Monte-Carlo Methods and Sampling for Computing

Estimating Averages, Quantiles, and Ratios

Francesco Banterle, Ph.D. - July 2021

Monte-Carlo Algorithms

Introduction

- In simple Monte-Carlo, we typically want to estimate three possible quantities:
  - Averages/Expected values
  - Median values/Quantiles
  - Ratios

Monte-Carlo:
Estimating Averages

Monte-Carlo Algorithms

Estimating Averages

- In a simple Monte-Carlo problem, we want to estimate the expected value of a random variable Y; i.e., $\mu = \mathbb{E}(Y)$.
- To achieve that, we draw n independently and random samples, $Y_1, \ldots, Y_n$, that have Y distribution.
- Finally, we average them:

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

where $\hat{\mu}_n$ is our estimate of μ.

Monte-Carlo Algorithms

Estimating Averages

- Typically, we have that:

$$Y = f(\mathbf{x}),$$

where $\mathbf{x} \in \mathscr{D} \subset \mathbb{R}^d$ that has a PDF $p(\mathbf{x})$.

- Therefore:

$$\mathbb{E}(Y) = \mu = \int_{\mathscr{D}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}.$$

Monte-Carlo Algorithms

Estimating Averages: Law of Large Numbers

- Let's assume that $\mu = \mathbb{E}(Y)$ for Y exists, and we have i.i.d. samples $Y_1, \ldots, Y_n$ drawn according to Y distribution. The weak law of large numbers tells us:

$$\lim_{n \to \infty} P\left(|\hat{\mu}_n - \mu| \leq \epsilon \right) = 1 \qquad \forall \epsilon > 0.$$

- More interesting is the strong laws:

$$P\left(\lim_{n \to \infty} |\hat{\mu}_n - \mu| = 0 \right) = 1$$

- **Here the error will get below ϵ.**

Monte-Carlo Algorithms

Estimating Averages: Law of Large Numbers

- Let's suppose that:

$$Var(Y) = \sigma^2 < \infty.$$

- Note that $\hat{\mu}_n$ is a random variable with mean:

$$\mathbb{E}(\hat{\mu}_n) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(Y_i) = \mu.$$

- and variance:

$$Var(\hat{\mu}_n) = \mathbb{E}\left((\hat{\mu}_n - \mu)^2\right) = \frac{\sigma^2}{n}.$$

Monte-Carlo Algorithms

Estimating Averages: Law of Large Numbers

- $\hat{\mu}_n = \mu$ tells us that simple Monte-Carlo is **unbiased**.
- $\mathbb{E}\left((\hat{\mu}_n - \mu)^2\right) = \dfrac{\sigma^2}{n}$ tells us another interesting thing:
  - The root mean squared error (RMSE) of $\hat{\mu}_n$ is:

$$\sqrt{\mathbb{E}((\hat{\mu}_n - \mu)^2)} = \frac{\sigma}{\sqrt{n}}.$$

    - This means that if we want to improve our estimate by one more decimal (i.e., 1/10) we need a 100-fold more samples!

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- Typically, we can have a rough idea of the error:

$$\hat{\mu}_n - \mu.$$

- Note that the average squared error is:

$$\frac{\sigma^2}{n}.$$

- It is rare to know σ^2, but we use estimates of it:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{\mu}_n)^2 \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\mu}_n)^2.$$

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- So, if we use s^2 ($\mathbb{E}(s^2) = \sigma^2$), the error is on the order of:

$$\frac{s}{\sqrt{n}}.$$

- From the Center Limit Theorem (CLT), we know that $\hat{\mu}_n - \mu$ has more or less a normal distribution with mean 0 and variance σ^2/n.

- Normal distribution for a variable $X \sim \mathcal{N}(0,1)$:

$$\phi(x) = \frac{e^{-\frac{1}{2}z^2}}{\sqrt{2\pi}} \qquad \Phi(y) = \int_{-\infty}^{y} \phi(x)dx.$$

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- CLT: given $X_1, \ldots, X_n$ i.i.d. random variable with mean μ and finite variance $\sigma^2 > 0$, where $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Then, we have:

$$\forall_{x\in\mathbb{R}} \quad P\left(\frac{\sqrt{n}}{\sigma}\left(\hat{\mu}_n - \mu\right) \leq x\right) \to \Phi(x),$$

  as $n \to \infty$.

