Lecture 4: Splitting and Killing

- ☐ Matrix inversion by Monte Carlo
- ☐ Self-avoiding walks
- ☐ Resampling
- ☐ Rare event sampling

Matrix inversion by Monte Carlo

- Consider linear system $Ax=b$, $A \in \mathbb{R}^{N\times N}$, $b \in \mathbb{R}^N$.
- Let's solve it using MC sampling.
- Write $A = I - G$, $x = (I-G)^{-1} b = \sum_{t=0}^{\infty} G^t b$, and assume $\|G\|_1 = \max_{1 \le j \le N} \sum_{i=1}^{N} |G_{ij}| < 1$.

Matrix inversion algorithm

Input: matrix $G \in \mathbb{R}^{N\times N}$, vector $b \in \mathbb{R}^N$.

Output: Approximate solution to $(I-G)x = b$.

1. Sample $X_0 \in \{1, \dots, N\}$ with $\mathbb{P}\{X_0 = i\} = p(i) = \dfrac{|b_i|}{\|b\|_1}$ and set $w_0 = b_i / p(i)$.
2. For $t = 1, 2, 3, \dots$
   a) with probability $1 - \sum_{i=1}^{N} |G_{i X_{t-1}}|$, set $t = \tau$ and STOP.
   b) otherwise, sample $X_t \in \{1, \dots, N\}$ with $\mathbb{P}\{X_t = i \mid X_{t-1}\} = p(X_{t-1}, i) = |G_{i, X_{t-1}}|$ and set $w_t = w_{t-1} \cdot G_{i, X_{t-1}} / p(X_{t-1}, i)$
3. Return estimate $\hat{x} = \sum_{t=0}^{\tau-1} w_t e_t$.

should be e_{X_t}

Proposition Matrix inversion algorithm is unbiased

$\mathbb{E}\hat{x} = x = (I-G)^{-1}b$ and $\mathbb{E}\|\hat{x}-x\|^2 \le \|b\|_1^2\, \mathbb{E}\tau^2$

Pf $\mathbb{E}\hat{x} = \mathbb{E}\left[\sum_{t=0}^{\infty} w_t e_t \mathbb{1}\{\tau > t\}\right]$

$$= \sum_{t=0}^{\infty} \sum_{x_0,\ldots,x_t} p(x_0,x_1,\ldots,x_t) \cdot \frac{b_{x_0}}{p(x_0)} \prod_{s=1}^{t} \frac{G_{x_s,x_{s-1}}}{p(x_{s-1},x_s)} e_{x_t}$$

$$= \sum_{t=0}^{\infty} \sum_{x_0,\ldots,x_t} e_{x_t} \prod_{s=1}^{t} G_{x_s,x_{s-1}} b_{x_0}$$

$$= \sum_{t=0}^{\infty} G^t b.$$

Next bound $\mathbb{E}\|x-\hat{x}\|^2 \overset{\text{unbiasedness}}{=} \mathbb{E}\|\hat{x}\|^2 - \|x\|^2 \le \mathbb{E}\|\hat{x}\|^2$

$\le \mathbb{E}\|\hat{x}\|_1^2 \underset{\substack{\uparrow \\ \text{each summand} \le \|b\|_1}}{\le} \mathbb{E}\left|\tau\|b\|_1\right|^2 = \|b\|_1^2\, \mathbb{E}\tau^2.$ □

- Matrix inversion algorithm is not terribly accurate
- But, we can average independent estimates

$$\mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} \hat{x}_i\right] = \frac{1}{N}\mathrm{Var}\left[\hat{x}_1\right] \to 0 \quad \text{as } N \to \infty.$$

- We never look at the full matrix, we just look at τ columns $\Rightarrow$ efficient for $N \ge 10^6$.
- There is an extension to column diagonally dominant A (use diagonal as right preconditioner).
- Invented by Ulam, Von Neumann, Wasow in late 1940s – early 1950s.

