Monte Carlo

Goal: compute $\alpha = EX$

Settings:

$X_1, \cdots, X_n$

$\alpha_n = \bar{X}_n$

rate of convergence as a function of c

$\alpha(c) = \bar{X}_{N(c)}$

$$c^{\frac{1}{2}} (\alpha(c) - \alpha) \Rightarrow \lambda^{-\frac{1}{2}} \sigma N(0,1)$$

$$\lambda^{-1} \sigma^2 = E\tau \cdot \mathrm{Var} X$$

- $\tilde{S}_n = \sum_{i=1}^{n} \tilde{X}_i$
- $\tilde{X}_i = X_i - EX$

$$c^{\frac{1}{2}} \left(\frac{S_{N(c)}}{N(c)} - \alpha \right) = c^{\frac{1}{2}} \left(\frac{S_{N(c)} - N(c)\alpha}{N(c)} \right)$$

$$= c^{\frac{1}{2}} \frac{\tilde{S}_{N(c)}}{c} \underbrace{\left(\frac{c}{N(c)} \right)}_{\downarrow \text{a.s.}\ E\tau}$$

$$= \frac{\tilde{S}_{N(c)}}{\sqrt{c}}$$

$$\frac{N(c)}{c} \xrightarrow{a.s.} \lambda$$

$$\frac{\tilde{S}_{N(c)}}{\sqrt{c}} \overset{?}{\approx} \frac{\tilde{S}_{\lfloor \lambda c \rfloor}}{\sqrt{c}}$$

$$\frac{\tilde{S}_{N(c)} - S_{\lfloor \lambda c \rfloor}}{\sqrt{c}} \xrightarrow{P} 0 \qquad \text{Anscombe's Theorem}$$

On $\{ |N(c) - \lambda c| \le \varepsilon c \}$

prob converges to 1

$$\frac{|\tilde{S}_{\lambda(c)} - S_{\lfloor \lambda c \rfloor}|}{\sqrt{c}} \le \frac{\max_{|j| \le \varepsilon c} |\tilde{S}_j - \tilde{S}_{\lfloor \lambda c \rfloor}|}{\sqrt{c}} \qquad \text{Kolmogorov maximal inequality}$$

Uniform Random Generator

$U_1, U_2, \cdots$ i.i.d. uniform on $[0,1]$

$\downarrow$

non-uniform r.v's

$\downarrow$

model randomness needed

Uniform R.N.G

1) physical r.n.g.

   e.g. coin flips

   1, 0, 0, 1, $\cdots$

   bits : $\sum_{i=1}^{64} z_i 2^{-i}$

   - expensive
   - slow
   - biased
   - correlated

   e.g. charged particles

   follows a Poisson distribution

   - need to estimate λ
   - lockout time

2) deterministic algorithm that produces unpredictable outcomes

• Midsquare Method

$$\begin{array}{r} \times\times\times\times \\ \times\times\times\times \\ \hline 0\ 0\ \underbrace{0\ 0\ 0\ 0}\ 0\ 0 \end{array}$$

"trap state" : 0000

- Full - Period Generator

X_n = "large random integers" $\in \{0, 1, \cdots, n-1\}$

$U_n = X_n / n$

often 0 ; large

$$X_{n+1} = (a X_n + b) \bmod n$$

Full period (hit $1 - n-1$) ← cyclic

$\begin{cases} n \text{ prime} & \text{e.g. } 2^{31} - 1 \quad (\sim 2 \text{ billion}) \\ a & \end{cases}$

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

uniform → non-uniform

e.g. Exponential

$$F(x) = 1 - e^{-\lambda x}$$

$$F^{-1}(u) = -\frac{1}{\lambda} \log(1-u) \overset{D}{=} \mathrm{Exp}(\lambda)$$

smallest $\frac{1}{2 \text{ billion}}$ ⇒ λ largest 30

- Tousworthe

  Masense Twitter

  10^{120}

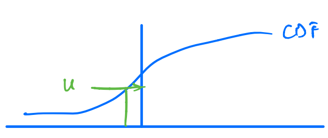
Statistical tests for random generators

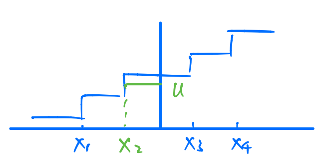
Non - Uniform Generator

- Inversion

  $X_1, X_2, \cdots$ dist F

  $X_i = F^{-1}(U_i)$





- Acceptance - Rejection

$P(\cdot)$ ← want algo

$Q(\cdot)$ ← good algo for generating outcomes

$$P(dw) = Q(dw)\, L(w)$$

$$f(x) = g(x) \left(\frac{f(x)}{g(x)} \right) \qquad \sup \frac{f(x)}{g(x)} < c < \infty$$

Suppose $L(w) \le c < \infty, \quad \forall\, w \in \Omega$

$$P(dw) \propto Q(dw) \underbrace{\frac{L(w)}{c}}_{\in [0,1]}$$

$$= Q(dw)\, P\left(u \le \frac{L}{c}\right)$$

Steps:

1. Generate w under Q
2. Generate an independent u and test

$$u \le \frac{L(w)}{c}$$

- if yes, return w (Accept)
- if no, return to 1 (Reject)

Statistics
Machine Learning
Data Science

- $X_1, X_2, \cdots, X_n$ i.i.d. F

sample from an underlying population

↓

Model-free analysis "non-parametric"

- Parametric stat. modeling

$\swarrow$ Frequentist $\searrow$ Bayesian

- Non-parametric