- Note, we can change s with σ for $n \to \infty$, and we obtain:

$$P\left(\frac{\sqrt{n}}{s}\left(\hat{\mu}_n - \mu\right) \leq x\right) \to \Phi(x).$$

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- So we have:

$$P\left(\frac{\sqrt{n}}{s}\left(\hat{\mu}_n - \mu\right) \leq x\right) \to \Phi(x)$$

- At this point, we set $\Delta = x$, and we move things around obtaining:

$$P\left(|\hat{\mu}_n - \mu| \geq \frac{\Delta s}{\sqrt{n}}\right)$$

- We find for $\Delta > 0$ that:

$$P\left(|\hat{\mu}_n - \mu| \geq \frac{\Delta s}{\sqrt{n}}\right) = P\left(\sqrt{n}\frac{\hat{\mu}_n - \mu}{s} \leq -\Delta\right) + P\left(\sqrt{n}\frac{\hat{\mu}_n - \mu}{s} \geq \Delta\right) \to \Phi(-\Delta) + \big(1 - \Phi(\Delta)\big)$$

- By symmetry of $\mathcal{N}(0,1)$:

$$\Phi(-\Delta) + \big(1 - \Phi(\Delta)\big) = 2\Phi(-\Delta).$$

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- Assuming a 99% of coverage, we have that:

$$2\Phi(-\Delta) = 1 - 0.99 = 0.01 \rightarrow \Phi(-\Delta) = 0.005.$$

- Finally:

$$\Delta = -\Phi^{-1}(0.005) = \Phi^{-1}(0.995) = 2.58.$$

- Therefore, a 99% confidence interval for μ is computed as:

$$\left[\hat{\mu}_n - 2.58\frac{s}{\sqrt{n}}, \hat{\mu}_n + 2.58\frac{s}{\sqrt{n}}\right].$$

- This leads to $\hat{\mu}_n \pm 2.58\frac{s}{\sqrt{n}}$.

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- Note that s requires a two-pass algorithm that is not very ideal; i.e., we need to store samples!
- A solution would be to compute it as:

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n} y_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} y_i\right)^2,$$

but this version is not numerically stable.

Monte-Carlo Algorithms

Estimating Averages: Error Estimation

- There are other two popular solutions. The first one:

$$\delta_i = y_i - \hat{\mu}_{i-1} \qquad \hat{\mu}_i = \hat{\mu}_{i-1} + \frac{1}{i}\delta_i \qquad S_i = S_{i-1} + \frac{i-1}{i}\delta_i^2.$$

  where $\hat{\mu}_1 = y_1$ and $S_1 = 0$.

- The other option is:

$$\tilde{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{\frac{n}{2}} \left(x_{2i} - x_{2i-1}\right)^2,$$ which works well for a large n.

Monte-Carlo Algorithms

Estimating Averages: How Many Samples?

- If we know $Var(Y) = \sigma_0^2$, we can say something about n.
- Given a random variable X, Chebychev's inequality tells us:

$$P(|X - \mathbb{E}(X)| \geq \epsilon) \leq \frac{Var(X)}{\epsilon^2}, \text{ for } \epsilon > 0.$$

- In our case, $X = \hat{\mu}_n$, $\mathbb{E}(X) = \mu$, and $Var(\hat{\mu}_n) = \sigma_0^2/n$:

$$P(|\hat{\mu}_n - \mu| \geq \epsilon) \leq \frac{\sigma_0^2}{n}\frac{1}{\epsilon^2}$$

Monte-Carlo Algorithms

Estimating Averages: How Many Samples?

- So if at confidence level α:

$$P(|\hat{\mu}_n - \mu| \geq \epsilon) \leq \frac{\sigma_0^2}{n}\frac{1}{\epsilon^2} = 1 - \alpha,$$

- Solving for n, we obtain:

$$n \geq \frac{\sigma_0^2}{\epsilon^2}\frac{1}{1-\alpha}.$$

Monte-Carlo:
Estimating Quantiles

Monte-Carlo Algorithms

Estimating Averages: Quantiles

- Given a random variable X, the β quantile is defined as:

$$P(X \leq Q^{\beta}) = \beta.$$

- To estimate Q^{β} with $\beta \in [0,1]$, we use the corresponding quantile of the sample.
- We draw sample, $X_1, \ldots, X_n$, from X, and then these are sorted. Obtaining:

$$X_{s(1)}, \ldots, X_{s(n)}.$$

- The quantile estimation is given by:

$$\hat{Q}_n^{\beta} = X_{s(\lceil \alpha n \rceil)}.$$

Monte-Carlo Algorithms

Estimating Averages: Quantiles

- When we estimate quantiles, we need to generate at least:

$$n > \frac{1}{\min(\beta, 1 - \beta)} \text{ samples,}$$

otherwise $\hat{Q}_n^\beta = X_{s(1)}$ or $\hat{Q}_n^\beta = X_{s(n)}$.