Ex: PageRank

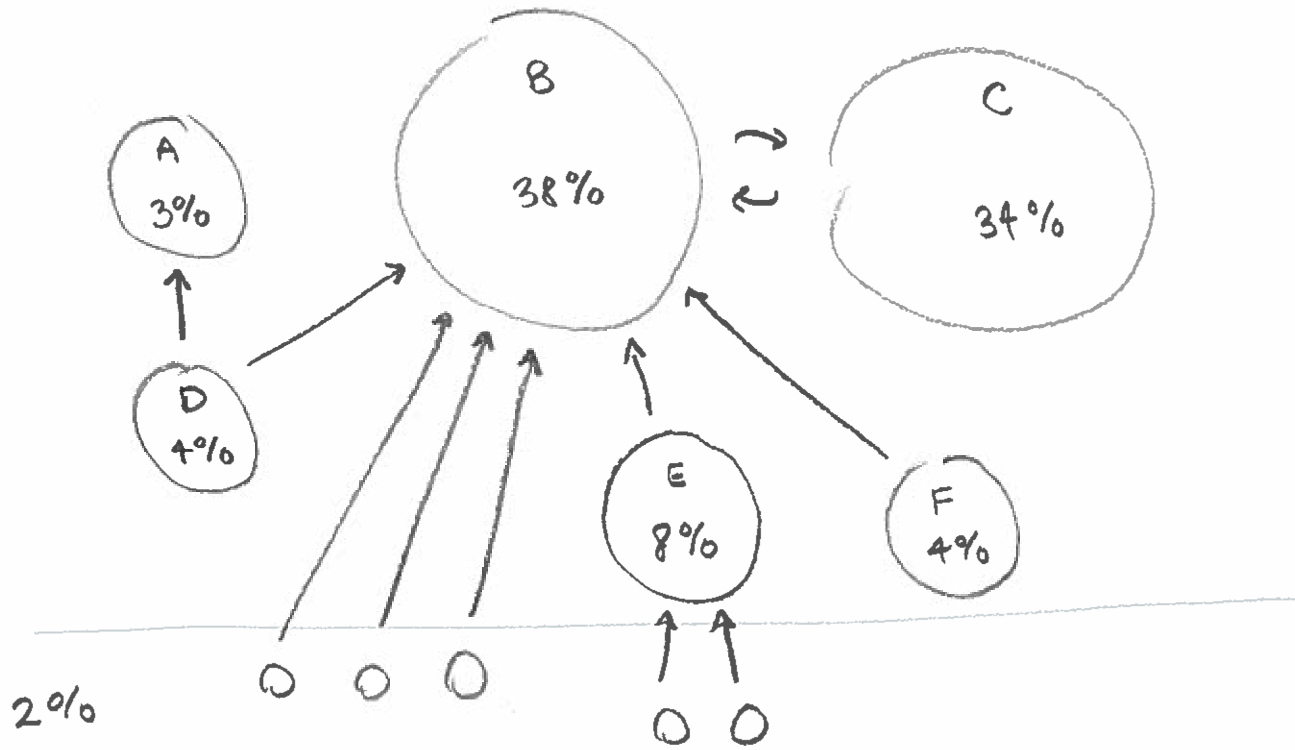
- Given a network of websites $\{1, \dots, N\}$, a restless web surfer picks one at random with probability $(v_i)_{1 \le i \le N}$.
- If the website has $d_j > 0$ outgoing links, with probability α the web surfer picks a link at random. Otherwise, she gets bored, abandons the chain of websites, and picks a fresh starting point with probability $(v_i)_{1 \le i \le N}$
- PageRank problem what is the long-run distribution of websites visited?
- Linear algebra solution

Set $P_{ij} = \begin{cases} 1/d_j, & \text{if website } j \text{ has } d_j > 0 \text{ outgoing links, including a link to } i; \\ v_i, & \text{if website } j \text{ has } d_j = 0 \text{ outgoing links,} \\ 0, & \text{otherwise.} \end{cases}$

