LECTURE 16

Last time

- Lower bound for testing triangle-freeness

Today

- Canonical testers for the dense graph model
- Approximating the average degree

Canonical Tester for Dense Graphs

Canonical Tester (**Input:** ε, n; query access to adjacency matrix of $G=(V,E)$)

1. Sample s nodes uniformly at random.
2. Query all pairs of sampled nodes.
3. **Accept** or **reject** based on available information.

- Consider any property $\mathcal{P}$ of graphs that does not depend on the names of the nodes. That is, if $G \in \mathcal{P}$ and G' is isomorphic to G then $G' \in \mathcal{P}$.

Exercise: Show that if there is an ε-tester T for $\mathcal{P}$ with query complexity $q(\varepsilon,n)$, then there is a canonical ε-tester T' for $\mathcal{P}$ with query complexity $O(q^2(\varepsilon,n))$. Moreover, if T has 1-sided error, so does T'.

A lower bound q for canonical tester implies a lower bound $\sqrt{q}$ for every tester

To complete triangle-freeness testing lower bound, it is sufficient to prove the lower bound $\Omega\left(\left(\frac{c}{\varepsilon}\right)^{c \log \frac{c}{\varepsilon}}\right)$ for 1-sided error canonical testers.

Exercise

Exercise: Show that if there is an ε-tester T for $\boldsymbol{\mathcal{P}}$ with query complexity $q(\varepsilon,n)$, then there is a canonical ε-tester T' for $\boldsymbol{\mathcal{P}}$ with query complexity $O(q^2(\varepsilon,n))$. Moreover, if T has 1-sided error, so does T'.

Exercise

Exercise: Show that if there is an ε-tester T for $\boldsymbol{\mathcal{P}}$ with query complexity $q(\varepsilon,n)$, then there is a canonical ε-tester T' for $\boldsymbol{\mathcal{P}}$ with query complexity $O(q^2(\varepsilon,n))$. Moreover, if T has 1-sided error, so does T'.

Completing the Triangle-Freeness Lower Bound

- A 1-sided error tester can reject only if it finds a triangle.
- Last time: ∃ a graph G that is ε-far from being tringle free, where $p = O\left(\left(\frac{\varepsilon}{c}\right)^{c \log \frac{c}{\varepsilon}}\right)$ fraction of triples are triangles
- Consider a canonical tester T that samples q vertices.
- Let X be the number of triangles the tester catches.

$$\mathbb{E}[X] = p\binom{q}{3} = \Theta(p \cdot q^3)$$

- Suppose q is set so that $\mathbb{E}[X] \leq 1/2$
- By Markov, $\Pr[T \text{ rejects } G] \leq \Pr[X \geq 1] \leq \mathbb{E}[X] \leq \frac{1}{2} < \frac{2}{3}$
- So, for T to reject with high enough probability, $q = \Omega\left(p^{-\frac{1}{3}}\right)$

$$q = \Omega\left(\left(\frac{c}{\varepsilon}\right)^{c' \log \frac{c}{\varepsilon}}\right)$$

Graph Models for Sublinear Algorithms

Dense Graph Model

- Input is represented by adjacency matrix
- Access: Adjacency queries: Is (i, j) an edge?
- For property testing, distance is normalized by n^2 or $\binom{n}{2}$

Bounded Degree Model

- Input is represented by adjacency lists of length Δ (degree bound)
- Access: Neighbor queries: What is the ith neighbor of vertex v?
- For property testing, distance is normalized by Δn

General Graph Model

- Input is represented by adjacency lists and adjacency matrix, sometimes with additional data structures
- Access: adjacency, neighbor and degree queries
- For property testing, distance is normalized by m

Approximating the Average Degree

Input: parameters ε, n, access to an undirected n-node graph $G = (V, E)$ represented by *adjacency lists*.

Queries

- **Degree queries:** given vertex v, return its degree $d(v)$
- **Neighbor queries:** given (v, i), return the i-th neighbor of v

Goal: Return, with probability at least 2/3, an estimate $\hat{d}$ of the average degree $\bar{d} = \frac{1}{n}\sum_{v \in V} d(v)$

Estimating the average degree is equivalent to estimating the number of edges:

$$\bar{d} = \frac{2m}{n}$$

Estimating the Average Degree: Results

- An estimate $\hat{d}$ is a c-approximation for $\bar{d}$ if
$$\bar{d} \leq \hat{d} \leq c \cdot \bar{d}$$
- Assumption: $\bar{d} \geq 1$

- [Feige 06]: $(2+\varepsilon)$-approximation with $\tilde{O}(\sqrt{n})$ degree queries
  Need $\Omega(n)$ degree queries to get better than 2-approximation

- [Goldreich Ron 08]: $(1+\varepsilon)$-approximation with $\tilde{O}(\sqrt{n})$ degree and neighbor queries

Simple Lower Bounds

We need $\Omega(n)$ queries to get a c-approximation to the average of numbers $x_1, \dots, x_n \in \{0,1,\dots,n-1\}$ for any constant c.

Proof: Use Yao's Minimax. To distinguish between

- all numbers are 1

  the average is 1

- random c numbers are n-1 and the rest are 1

  the average is $> c$

we need $\Omega\left(\frac{n}{c}\right) = \Omega(n)$ queries.

But degree sequences are special!

