Lecture 4: Splitting and K读书

- 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 \( \|G\|_1 = \max_{1 \leq j \leq N} \sum_{i=1}^{N} |G_{ij}| < 1 \).

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 \( X_0 \in \{1, \ldots, N\} \) with \( \text{IP}\{X_0 = i\} = \rho(i) = \frac{1}{\|b\|_1} \) and set \( w_0 = b_i / \rho(i) \).

2. For \( t = 1, 2, 3, \ldots \)
   a) With probability \( 1 - \sum_{i=1}^{N} |G_{iX_{t+1}}| \), set \( t = \tau \) and stop.
   b) Otherwise, sample \( X_t \in \{1, \ldots, N\} \) with
      \( \text{IP}\{X_t = i \mid X_{t+1}\} = \rho(X_{t+1}, i) = |G_{iX_{t+1}}| \) and set
      \( w_t = w_{t-1} \cdot G_{iX_{t+1}} / \rho(X_{t-1}, i) \).

3. Return estimate \( \hat{x} = \sum_{t=1}^{\infty} w_t e_t \).
Proposition. Matrix inversion algorithm is unbiased. 

\[ \mathbb{E} \hat{x} = x = (I-G)^{-1}b \text{ and } \mathbb{E} \| \hat{x} - x \| ^2 \leq \| b \| _2^2 \mathbb{E} \xi ^2 \]

Let 

\[ \hat{p} = \mathbb{E} \left[ \sum \frac{b}{p(x_0)} \prod_{s=1}^t \frac{G_{x_s,x_{s+1}}}{p(x_s,x_{s+1})} \epsilon_x \right] \]

\[ = \sum_{x_0} \sum_{x_t} \rho(x_0,x_1,...,x_t) \cdot \frac{b_{x_0}}{\rho(x_0)} \prod_{s=1}^t \frac{G_{x_s,x_{s+1}}}{p(x_s,x_{s+1})} \epsilon_x \]

\[ = \sum_{x_0} \sum_{x_t} \rho(x_0) \prod_{s=1}^t G_{x_s,x_{s+1}} b_{x_0} \]

\[ = \sum_{x_0} \sum G^{+} b \text{ unbiasedness} \]

Next bound 

\[ \mathbb{E} \| \hat{x} - x \| ^2 = \mathbb{E} \| \hat{x} \| ^2 - \| x \| ^2 \leq \mathbb{E} \| \hat{x} \| ^2 \]

\[ \leq \mathbb{E} \| \hat{x} \| _2^2 \leq \mathbb{E} \left( \sum \| b \| _2^2 \right) ^2 = \| b \| _2^2 \mathbb{E} \xi ^2 \]

- Matrix inversion algorithm is not terribly accurate
- Each summation is \( \| b \| _2^2 \)
- We can average independent estimates
- \( \text{Var} \left[ \frac{1}{N} \sum_{i=1}^N \hat{x}_i \right] = \frac{1}{N} \text{Var} \left[ \hat{x}_i \right] \to 0 \) as \( N \to \infty \)
- We never look at the full matrix, we just look at \( T \) columns \( \Rightarrow \) efficient for \( N \approx 10^6 \)
- There is an extension to column diagonally dominant \( A \) (use diagonal as right preconditioner).
- Invented by Ulam, Von Neumann, Wosow in late 1940s - early 1950s.
Ex: PageRank
- Given a network of websites \( \{1, \ldots, N\} \), a restless web surfer picks one at random with probability \( (V_i)_{i \in \mathbb{N}} \).
- If the website has \( d_j > 0 \) outgoing links, with probability \( d_j \) 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 \( (V_i)_{i \in \mathbb{N}} \).
- PageRank problem: What is the long-run distribution of websites visited?