$$\Rightarrow \vec{x} = \alpha P \vec{x} + (1-\alpha)\vec{v}$$

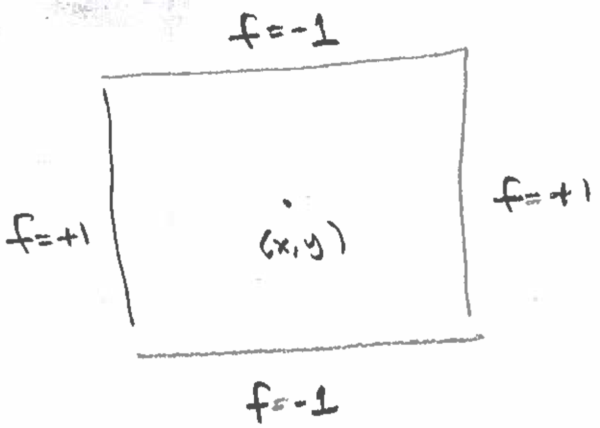
$$\Rightarrow (I - \alpha P)\vec{x} = (1-\alpha)\vec{v}$$

- Used to rank website "importance" by Google, with $\alpha = .85$ and $v_i = \frac{1}{N}$, $1 \le i \le N$.



Ex: Finite differences

$$\begin{cases} -\frac{1}{2}\left(\partial^2_{xx} + \partial^2_{yy}\right) u(x,y) = 0, & (x,y) \in E = [-K, K]^2 \\ u(x,y) = f(x,y), & (x,y) \in \partial E \end{cases}$$



Discretize using $(x,y) \in \{-K, -K+1, \dots, K-1, K\}^2$

$$L_{(x,y),(x',y')} = \begin{cases} -1/4, & |x-x'| + |y-y'| = 1, \\ 1, & |x-x'| + |y-y'| = 0, \\ 0, & |x-x'| + |y-y'| \geq 2. \end{cases}$$

$$\Rightarrow \begin{cases} (Lu)_{(x,y)} = 0, & \max\{|x|,|y|\} < K, \\ u_{(x,y)} = f_{(x,y)}, & \max\{|x|,|y|\} = K. \end{cases}$$

$$\Rightarrow (I-P)\vec{u} = \vec{f}$$

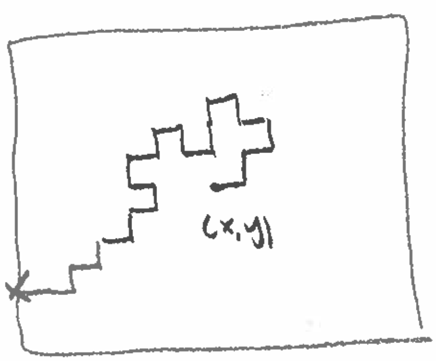
where $$P_{(x,y),(x',y')} = \begin{cases} 1/4, & |x-x'|+|y-y'| = 1, \ \max\{|x|,|y|\}<K, \\ 0, & \text{otherwise.} \end{cases}$$

<u>Option 1</u> Solve for $\vec{u} = (I-P)^{-1}\vec{f}$ using MC :(

<u>Option 2</u> Solve for $u_{(x,y)} = \vec{e}_{(x,y)}^{\,T}(I-P)^{-1}\vec{f}$ using MC :)

a) solve for $(I-P^T)^{-1}\vec{e}_{(x,y)}$ using MC.

b) Calculate $\langle (I-P^T)^{-1}\vec{e}_{(x,y)}, \vec{f} \rangle$ (possibly for many different vectors $\vec{f}$)

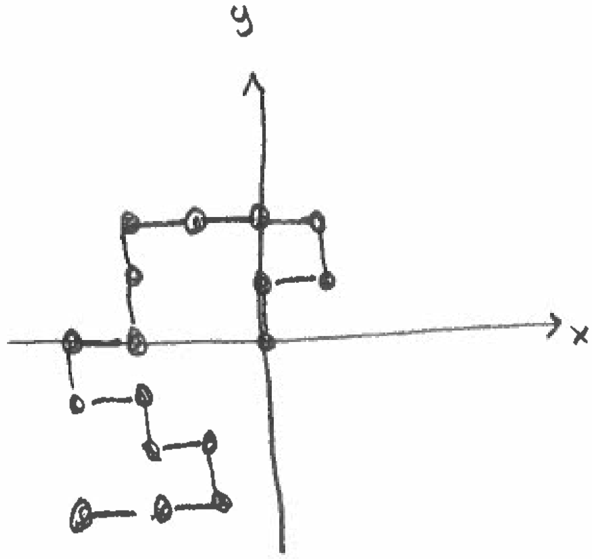


$$u_{(x,y)} = \mathbb{E}_{Z_0=(x,y)}\left[f(Z_\tau) \right]$$

random walk, stopped at boundary

"Feynman-Kac formula"

