) \geq u\},\ 0 \leq u \leq 1$$

It follows that, if $U \sim$ U(0,1), then,

$$X = F^{-1}(U)$$

Example 1

Generate the exponential distribution with a density,

$$f(x) = \lambda e^{-\lambda x} \text{ for } x \geq 0$$

Solution: The CDF is given by,

$$F(a) = \int_0^a f(x)\,dx = \int_0^a \lambda e^{-\lambda x}\,dx = 1 - e^{-\lambda a}$$

$$1 - e^{-\lambda x} = u \quad \rightarrow \quad x = -\frac{\log(1-u)}{\lambda} \quad \rightarrow \quad X = -\frac{\log(1-U)}{\lambda} \sim -\frac{\log U}{\lambda}$$

Inverse-transform method

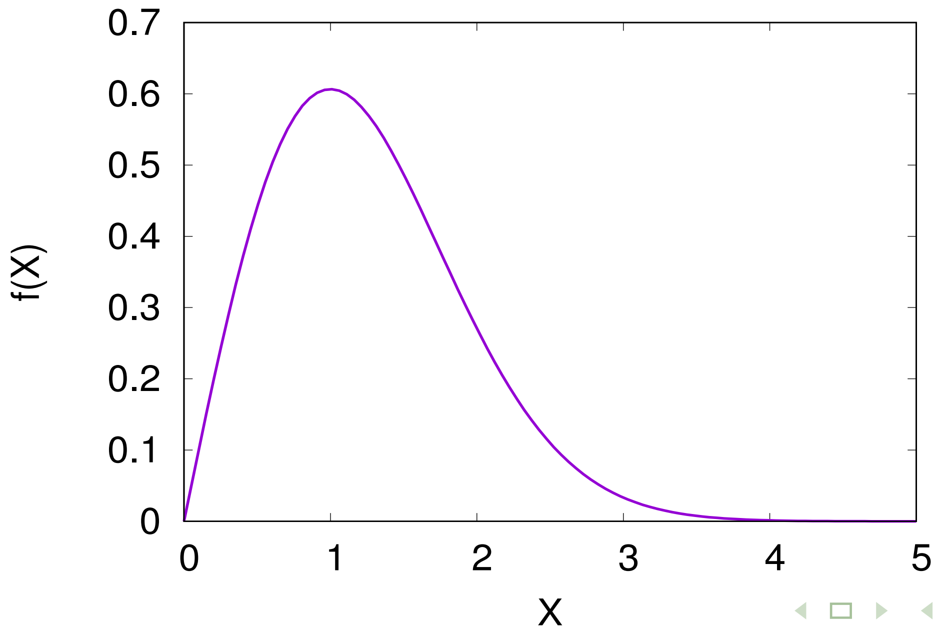
Example 2: Rayleigh distribution

The Rayleigh distribution with parameter $\sigma > 0$ has the density,

$$f(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad \text{for} \quad x \geq 0$$

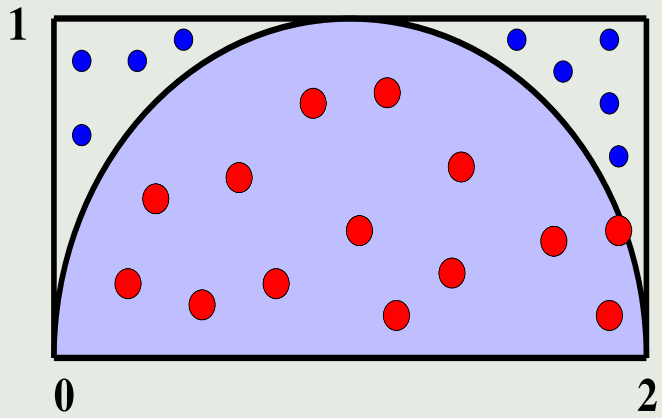
$$\text{Here, } F(a) = \int_0^a \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) = 1 - \exp\left(-\frac{a^2}{2\sigma^2}\right)$$

$$\text{Solving } u = F(x) = 1 - e^{-\frac{x^2}{2\sigma^2}} \quad \rightarrow \quad x = \sqrt{-2\sigma^2 \log(1-u)} = \sqrt{-2\sigma^2 \log(u')}$$



Acceptance-Rejection Method

Example 1: Computation of π



$$\frac{N_r}{N_r + N_b} = \frac{\text{Area of the semicircle}}{\text{Area of the rectangle}} = \frac{\pi/2 \times r^2}{1 \times 2} = \frac{\pi}{4}$$

$$\therefore \pi = 4.\frac{N_r}{N_r + N_b} = 4 \times r, \quad r = \text{acceptance ratio}$$

$$N_r(N_b) = \text{Number of red (blue) balls}$$

Key idea and the algorithm

Let $g(x)$ be a proposal density, such that $\phi(x) = Cg(x)$, where $C = \sup\{f(x) : x \to [a, b]\}$ and $\phi(x) \geq f(x)$. Then the A-R algorithm reads:

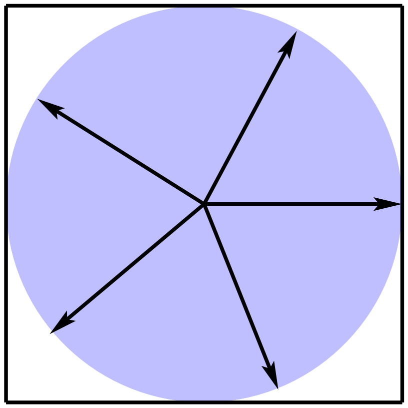
1. Generate X from $g(x)$, and U from U(0,1)
2. Obtain a minimal C, such that $Cg(x) \geq f(x)$
3. If $U \leq \frac{f(X)}{Cg(X)}$, accept X. Otherwise retrn to step 1

Problem

Generate a random variable X from the semicircular distribution,

$$f(x) = A(R)\sqrt{R^2 - x^2} = \frac{2}{\pi R^2}\sqrt{R^2 - x^2}$$

Random vector generation



- To sample random variates $\mathbf{x}$ in high dimensions (or a Markov chain $\mathbf{z_t}$ in the state space).
- Conventional methods for 2 and 3 dimensions do not work in d dimensions.
- Prodecures for generating RVs inside a hypersphere are different than on a hypersurface

Two important relations

- Volume of a sphere in d dimensions:

$$V_d(R) = \int_{\sum_i x_i^2 \leq R^2} dx_1\, dx_2 \ldots dx_d = \left[\frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)}\right] = A_d R^d$$

$$\text{Here, } \Gamma(n+1) = n\,\Gamma(n) = n!, \quad \Gamma(1/2) = \sqrt{\pi}, \quad \text{and} \quad A_{(3,2)} = (\frac{4\pi}{3}, \pi)$$

- Acceptance ratio r in high dimensions:

$$r = \frac{\text{Vol. of a } d\text{-dimensional unit sphere}}{\text{Vol. of a } d\text{-dimensional cube of length 2}} = \frac{1}{d\, 2^{d-1}} \frac{\pi^{d/2}}{\Gamma(d/2)} \rightarrow 0, \text{ for } d \geq 10$$

Random vector generation

Algorithm: Random vectors inside the hypersphere

- Generate a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ from a normal distribution, $\mathcal{N}(0,1)$.
- Compute $\sigma = U^{1/d}$, where $U \sim \mathcal{U}(0,1)$.
- Return $\mathbf{R} = \frac{\sigma \mathbf{X}}{||\mathbf{X}||}$

Rubenstein 2007

Algorithm: Random vectors on the hypersphere

- Compute $\sigma = \frac{1}{\sqrt{d}}$

  Generate a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ from a normal distribution, $\mathcal{N}(0,\sigma)$.
- Return $\mathbf{R} = \frac{\mathbf{X}}{||\mathbf{X}||}$

Krauth 2006

From importance sampling to function optimization

Intuitive ideas

- Define the average of $f(\mathbf{x})$

$$\langle f(\mathbf{x}) \rangle = \int f(\mathbf{x})\, \rho(\mathbf{x}, \beta)\, d\mathbf{x} \qquad \beta = \text{a suitable parameter}, \beta \geq 0$$

  and use the normalizd density $\rho(\mathbf{x}, \beta)$

$$\rho(\mathbf{x}, \beta) = \frac{e^{-\beta H(\mathbf{x})}}{\int e^{-\beta H(\mathbf{x})}\, d\mathbf{x}} = \frac{e^{-\beta H(\mathbf{x})}}{Z}, \quad \text{where} \quad Z = \text{Partition function}$$

- $\rho(\mathbf{x}, \beta)$ can be constructed, in general, up to a multiplicative constant (no Z information)
- Large values of $\rho(\mathbf{x}, \beta)$ are of interest here; correspond to low values of $H(\mathbf{x})$
- Good samples of $\rho(\mathbf{x})$ originate from the region of $\mathbf{x}$ that likely to minimize $H(\mathbf{x})$.
- These considerations lead to the Metropolis and related algorithms.
- Equilibrium statistical mechanics provides a theoretical framework and the sampling density $\rho(\mathbf{x}, \beta)$.

