* From matchings to the permanent (to FPRAS)

- We saw the Gibb's measure on matchings, i.e., for matching M of G, $\pi(M) = \frac{1}{Z(\lambda)} \lambda^{|M|}$, $\lambda \geq 0$
- MC (based on Metropolis) $\equiv$ $t_{mix} = O(\lambda^3 |E|^2 |V|)$.
  (starting from x_0 = max matching)
- Q: What is $\underbrace{Z(\lambda)}_{\text{partition fn}}$ for given G, λ?

Aside - #-P problems and FPRAS

- #P $\equiv$ natural analog of NP (for search/decision problems) in the context of counting problems
  - Eg - #SAT (# of satisfying assignments of CNF formula)
    - (# of solns for a given 2SAT formula, or DNF)
    - # of matchings - estimating perm(A)
    - Estimating $Z(\lambda)$

- A Fully Polynomial Randomized Approx Scheme (FPRAS) is a randomized algorithm for computing $f(x)$ which for any x, $\varepsilon > 0$, outputs r.v. Z s.t $\mathbb{P}\left[\frac{f(x)}{1+\varepsilon} \leq f(x) \leq f(x)(1+\varepsilon)\right] \geq \frac{3}{4}$ and runs in time poly$(x, 1/\varepsilon)$.

- Given FPRAS for f, can boost confidence from $3/4$ to $1-\delta$ at a slowdown cost of $O(\log 1/\delta)$

 - Take $t = O(\ln(1/\delta))$ indep trials of the FPRAS and output the median of the result. (median trick)

$$\mathbb{P}\left[\text{median} \notin [f(x)/(1+\varepsilon), f(x)\cdot(1+\varepsilon)]\right] \le \mathbb{P}\left[\ge t/2 \text{ trials lie outside interval}\right]$$
$$= \mathbb{P}\left[\mathrm{Bin}(t, 3/4) \le t/2\right]$$
$$\le 2e^{-t/48} = O(e^{-ct})$$

$\Rightarrow$ for $\mathbb{P}[\text{error}] \le \delta$, we need $t = O(\ln(1/\delta))$ samples.

---

* Estimating $Z(\lambda)$

Idea - Write
$$\boxed{Z(\lambda) = \frac{Z(\lambda_r)}{Z(\lambda_{r-1})} \cdot \frac{Z(\lambda_{r-1})}{Z(\lambda_{r-2})} \cdots \frac{Z(\lambda_1)}{Z(\lambda_0)} \cdot Z(\lambda_0)}$$

where $\lambda_0 = 0$, $\lambda_1 = \frac{\varepsilon}{|E|}$, $\lambda_i = \left(1+\frac{1}{n}\right)\lambda_{i-1}$, $\lambda_r = \lambda \le \left(1+\frac{1}{n}\right)\lambda_{r-1}$

$\Rightarrow \lambda_1 < \lambda_2 < \cdots < \lambda_r$, $r = O(n(\ln \lambda + \ln|E| + \ln(1/\varepsilon)))$

- Note: $Z(\lambda_1) = 1 + O(\varepsilon)$

 $Z(\lambda_0) = 1$

- Claim 1 - $1 \le \dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})} = \dfrac{\sum_k m_k \lambda_i^k}{\sum_k m_k \lambda_{i-1}^k} \le \left(1+\frac{1}{n}\right)^n \le e$

- Claim 2 - $\dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})} = \mathbb{E}_{\pi_{\lambda_{i-1}}}\left[\left(\dfrac{\lambda_i}{\lambda_{i-1}}\right)^{|M|}\right] \quad \forall\ i \geq 2$

Pf - $\mathbb{E}_{\pi_{\lambda_{i-1}}}\left[\left(\dfrac{\lambda_i}{\lambda_{i-1}}\right)^{|M|}\right] = \dfrac{1}{Z(\lambda_{i-1})^k}\sum m_k (\lambda_{i-1})^k \cdot \left(\dfrac{\lambda_i}{\lambda_{i-1}}\right)^k = \dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})}$

- Now we can estimate $\dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})}$ by sampling from $\pi_{\lambda_{i-1}}$