Monte-Carlo Algorithms

Estimating Averages: Quantiles

- In this case, the 99%, $\alpha = 0.01$, confidence interval is:

$$\left[Y_{s(L)}, Y_{s(R)}\right];$$

where:

$$L = \max\left[l \in \{0,\ldots,n+1\} \,\middle|\, \sum_{x=0}^{l-1} \binom{n}{x} \theta^x (1-\theta)^{n-x} \geq \frac{\alpha}{2}\right] \text{and,}$$

$$R = \min\left[r \in \{0,\ldots,n+1\} \,\middle|\, \sum_{x=r}^{n} \binom{n}{x} \theta^x (1-\theta)^{n-x} \geq \frac{\alpha}{2}\right].$$

Monte-Carlo:
Estimating Ratios

Monte-Carlo Algorithms

Estimating Averages: Ratios

- Given two random variables X and Y, we would like to compute their ratio:

$$\theta = \frac{\mathbb{E}(X)}{\mathbb{E}(Y)}.$$

- To estimate θ, we draw n independent pairs (X_i, Y_i) from the target distribution, and we compute the ratio as:

$$\hat{\theta}_n = \frac{\hat{X}_n}{\hat{Y}_n},$$

where:

$$\hat{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \hat{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

Monte-Carlo Algorithms

Estimating Averages: Ratios

- In this case, the 99% confidence interval is:

$$\hat{\theta} \pm 2.58\sqrt{\hat{Var}(\hat{\theta})};$$

where:

$$\hat{Var}(\hat{\theta}) = \frac{1}{n^2\hat{X}^2}\sum_{i=1}^{n}\left(Y_i - \hat{\theta}X_i\right)^2.$$

Monte-Carlo:
Failure

Monte-Carlo Algorithms

When MC fails

- Monte-Carlo methods are typically robust; but it can fail:
  - We may have a failure when $\mu = \mathbb{E}(X)$ does not exist. Its existence is linked to:

$$\mathbb{E}(X) < \infty.$$

  - We may have a failure when μ is finite, $\mathbb{E}(X) < \infty$, but the variance is infinite; i.e., $Var(X) = \infty$:
    - The Law of Large Numbers still converge!
    - We lose the rate $O\left(n^{-\frac{1}{2}}\right)$ and the CLT's confidence intervals.

Monte-Carlo Algorithms

When MC Fails: Saint Petersburg Lottery

- A fair coin will be flipped until tails appear for the first time.
- $X = x$ is the total number of flips.
- If $X = x$ then you will get 2^x euros.
- For independent coin flips $\forall_{i>0} P\left(X = i\right) = 2^{-i}$. Therefore, the expected pay off is:

$$\mu = \sum_{i=1}^{\infty} P(X = i) \cdot 2^i = \sum_{i=1}^{\infty} 2^{-i} \cdot 2^i = \sum_{i=1}^{\infty} 1 = \infty.$$

