Lecture 14

- Recall J-S MC for sampling matchings in unweighted graphs $G=(V,E)$, according to the Gibbs distn': $\pi_\lambda(M) = \frac{1}{Z(\lambda)} \lambda^{|M|}$.
- Running time: poly(n, λ), $n = |V|$
- $Z(\lambda) = \sum_k m_k \lambda^k$ — partition function (or matching polynomial); m_k: # of k-matchings
- Last time: Estimation of $Z(\lambda)$.
- Today: Estimation of coefficients m_k
- **Estimating coefficients m_k**

exact estimation: when $k = \frac{|V|}{2}$, m_k is # perfect matchings

if G is bipartite with n vertices on each side, computing $m_n \equiv$ computing per(A) — adjacency matrix

def: If A is $n \times n$, $\text{per}(A) = \sum_\sigma \prod_{i=1}^{n} A(i, \sigma(i))$, where the summation is over all permutations σ of $\{1, \dots, n\}$

(permanent of a matrix)

contrast this, with the determinant:

def: If A is $n \times n$, the determinant of A is

$$\det(A) = \sum_\sigma (-1)^{\text{sign}(\sigma)} \prod_{i=1}^{n} A(i, \sigma(i))$$

where sign(σ) is # of pairs $i<j$ s.t. $\sigma(i) > \sigma(j)$

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

Proof: • First verify that: $m_k \lambda^k = m_k \lambda \cdot \lambda^{k-1} = m_{k-1} \cdot \lambda^{k-1}$

• Moreover,

for all $k' \le k-1$: $\frac{m_{k'-1}\lambda^{k'-1}}{m_{k'}\lambda^{k'}} = \frac{1}{\lambda} \cdot \frac{m_{k'-1}}{m_{k'}} = \frac{m_k}{m_{k-1}} \cdot \frac{m_{k'-1}}{m_{k'}} \underset{\text{using claim 1}}{\le} \frac{m_{k'}}{m_{k'-1}} \cdot \frac{m_{k'-1}}{m_{k'}} \le 1$

for all $k' \ge k$: $\frac{m_{k'+1}\cdot\lambda^{k'+1}}{m_{k'}\lambda^{k'}} = \lambda \frac{m_{k'+1}}{m_{k'}} = \frac{m_{k-1}}{m_k} \cdot \frac{m_{k'+1}}{m_{k'}} \underset{\text{using claim 1}}{\le} \frac{m_{k'}}{m_{k'+1}} \cdot \frac{m_{k'+1}}{m_{k'}} \le 1$

$\boxtimes$.

• Claims 1, 2 suggest the following algorithm; for estimating $\frac{m_{k'}}{m_{k'-1}}$ for $k' = 2, \dots, k$:

i) gradually raise λ (starting at $\frac{m_0}{m_1} = \frac{1}{|E|}$) and sample from Gibbs distn π_λ repeatedly until we see "lots of" $(k'-1)$- and k'-matchings, where "lots of" ≡ they appear w/ prob. at least $\frac{c}{n}$, for some constant c (Claim 2, confirms that we'll not need to raise λ beyond $\frac{m_{k'-1}}{m_{k'}}$)

ii) For the λ chosen in step i, sample MC repeatedly to estimate $\frac{m_{k'}}{m_{k'-1}}$; take as estimator the ratio of the number of k'-matchings to the number of $(k'-1)$-matchings multiplied by $\frac{1}{\lambda}$.

ex 1pt: Can estimate $\frac{m_{k'}}{m_{k'-1}}$ to within a factor of $(1 \pm \frac{\varepsilon}{n})$.

w/ probability at least $1 - \frac{1}{4n}$ in time $\text{poly}(n, \frac{1}{\varepsilon}, \frac{m_{k'-1}}{m_{k'}})$.

- computing $\det(A)$ can be done efficiently w/ Gaussian Elimination
- computing $\mathrm{per}(A)$ is #P-complete (Valiant '79).

estimation of m_k:

- if we could compute $Z(\lambda)$ exactly at any desired λ, we would obtain all the m_k's exactly; indeed $Z(\lambda)$ is a polynomial of degree n w/ coefficients $m_0, .., m_n$, so we could do that by computing $Z(\lambda)$ at $n+1$ points and interpolating.
- however, approach is not robust to error.
- We use a different approach, aiming at estimating ratios of the form $\frac{m_k}{m_{k-1}}$ and taking a telescoping product:

$$m_k = \left(\prod_{k'=1}^{k} \frac{m_{k'}}{m_{k'-1}} \right) \cdot m_0 \, . \qquad (*)$$

$m_0 = 1$ (trivially) so we only need estimates of $\frac{m_{k'}}{m_{k'-1}}$ for all $k'=1, .., k$.

- **Claim 1:** The sequence $\{m_k\}_k$ is log-concave, i.e.

$$m_{k-1} \cdot m_{k+1} \leq m_k^2 \quad , \text{for all } k.$$

exercise (1pt): Show claim 1, by defining an injective mapping from pairs of $(k+1)$- and $(k-1)$-matchings to pairs of k-matchings

- plugging the above estimator into (*) and noticing that the worst possible ratio $\frac{m_{k'-1}}{m_{k'}}$ is $\frac{m_{k-1}}{m_k}$ (by claim 1), we obtain that we can estimate m_k to within a factor of $(1 \pm \varepsilon)$ w/ probability $\geq \frac{3}{4}$ in time $\text{poly}(n, \frac{1}{\varepsilon}, \frac{m_{k-1}}{m_k})$.

NOTE: there is nothing special about $\frac{3}{4}$; we can boost this by repeating many times & outputting the median of the results.

Ex 1pt: If k^* is the (unknown) size of a maximum matching in a given graph, show how to use the above algorithm to compute a matching of size $\geq (1-\varepsilon)\cdot k^*$ in time $n^{O(1/\varepsilon)}$.

- How large can $\frac{m_{k-1}}{m_k}$ be?

  - In many interesting classes of graphs, such as
    - dense graphs (every degree is $\geq \frac{n}{2}$)
    - random graphs (e.g. in the $G_{n,1/2}$ model, whp)
    - regular lattices (finite square regions of $\mathbb{Z}^2$)

    the ratio $\frac{m_{k-1}}{m_k}$ is polynomially bounded in n.

  - However, in general it can be very bad, e.g.

- $|V| = 4\ell + 2 =: 2n$, where ℓ = # squares

easy to see $m_n = 1$ (unique perfect matching), but $m_{n-1} \geq 2^{\ell}$.