8^{TH} TUTORIAL ON RANDOMIZED ALGORITHMS

Approximately counting matchings a.k.a. estimating permanent

1. *Counting matchings.* Let $G = (U \cup V, E)$ be a bipartite graph where $|U| = |V| = n$ and $\delta(G) > n/2$. We define:

$M_k =$ the set of matchings of size k in G,
$m_k = |M_k|$ the number of matchings of size k in G, and
$r_k = m_k/m_{k-1} =$ the fraction of the $\#$ of k-matchings to the $\#$ of $k-1$-matchings.

Let $\alpha \geq 1$ be a real number such that $1/\alpha \leq r_k \leq \alpha$. Pick $N = n^7\alpha$ elements from $M_k \cup M_{k-1}$ independently uniformly at random (approximately uniform generation covered in the lecture). Set $\hat{r_k}$ to the fraction of observed k-matchings to $(k-1)$-matchings. Show that

$$\left(1 - 1/n^3\right) r_k \leq \hat{r_k} \leq \left(1 + 1/n^3\right) r_k$$

with probability at least $1 - \exp(-n)$. (Hint: use the Estimator theorem from the lecture.)

(Also recall why accurate approximations of r_k's are useful for estimating the number of perfect matchings.)

2. Let G_k be the graph constructed from $G = (U \cup V, E)$ such that we add $n-k$ vertices to each partite and connect each new vertex with all old vertices in the opposite partite. Show that if R is the fraction of perfect matchings to the number of almost perfect matchings (all but one vertex in each partite is matched) in the new graph G_k then

$$R = \frac{m_k}{m_{k+1} + 2(n-k)m_k + (n-k+1)^2 m_{k-1}}$$

3. *Estimating permanent.* Let $A \in \{0,1\}^{n\times n}$ be a matrix. Let $\varepsilon_{i,j}$ be independent random ± 1 variables. Let $B \in \{-1,0,1\}^{n\times n}$ be a matrix such that $B_{i,j} = \varepsilon_{i,j} A_{i,j}$ (uniformly randomly independently assign signs to entries of A).

a) Show that $\mathbb{E}[\det(B)] = 0$

b) Show that $\mathbb{E}[\det(B)^2] = \operatorname{perm}(A)$ (permanent of A)

Now it may look like this gives an efficient and accurate estimation for the permanent. Where's the catch?

4. *Bonus: polynomial-time interactive protocol for permanent.* Show that permanent is in IP. We say that a language $L \subseteq \{0,1\}^*$ is in IP if

- The verifier V gets a word $w \in \{0,1\}^*$, works in polynomial time in $|w|$ and can use random bits.
- The verifier V can communicate with the prover P (which is computationally unbounded).
- We say that $L \in IP$ if there is a prover P and a verifier V such that:
  - Completeness: for each $w \in L$ we have
  $$\Pr[V(w) \text{ accepts the proof of } P] \geq 2/3$$
  - Soundness: for any $x \notin L$ and any prover Q we have
  $$\Pr[V(x) \text{ accepts the proof of } Q] \leq 1/3$$

Our goal is to show that the decision problem whether or not $\mathrm{perm}(A) = k$ for a given matrix $A \in \{0,1\}^{n \times n}$ and $k \in \mathbb{N}$ is in IP.