**Linear algebra solution**

Set \( P_{ij} = \begin{cases} \frac{1}{d_j} & \text{if website } j \text{ has } d_j > 0 \text{ outgoing links, including a link to } i \\ V_i & \text{if website } j \text{ has } d_j = 0 \text{ outgoing links,} \\ 0 & \text{otherwise.} \end{cases} \)

\[
\Rightarrow \hat{X} = \alpha P \hat{X} + (1-\alpha) \hat{V}
\]
\[
\Rightarrow (I - \alpha P) \hat{X} = (1-\alpha) \hat{V}
\]

- Used to rank website "importance" by Google, with \( \alpha = 0.85 \) and \( V_i = \frac{1}{N} \), \( i \in \mathbb{N} \).
\[ -\frac{1}{2} \left( \partial^2_{xx} + \partial^2_{yy} \right) u(x, y) = 0, \quad (x, y) \in E = [-K, K]^2 \]

\[ u(x, y) = f(x, y), \quad (x, y) \in \partial E \]

Discretize using

\[ L((x, y), (x', y')) = \begin{cases} 
-1/4, & |x-x'| + |y-y'| = 1, \\
1, & |x-x'| + |y-y'| = 0, \\
0, & |x-x'| + |y-y'| \geq 2.
\]
\[
\begin{align*}
\Rightarrow \begin{cases} 
(Lu)(x,y) = 0, & \max\{|x|, |y|\} < K, \\
\mathbb{U}(x,y) = f(x,y), & \max\{|x|, |y|\} = K.
\end{cases}
\end{align*}
\]

\[
\Rightarrow (\mathbb{I} - P) \mathbb{U} = \mathbb{f}
\]

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

**Option 1** solve for \(\mathbb{U} = (\mathbb{I} - P)^{-1} \mathbb{f}\) using MC

**Option 2** solve for \(\mathbb{U}(x,y) = \mathbb{E}\mathbb{f}_{(x,y)} \mathbb{I}^{-1} \mathbb{f}\) using MC.

a) solve for \((\mathbb{I} - P^{-1})^{-1} \mathbb{E}_{(x,y)} \mathbb{f}\) using MC.

b) Calculate \(\langle (\mathbb{I} - P^{-1})^{-1} \mathbb{E}_{(x,y)} \mathbb{f} \rangle\) (possibly for many different vectors \(\mathbb{f}\)).

\[
\mathbb{U}(x,y) = \mathbb{E}_{Z_0 = (x,y)} \left[ \mathbb{f}(Z_{\infty}) \right]
\]

random walk, stopped at boundary

"Feynman-Kac formula"

---

**Self-avoiding walk (SAW)** \(X = (\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_T)\) where
\(\mathbb{X}_0 = (0,0)\), each \(\mathbb{X}_t = (a,b)\) for integers \(a, b\),
\(\|\mathbb{X}_t - \mathbb{X}_{t+1}\| = 1\), and all the \(\mathbb{X}_t\) are distinct.
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 \( t=1, 2, ..., T-1 \), 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 \( t=0, 1, ..., T-1 \), let \( n_t = \# \text{ open vertices} \)
   - 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
\[ E\left[ \mathbb{1}_{\text{success}} \; w_T f(x) \right] = \sum_{y \in \{\text{length } T \text{ SAWs}\}} f(y) \text{ for any function } f. \]
\[ \mathbb{E} \left[ \prod_{t=0}^{T-1} \mathbb{1}_{\text{success } w_{t+1}(y)} \right] = \sum_{y \in \{\text{length } T \text{ SAWs}\}} \prod_{t=0}^{T-1} \frac{1}{n_t} n_{t+1} f(y) \]

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

\[ = \mathbb{E} \left[ f(y) \right] \cdot n_0 \cdot n_1 \cdot \cdots \cdot n_T \]

\[ \rightarrow \text{ In particular, we have an unbiased estimator } \]

\[ \hat{Z}_T = \# \{\text{SAWs of length } T\} = \mathbb{E} \left[ \prod_{t=0}^{T-1} \mathbb{1}_{\text{success } w_{t+1}} \right] \]

**Resampling** Let's say we have a weighted ensemble \((w_i, x_i)_{i \in \mathbb{N}}\).

- 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 \(k w_i\).
- Resampling is the most extreme splitting/killing approach: we replace \((w_i, x_i)_{i \in \mathbb{N}}\) with a new ensemble \((\hat{w}_i, \hat{x}_i)_{i \in \mathbb{N}}\) of equally weighted particles.
Bootstrap resampling

Input: Unevenly weighted ensemble \((w_i, x_i)\) for \(i = 1, \ldots, N\).