Monte-Carlo Algorithms

When MC Fails: Saint Petersburg Lottery

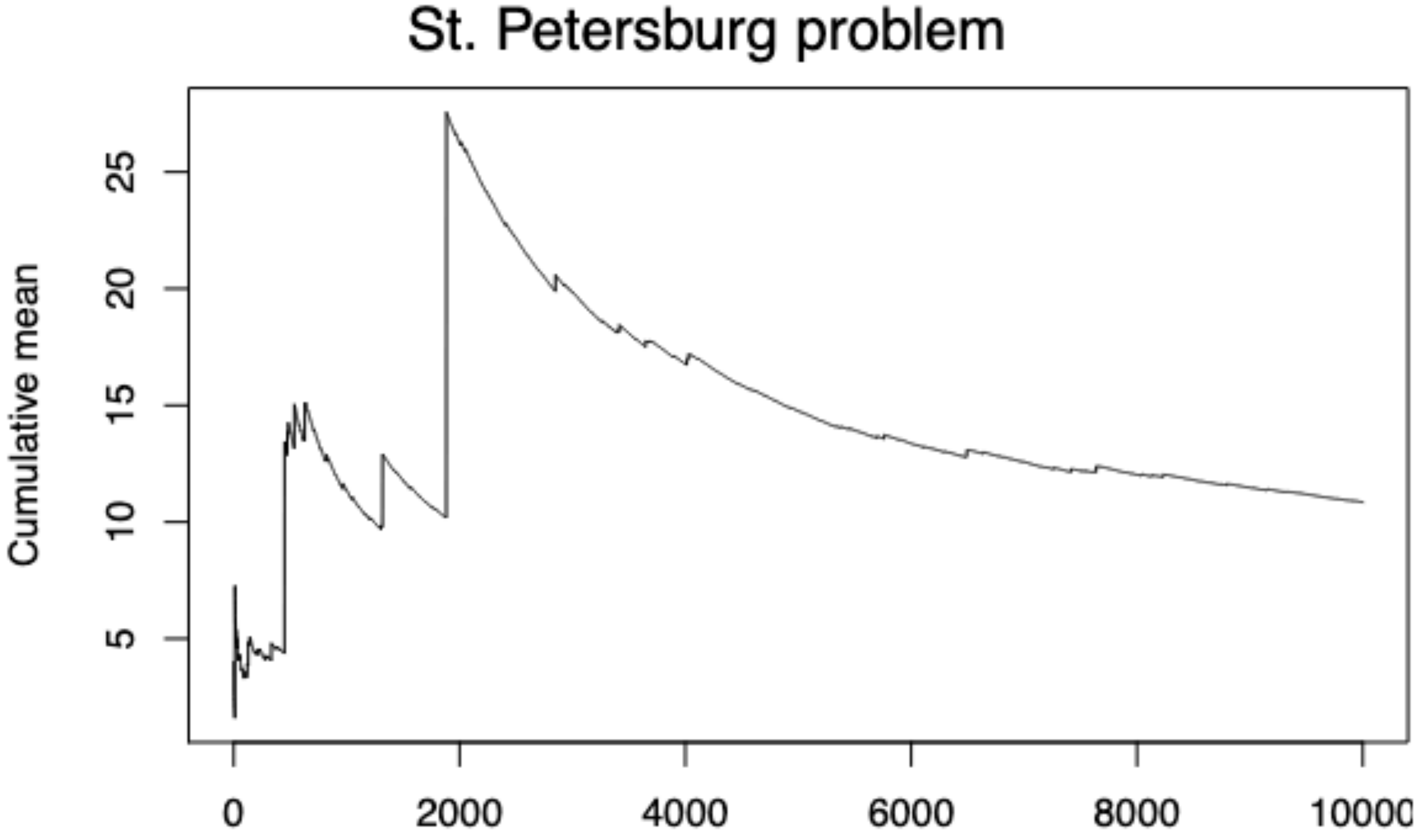


Image from Art Owen book - Chapter 2.

Monte-Carlo Algorithms

When MC Fails: Long Lived Comets

- Hammersley and Handscomb proposed how to calculate the lifetime of a long lived comet.

- A comet has an energy level x_e:

  - if $x_e > 0$ it leaves the solar system.

  - Otherwise, the comet completes an orbit in $(-x_e)^{-\frac{3}{2}}$ time.

  - x_e varies when the comet interacts with planets:

    - Model: $x_e + Z \qquad Z \sim \mathcal{N}(0,\sigma^2)$

Monte-Carlo Algorithms

When MC Fails: Long Lived Comets

- How long does the comet stay in the solar system?