  - Run MC for $t_{mix}(\varepsilon)$ steps to get $\hat{\pi}_{\lambda_{i-1}}$ s.t $\|\pi_{\lambda_{i-1}} - \hat{\pi}_{\lambda_{i-1}}\|_{TV} < \varepsilon$, and estimate $\mathbb{E}_{\hat{\pi}_{\lambda_{i-1}}}\left[(1+1/n)^{|M|}\right]$

  - $\left|\mathbb{E}_{\pi_{\lambda_{i-1}}}\left[(1+1/n)^{|M|}\right] - \mathbb{E}_{\hat{\pi}_{\lambda_{i-1}}}\left[(1+1/n)^{|M|}\right]\right| \le (e-1)\|\pi_{\lambda_{i-1}} - \hat{\pi}_{\lambda_{i-1}}\|_{TV}$

  - Use $t^* = O(\lambda^3 |E|^2 |V| \ln(n/\varepsilon))$ to get $\|\pi-\hat{\pi}\|_{TV} \le \frac{\varepsilon}{6}$

    Use $\dfrac{n^2}{\varepsilon^2}\ln\left(\dfrac{1}{\delta}\right)$ samples from MC to get $\widehat{\left(\dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})}\right)} = \dfrac{Z(\lambda_i)}{Z(\lambda_{i-1})} \cdot (1 \mp \varepsilon/n)$

    wp $\geq 1-\delta$

- Overall, set $\hat{Z}(\lambda) = \left(\frac{\hat{Z}(\lambda_r)}{Z(\lambda_{r-1})}\right)\left(\frac{\hat{Z}(\lambda_{r-1})}{Z(\lambda_{r-2})}\right)\cdots\left(\frac{\hat{Z}(\lambda_1)}{Z(\lambda_0)}\right)$

$$\Rightarrow \frac{\hat{Z}(\lambda)}{Z(\lambda)} = \underbrace{(1 \mp O(\varepsilon/r))(1 \mp O(\varepsilon/r))\cdots(1 \mp O(\varepsilon/r))}^{r \text{ terms}}$$

$$= 1 \mp O(\varepsilon) \qquad \text{w.p.} \geq 1 - \underbrace{r\delta}_{\text{union bound}}$$

will set $\delta = 1/4r$ for FPRAS

- Total Running Time $= O\Big(\underbrace{r}_{\text{\# of estimates}} \cdot \underbrace{\left(\frac{r^2}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)\right)}_{\text{\# of samples}} \cdot \underbrace{\lambda^3 |E|^2 |V| \ln\left(\frac{r}{\varepsilon}\right)}_{\text{one run of MC}}\Big)$

Estimating m_k (a.k.a, the matching polynomial coeffs)

- When $k = |V|/2$, m_k is a perfect matching
- If G is bipartite (on $2n$ vertices), $m_n \equiv \text{perm}(A)$ ← L×R adjacency matrix
  - For $n \times n$ matrix A (with S_n = symmetric gp / set of permutations)
    - $\det(A) = \sum_{\sigma \in S_n} (-1)^{\text{sign}(\sigma)} \prod_{i=1}^{n} A(i, \sigma(i))$ (determinant)

      ↑ # of (i,j) st $i > j$ and $\sigma(i) < \sigma(j)$, ie, inversions

      $\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A(i, \sigma(i))$ (permanent)
  - Computing $\det(A) \equiv$ poly-time (e.g., via Gaussian elimination)

    Computing $\text{perm}(A) \equiv$ #P-complete (Valiant '79)

- Approximating m_k
  - Approximate $Z(\lambda)$ at $(n+1)$ pts and interpolate to find $\hat{f}(\lambda) = \sum_{k=0}^{n} \hat{m}_k \lambda^k$
    - Not stable / robust to perturbations.
  - Estimate $\boxed{m_k = \left(\frac{m_k}{m_{k-1}}\right) \cdot \left(\frac{m_{k-1}}{m_{k-2}}\right) \cdots \left(\frac{m_1}{m_0}\right) \cdot m_0}$, where $m_0 = 1$

Claim 1 - $\{m_k\}$ is log-concave, i.e. $m_{k-1} m_{k+1} \leq m_k^2 \; \forall k$

$\left(\text{or } \frac{\log m_{k-1} + \log m_{k+1}}{2} \leq \log m_k\right)$

Pf sketch - Construct an injective mapping from pairs of matchings with $k-1$ and $k+1$ edges, to a pair with k edges each (similar to injective map for bounding trix)