<u>Self-avoiding walk (SAW)</u> $X = (\vec{X}_0, \vec{X}_1, \ldots, \vec{X}_T)$ where $\vec{X}_0 = (0,0)$, each $\vec{X}_t = (a,b)$ for integers a, b, $\|\vec{X}_t - \vec{X}_{t+1}\| = 1$, and all the $\vec{X}_t$ are <u>distinct</u>.



Goal 1: Count

$Z_T = \#\{\text{SAWs of length } T\}$

Goal 2: Sample

$X \sim \text{Unif}\{\text{SAWs of length } T\}$

Dumb method:

1. Set $\vec{X}_0 = (0,0)$ and sample $\vec{X}_1 \in \text{Unif}\{(+1,0),(-1,0),(0,+1),(0,-1)\}$
2. For each $t = 1, 2, \ldots, T-1$, continue straight with prob $1/3$, turn right with prob $1/3$, and turn left with prob $1/3$.
3. If the RW self-intersects, call it a failure.

Rosenbluth & Rosenbluth method:

1. Set $\vec{X}_0 = (0,0)$ and $w_0 = 1$.
2. For each $t = 0, 1, \ldots, T-1$, let $n_t = \#$ open vertices. If $n_t > 0$, move to a random open vertex and set $w_{t+1} = w_t \cdot n_t$. If $n_t = 0$, call it a failure.

Prop $\mathbb{E}\left[\mathbb{1}_{\text{success}}\, w_T\, f(X)\right] = \sum_{y \in \{\text{length } T \text{ SAWs}\}} f(y)$ for any function f.

⑦

Pf $\mathbb{E}\left[\mathbb{1}_{\text{success}}\, w_T f(x)\right] = \sum_{y \in \{\text{length } T \text{ SAWs}\}} p(y) \prod_{t=0}^{T-1} n_t f(y)$

$$= \sum_{y \in \{\text{length } T \text{ SAWs}\}} \prod_{t=0}^{T-1} \frac{1}{n_t} \prod_{t=0}^{T-1} n_t f(y)$$

$$= \sum_{y \in \{\text{length } T \text{ SAWs}\}} f(y). \qquad \square$$

$\Rightarrow$ In particular, we have an unbiased estimator

$$Z_T = \#\{\text{SAWs of length } T\} = \mathbb{E}\left[\mathbb{1}_{\text{success}}\, w_T\right]$$

Resampling Let's say we have a weighted ensemble $(w^i, X^i)_{1 \le i \le N}$.

- We can split X^i into K "children" particles and assign them weights w^i/K. This only improves the variance.
- Conversely, we can randomly kill $K-1$ out of K particles with equal weights w^i and assign the survivor the sum of the weights Kw^i.
- Resampling is the most extreme splitting / killing approach: we replace $(w^i, X^i)_{1 \le i \le N}$ with a new ensemble $(\hat{w}^i, \hat{X}^i)_{1 \le i \le N}$ of equally weighted particles.

Bootstrap resampling

Input: Unevenly weighted ensemble $(w^i, X^i)_{1\le i\le N}$.

Output: Evenly weighted ensemble $(\hat{w}^i, \hat{X}^i)_{1\le i\le N}$ s.t.

$$\mathbb{E}\left[\sum_{i=1}^N \hat{w}^i f(\hat{X}^i)\right] = \sum_{i=1}^N w^i f(X^i) \quad \text{for all functions } f.$$

1. Set $\bar{w} = \frac{1}{N}\sum_{i=1}^N w^i$.

2. For $i = 1, 2, \ldots, N$:

   a) Independently sample $\hat{X}^i \in \{X^1, \ldots, X^N\}$ with
   $$\mathbb{P}\{\hat{X}^i = X^j\} = w^j \Big/ \sum_{k=1}^N w^k.$$

   b) Set $\hat{w}^i = \bar{w}$.