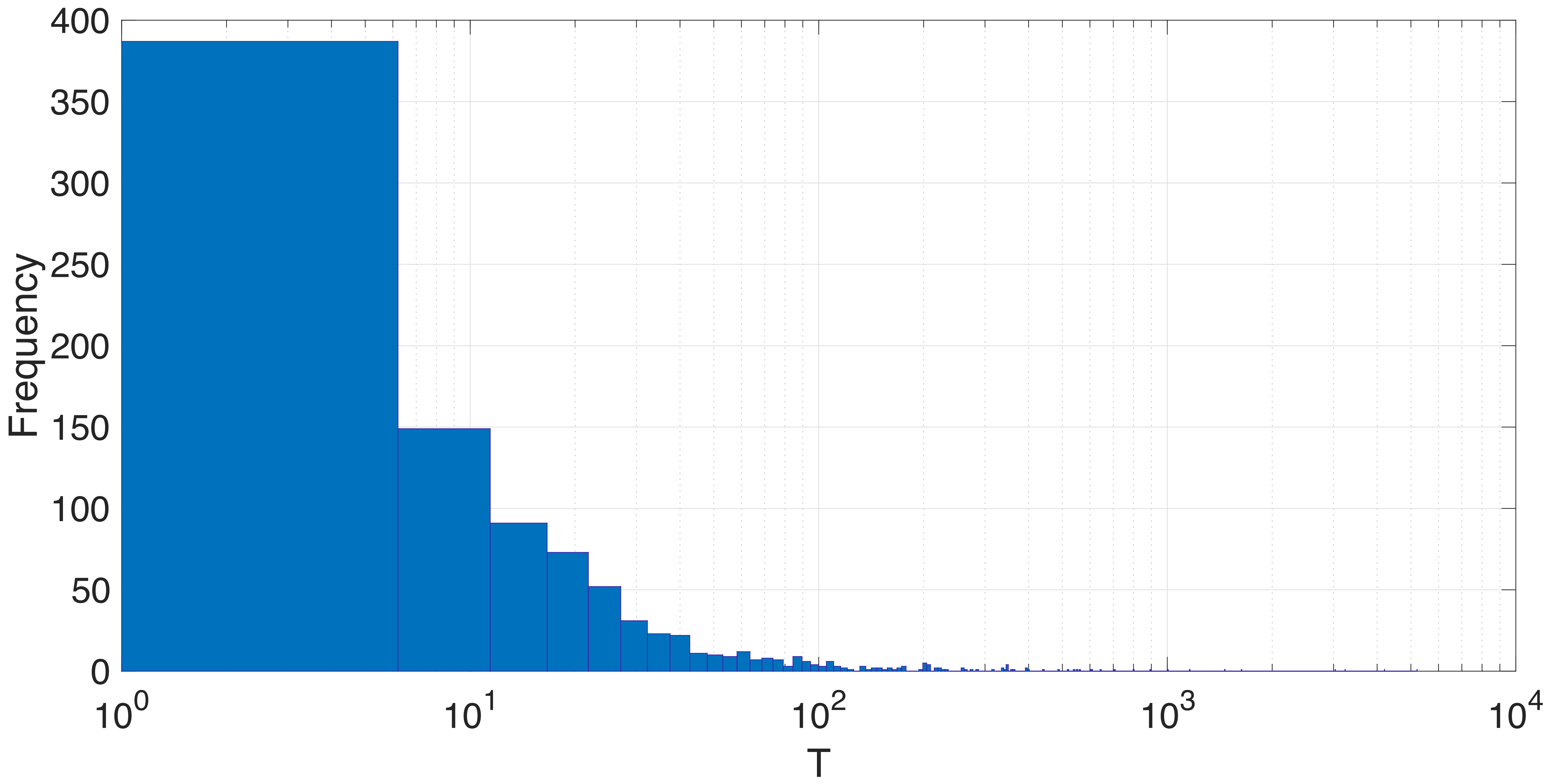
$$T = \sum_{i=1}^{n} (-x_i)^{-\frac{3}{2}} \qquad x_{i+1} = x_i + z_i.$$

- n is random itself —> difficult to study this analytically!

Monte-Carlo Algorithms

When MC Fails: Long Lived Comets

- Hammersely showed that $P(T > t) \propto t^{-\frac{3}{2}}$ for large t; so $f_T(t) \propto t^{-\frac{5}{3}}$. This means that $\mathbb{E}(T) = \mathbb{E}(T^2) = \infty$, and so the variance is infinite!