Claim 2 - If $\lambda = \frac{m_{k-1}}{m_k}$, then $m_{k'} \lambda^{k'}$ is maximized at $k' = k$ or $k-1$

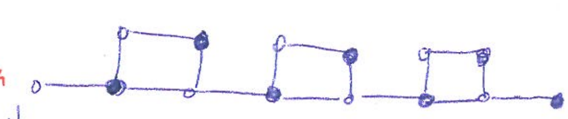
Pf - Note $m_k \lambda^k = m_{k-1} \lambda^{k-1}$. Now we show $\log(m_{k'})$ is maximized at $k' = k^*$. Using concavity, enough to show $m_k \lambda^k \geq m_{k+1} \lambda^{k+1}$ and $m_{k-1} \lambda^{k-1} \leq m_{k-2} \lambda^{k-2}$.
For first, by logconcavity, $\frac{m_{k+1} \lambda^{k+1}}{m_k \lambda^k} = \frac{m_{k+1} \lambda}{m_k} = \frac{m_{k+1} m_{k-1}}{m_k \cdot m_k} \leq 1$. Same for other ineq.

Algorithm for estimating m_k

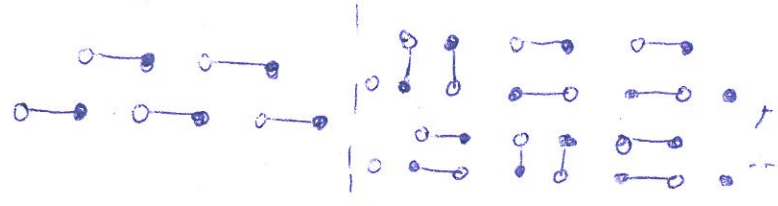
- Start with $\lambda = \frac{m_0}{m_1} = \frac{1}{|E|}$, and gradually raise λ, and sample from π_λ repeatedly, till the fraction of $(k-1)$ and (k) matchings (among samples for a given λ) are $\geq \frac{c}{n}$ for some constant c
- For this final choice of λ, use samples from π_λ to estimate m_k/m_{k-1} $\left(\text{estimate} = \frac{\#\,k\text{ match}^d}{\#\,k-1\text{ match}^d} \cdot \frac{1}{\lambda}\right)$
- Compute $\hat{m}_k = \left(\widehat{\frac{m_k}{m_{k-1}}}\right) \cdot \left(\widehat{\frac{m_{k-1}}{m_{k-2}}}\right) \cdots \left(\frac{m_0}{m_1}\right)$

Problem - For approximating $\mathrm{perm}(A)$, need m_n (fixed in Jerrum-Sinclair-Vigoda '04 via more complex reweighting)

- The above algo is poly time if m_n/m_{n-1} is poly(n)
- True for dense graphs (min degree $\geq \frac{n}{2}$), $G_{n,1/2}$, lattices
- However, consider $G \equiv$



  - bipartite, $2n = 4\ell + 2 = |V|$ (# squares)
  - $m_n = 1$, $m_{n-1} \geq 2^\ell$



* <u>For more general #P problems</u>

- Can not get an FPRAS for problems where the decision version is NP-complete
  - hardcore model / max independent sets

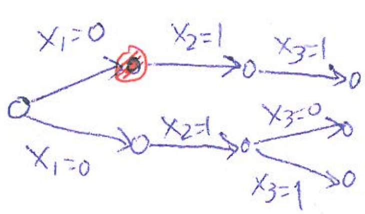
- However many #P problems are not of this form
  - matchings, perfect matchings, partition fn of Ising model, volume of a convex set, counting indep sets / colorings with $q \geq \Delta + 1$ colors

---

<u>Thm</u>. For <u>self-reducible problems</u>, $\exists$ FPRAS iff $\exists$ a poly-time sampler for the uniform distribution

- $\boxed{\text{Self-reducible}}$ $\equiv$ an NP search-problem where the set of solutions can be partitioned into a poly number of sets, each in 1-1 correspondence with set of solutions of smaller instances

- <u>Eg</u> - SAT $\quad \phi = (\overline{x}_1 \vee x_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_3)$



- Can represent solns as tree $T(x)$
- Leaves $\equiv$ satisfying assignments
- Nodes $\equiv$ smaller formulas (Eg. $\equiv x_2 \wedge x_3$)