Metropolis Monte Carlo

Markov chains

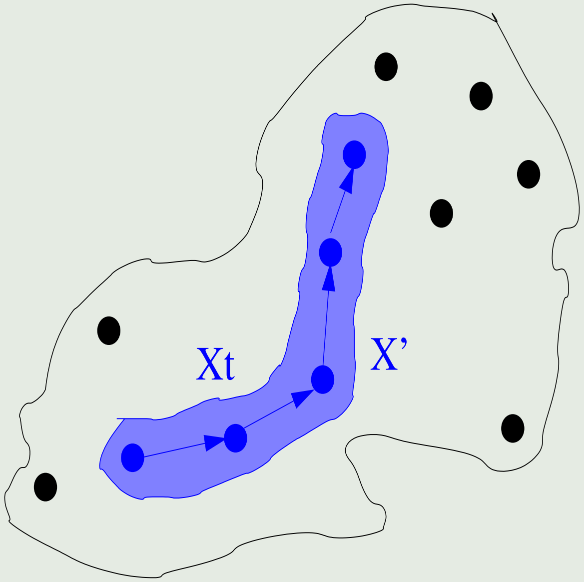


Figure: Generation of a Markov chain in state-vector space

Metropolis Algorithm

1. Start from the current state x^t and generate x^1 with a symmetric transition rule, $T(x^t, x^1) = T(x^1, x^t)$ and $\Delta H = H(x^1) - H(x^t)$
2. Generate a random number $U \sim \mathcal{U}[0,1]$
3. Accept $x^{t+1} = x^1$, if $r < \rho(x^1)/\rho(x^t) = \exp(-\Delta H)$ and let $x^{t+1} = x^t$ otherwise
4. For symmetric $T(x, y)$, both algorithms are identical

Metropolis-Hastings Algorithm

1. $T(x, y) \neq T(y, x)$ but $T(x, y) > 0$ if $T(y, x) > 0$
2. Generate a random number $U \sim \mathcal{U}[0,1]$ and form

$$r(x, y) = \min\left[1, \frac{\rho(y)T(y, x)}{\rho(x)T(x, y)}\right]$$

3. Accept $X^{t+1} = X^1$, if $U < r(x, y)$ and let $X^{t+1} = X^t$ otherwise

Markov Chain Monte Carlo (MCMC)

Comments

- Metropolis works fine for our purpose
- How to choose a Markov transition rule $T(\mathbf{x}, \mathbf{y})$?
- The rule must leave the target distribution, $\rho(\mathbf{x})$, invariant
- Often Ts are chosen for convenience
- More general approaches are needed – Gibbs sampling, partial resampling

Research stuffs

- Gibbs sampling (Geman and Geman 1984)
- Partial resampling techniques (Goodman and Sokal 1989)
- Generalized conditional sampling (Liu and Sabati 2000)
- Hybrid Monte Carlo

Hunting the global minimum in a rugged landscape: Simulated annealing

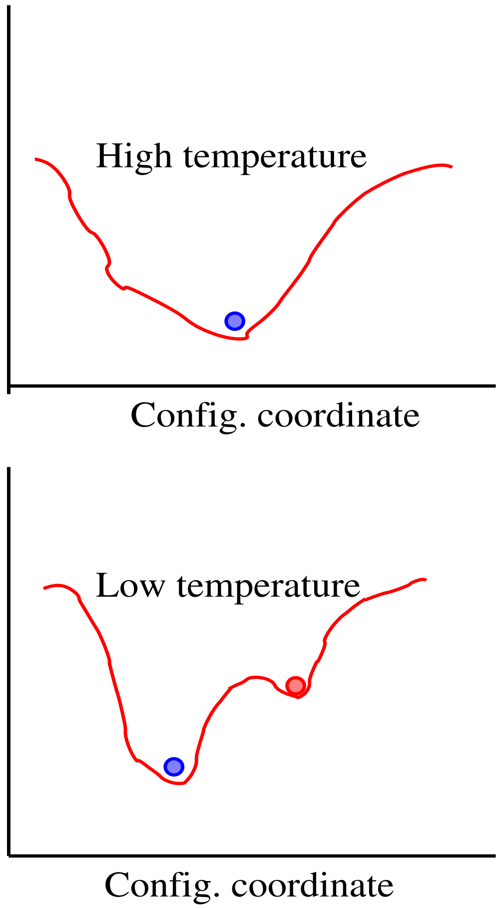


Figure: Simulated annealing

Simulated annealing

- Evolution toward thermodynamic equilibrium (i.e., approaching target 'Boltzmann' density)
- Target density is determined by equilibrium statistical mechanics
- Global minimum is reachable in principle but infeasible in practice (unrealistic logarithmic cooling)
- Plagued by local minima
- Complexity and dimension of the objective-function space ("rugged landscape")
- No gradient information needed (Atta-Fynn talk)
- Smart cooling protolcol can help $T_{n+1} = f(T_n, \beta)$

List of programs for Tutorial session

1. P1: Generation of random variates: Exp, Rayleigh, Cauchy, semicircular
2. P2: Random vectors *within* a hypersphere in n dimension
3. P3: Random vectors *on* the surface of a hypersphere
4. P4: Function minimization via Metropolis Monte Carlo
5. P5: Random number genration inside a three-dimensional cube
6. P6: Simple nearest-neigbor list generation
7. P7: Two-body or pair-correlation function
8. P8: Reduced three-body or bond-angle distribution

"Come, my friend. 'T is not too late to seek a newer world" Lord Tennyson

Let us explore the beautiful world of disordered materials