Output: Evenly weighted ensemble \((\hat{w}_i, \hat{x}_i)\) for \(i = 1, \ldots, N\) s.t.

\[ E\left[ \sum_{i=1}^{N} \hat{w}_i f(\hat{x}_i) \right] = \sum_{i=1}^{N} w_i f(x_i) \quad \text{for all functions } f. \]

1. Set \(\overline{w} = \frac{1}{N} \sum_{i=1}^{N} w_i.\)

2. For \(i = 1, 2, \ldots, N:\)
   a) Independently sample \(\hat{x}_i \in \{x^1, \ldots, x^N\}\) with
   \[ P\{\hat{x}_i = x^j\} = \frac{w_j}{\sum_{k=1}^{N} w_k}. \]
   b) Set \(\hat{w}_i = \overline{w}.\)

\(\Rightarrow\) Number of copies \(\tilde{C} = (c^1, \ldots, c^N)\) for each particle \(x^i\) has distribution \(\tilde{C} \sim \text{Mult}(N, \frac{w^1}{\sum_{k=1}^{N} w_k}, \ldots, \frac{w^N}{\sum_{k=1}^{N} w_k}).\)

Q: Is this optimal?

Improved (pivotal) resampling

1. Set \(\overline{w} = \frac{1}{N} \sum_{i=1}^{N} w_i, \quad \overline{C} = \left(\frac{w^1}{\overline{w}}, \frac{w^2}{\overline{w}}, \ldots, \frac{w^N}{\overline{w}}\right)\),

   \(S = 1, \quad E = 2.\)

2. For \(i = 1, 2, \ldots, N-1:\)
   a) Let \(a = C^S - LC^S\) and \(b = C^E \cdot L C^E\) be the non-integer parts of \(C^S\) and \(C^E\).
   b) If \(a + b \leq 1:\)
i. Set $C^s = C^s + b, \ C^e = C^e - b, \ e = e + 1$ \(\text{with probability } a/(a+b)\)

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

If $a+b > 1$:

i. Set $C^s = C^s + (1-a), \ C^e = C^e - (1-a), \ s = e, \ e = e + 1$ \(\text{with probability } (1-b)/(2-a-b)\).

ii. Otherwise, set $C^s = C^s - (2-a-b), \ C^e = C^e + (1-b), \ e = e + 1$.

3. Let the weighted ensemble $(\hat{\omega}^i, \hat{x}^i)$ consist of $C^s$ copies of each particle $x^i$, with all the weights equal to $\omega_i = \overline{\omega}$.

$\Rightarrow$ Entries of $\hat{C}$ are negatively correlated with mean $E[C_i] = \omega_i/\overline{\omega}$ and variance $\text{Var}[C_i]$ as low as possible.

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

**Bootstrap**

\[C_1 = 4\]

\[
\begin{array}{cccc}
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\end{array}
\]

**Pivotal**

\[C_2 = 3 \quad C_3 = C_4 = 1\]

\[
\begin{array}{cccc}
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ \\
\end{array}
\]

$\omega_i/\overline{\omega} = 3/2$ \(\text{round to } 0 \text{ or } 1\)
**Takeaway** Combining pivotal resampling with the Rosenbluth & Rosenbluth algorithm leads to a powerful sampler for SAWs.

**Forward flux sampling**

- We want to calculate \( P \{ \tau_{10} < \tau_{-1} \} \)

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 \( P \{ \tau_{10} < \tau_{-1} \} \).

- Idea: Use the fact that

\[
P \{ \tau_{10} < \tau_{-1} \} = P \{ \tau_{-1} < \tau_{-1} \} \prod_{s=2}^{10} P \{ \tau_s < \tau_{s-1} | \tau_{s-1} > \tau_{s-2} \}.
\]

**Forward flux sampling (FFS)**

1. Set \( X_0^i = 0, w_0^i = 1/N \) for \( 1 \leq i \leq N \).

2. For \( s = 1, \ldots, 10 \):
   - a) Simulate each \( X^i \) until hitting \( X^i = s \) or \( X^i = -1 \). In the former case, set \( w_0^i = w_{s+1}^i \). In
the latter case, set $w_s^i = 0$.

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

**Proposition**

$$E \left[ \sum_{i=1}^{N} w_{i_0} f(x^i) \right] = E \left[ f(x) \mathbb{1}\{\tau_{i_0} < \tau_{-1}\} \right]$$

for all bounded functions $f$.

$\Rightarrow$ In particular,

$$P\{\tau_{i_0} < \tau_{-1}\} = E \left[ \sum_{i=1}^{N} w_{i_0} \right].$$

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