$\Rightarrow$ Number of copies $\vec{C} = (C^1, \ldots, C^N)$ for each particle X^i has distribution $\vec{C} \sim \mathrm{Mult}\left(N, \frac{w^1}{\sum_{k=1}^N w^k}, \ldots, \frac{w^N}{\sum_{k=1}^N w^k}\right)$.

Q: Is this optimal?

Improved (pivotal) resampling

1. Set $\bar{w} = \frac{1}{N}\sum_{i=1}^N w^i$, $\vec{C} = \left(\frac{w^1}{\bar{w}}, \frac{w^2}{\bar{w}}, \ldots, \frac{w^N}{\bar{w}}\right)$,

   $s = 1$, $e = 2$.

2. For $i = 1, 2, \ldots, N-1$:

   a) Let $a = C^s - \lfloor C^s \rfloor$ and $b = C^e - \lfloor C^e \rfloor$ be the non-integer parts of C^s and C^e.

   b) If $a + b \le 1$:

i. Set $C^s = C^s + b$, $C^e = C^e - b$, $e = e+1$ w/ probability $a/(a+b)$

ii. Otherwise, set $C^s = C^s - a$, $C^e = C^e + a$, $s = e$, $e = e+1$.

c) If $a+b > 1$:

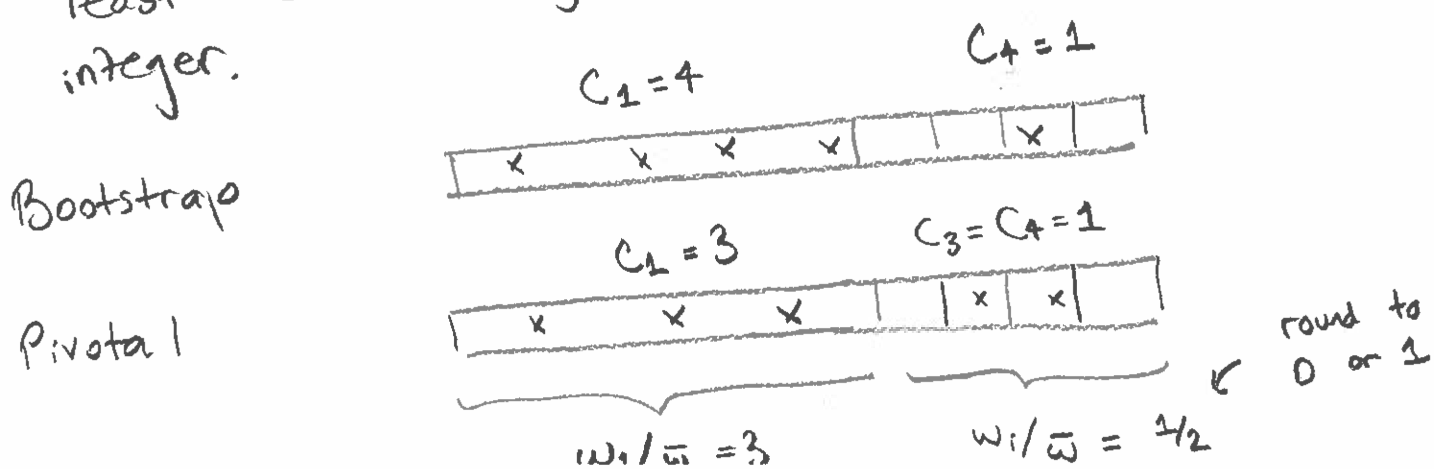
i. Set $C^s = C^s + (1-a)$, $C^e = C^e - (1-a)$, $s = e$, $e = e+1$ w/ probability $(1-b)/(2-a-b)$.

ii. Otherwise, set $C^s = C^s - (1-b)$, $C^e = C^e + (1-b)$, $e = e+1$.

3. Let the weighted ensemble $(\hat{w}^i, \hat{X}^i)_{1 \le i \le N}$ consist of C^j copies of each particle X^j, with all the weights equal to $w^i = \bar{w}$.