Monte-Carlo:
A Final Note

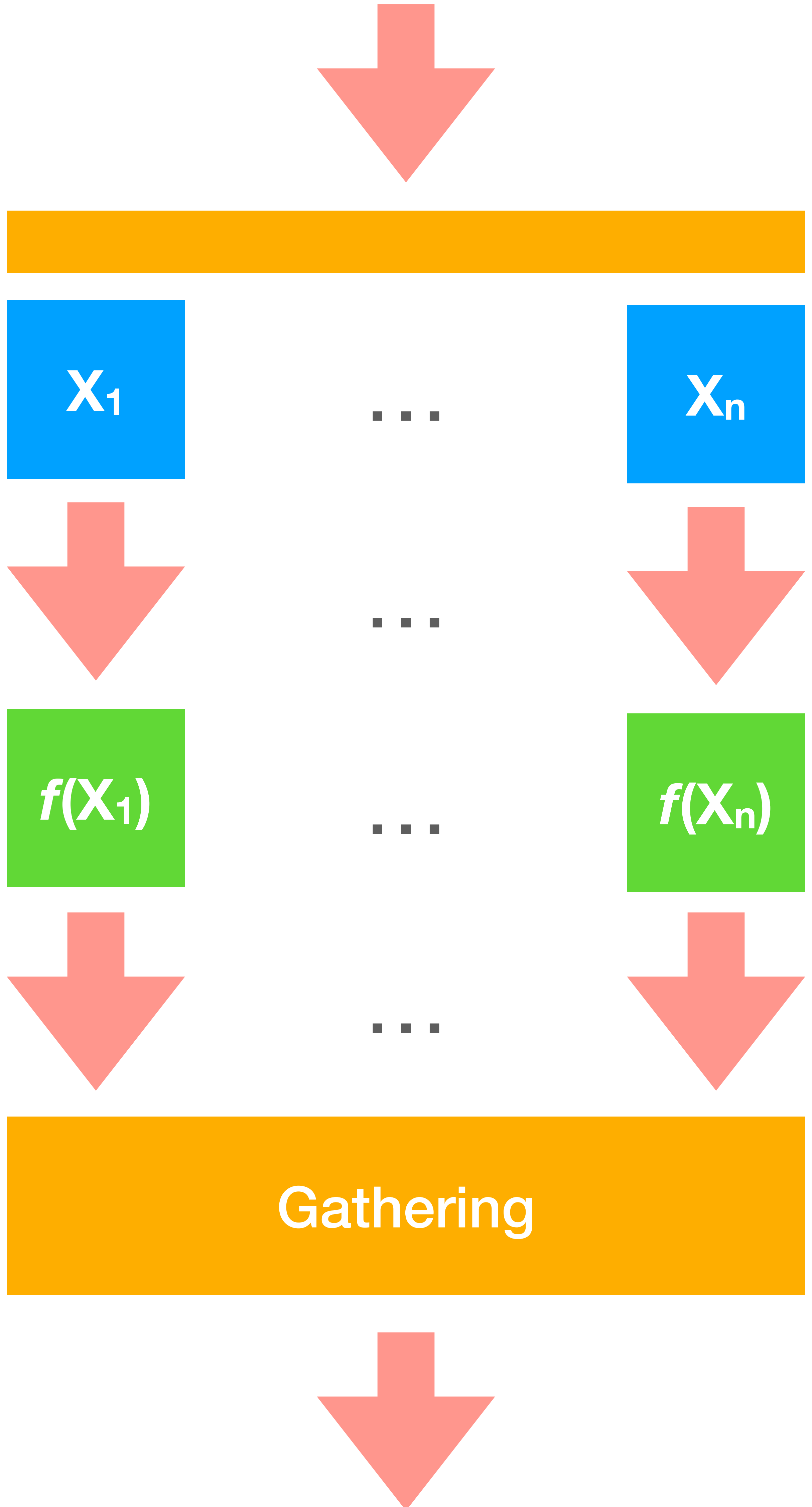
Monte-Carlo Algorithms

A Final Note

- In this process, we draw **independently** samples that are distributed with a given PDF.
- The fact that samples are independent is extremely important:
  - We can generate samples in parallel on different threads, cores, CPUs, and machines.
  - This means that Monte-Carlo algorithms are massively parallel.

Monte-Carlo Algorithms

A Final Note



Bibliography

- Art Owen. “Chapter 1: Introduction” from the book “Monte Carlo theory, methods and examples”. 2013.
- Art Owen. “Chapter 2: Simple Monte Carlo” from the book “Monte Carlo theory, methods and examples”. 2013.
- Peter Shirley, Changyaw Wang, Kurt Zimmerman. “Monte Carlo Techniques for Direct Lighting Calculations”. ACM Transactions on Graphics. Volume 15. Issue 1. Jan. 1996.

Thank you for your attention!