$\Rightarrow$ Entries of $\vec{C}$ are negatively correlated with mean $\mathbb{E}[C^i] = w^i/\bar{w}$ and variance $\mathrm{Var}[C^i]$ as low as possible.

$\Rightarrow$ Each step of pivotal resampling rounds at least one entry of $\vec{C}$ to the nearest integer.

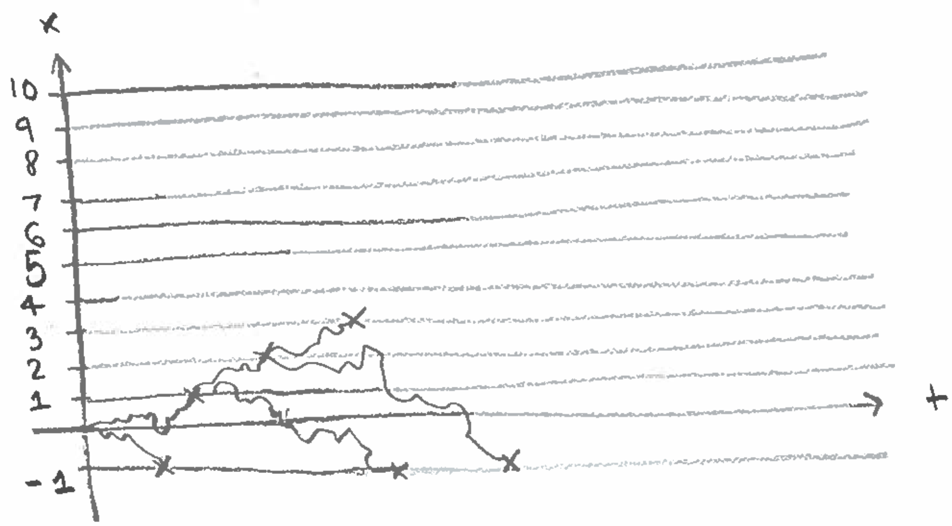


Takeaway: Combining pivotal resampling with the Rosenbluth & Rosenbluth algorithm leads to a powerful sampler for SAWs.

Forward flux sampling

- We want to calculate $\mathbb{P}\{X_t = 10 \text{ before } X_t = -1\}$ where $(X_t)_{t \geq 0}$ is fractional Brownian motion.
- Set $\tau_x = \min\{t \geq 0 : X_t = x\}$ for $x > 0$, $\tau_{-1} = \min\{t \geq 0 : X_t = -1\}$. We want $\mathbb{P}\{\tau_{10} < \tau_{-1}\}$.
- Idea: Use the fact that

$$\mathbb{P}\{\tau_{10} < \tau_{-1}\} = \mathbb{P}\{\tau_1 < \tau_{-1}\} \prod_{s=2}^{10} \mathbb{P}\{\tau_s < \tau_{-1} \mid \tau_{s-1} < \tau_{-1}\}.$$



Forward flux sampling (FFS)

1. Set $x_0^i = 0$, $w_0^i = 1/N$ for $1 \leq i \leq N$.
2. For $s = 1, \dots, 10$:
   a) Simulate each X^i until hitting $X^i \geq s$ or $X^i \leq -1$. In the former case, set $w_s^i = w_{s-1}^i$. In

the latter case, set $w_s^i = 0$.

b) Resample $(w_s^i, X^i)_{1 \le i \le N}$ to obtain an equally weighted ensemble $(\hat{w}_s^i, \hat{X}^i)_{1 \le i \le N}$ and set $(w_s^i, X^i)_{1 \le i \le N} = (\hat{w}_s^i, \hat{X}^i)_{1 \le i \le N}$.

<u>Proposition</u> $\mathbb{E}\left[\sum_{i=1}^{N} w_{10}^i f(X^i) \right] = \mathbb{E}\left[f(X) \mathbb{1}\{\tau_{10} < \tau_{-1}\} \right]$

for all bounded functions f.

$\Rightarrow$ In particular, $\mathbb{P}\{\tau_{10} < \tau_{-1}\} = \mathbb{E}\left[\sum_{i=1}^{N} w_{10}^i \right]$.

Takeaway: Variations on FFS are used all the time for rare event sampling in chemistry.