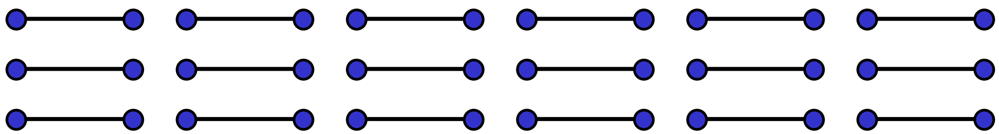
1 1 1 1 1 1 1 1 1 1 n-1 n-1 is not a degree sequence

Simple Lower Bounds

We need $\Omega(\sqrt{n})$ degree queries to get a c-approximation for any constant c.

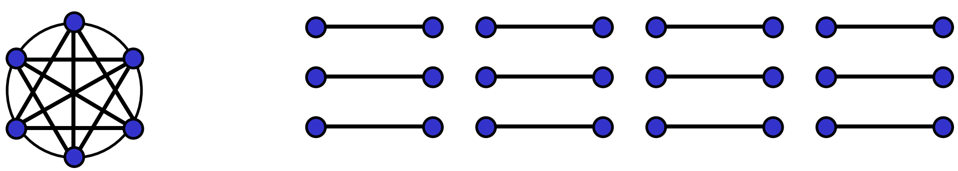
Proof: Use Yao's Minimax. To distinguish between random isomorphisms of

- a matching of $n/2$ edges



$\bar{d} = 1$

- $\sqrt{cn}$-clique and a matching on remaining nodes



$\bar{d} > c$

We need $\Omega\left(\frac{\sqrt{n}}{\sqrt{c}}\right) = \Omega(\sqrt{n})$ queries.

Average: Degree Approximation Guarantee

- $\Pr\left[\left|\hat{d} - \bar{d}\right| \geq \varepsilon \cdot \bar{d}\right] \leq \frac{1}{3}$
- In particular, $\hat{d}$ is an *unbiased* estimator: $\mathbb{E}\left[\hat{d}\right] = \bar{d}$
- The approximation guarantee is equivalent to $(1+\varepsilon)$-approximation

$$(1-\varepsilon)\cdot \bar{d} \leq \hat{d} \leq (1+\varepsilon)\cdot \bar{d}$$

$$\bar{d} \leq \frac{\hat{d}}{1-\varepsilon} \leq \frac{1+\varepsilon}{1-\varepsilon}\cdot \bar{d}$$

$\frac{1+\varepsilon}{1-\varepsilon} \leq 1 + \frac{2\varepsilon}{1-\varepsilon} \leq 1 + 4\varepsilon$ for $\varepsilon \leq 1/2$

Conclusion: $\frac{\hat{d}}{1-\varepsilon}$ gives a $(1+\epsilon')$-approximation, where $\epsilon' = 4\varepsilon$

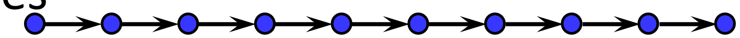
- Amplification of success probability: If we want error probability δ,

  we repeat the algorithm $\Theta\left(\log\frac{1}{\delta}\right)$ and output the median answer.

Average Degree Estimation [Eden Ron Seshadhri]

Main idea: To reduce variance (by reducing the range of degrees), we will count each edge towards its endpoint with smaller degree.

- Define ordering on V: for $u, v \in V$, we say $u \prec v$ if $d(u) < d(v)$ or if $d(u) = d(v)$ *and* $id(u) < id(v)$. to break ties
- ``Orient'' the edges towards higher-ID nodes



- Define $N(v)$ to be the set of neighbors of v.

Algorithm (**Input:** ε, n; degree and neighbor query access to $G=(V,E)$)

1. Set $k = \frac{12}{\varepsilon^2} \cdot \sqrt{n}$ and initialize $X_i = 0$ for all $i \in [k]$
2. For $i = 1$ to k **do**
   a. Sample a vertex $u \in V$ u.i.r. and query its degree $d(u)$
   b. Sample a vertex $v \in N(u)$ u.i.r. by making a neighbor query to v.
   c. If $u \prec v$, set $X_i = 2d(u)$
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i \in [k]} X_i$

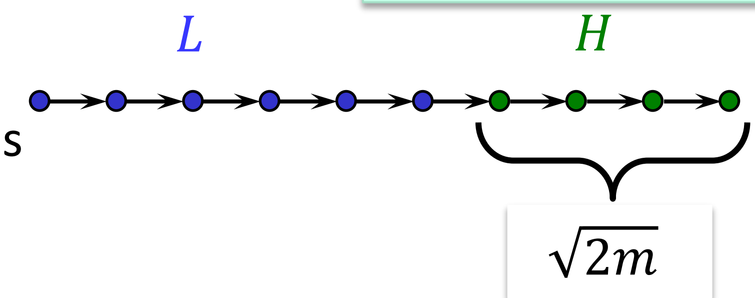
Outdegree Lemma

Let $d^+(u)$ denote the number of neighbors v of u with $u \prec v$.

Outdegree Lemma

For all vertices v, the outdegree $d^+(v) < \sqrt{2m}$.

vertices with the highest degree



Proof:

- Let $H \subseteq V$ be the set of the $\sqrt{2m}$ vertices with highest rank according to $\prec$.
- Let $L = V \backslash \mathrm{H}$.

1. Consider $v \in H$.
   $d^+(v)$ is the number of neighbors of v of rank higher than v.
   v is among the $\sqrt{2m}$ vertices of the highest rank, so $d^+(v) < \sqrt{2m}$
2. Consider $v \in L$. All $u \in H$, by definition, have degree at least $d(v)$.

   Then the sum of all degrees, $2m$, is greater than $\sqrt{2m} \cdot d(v)$.

$$d^{+(v)} \leq d(v) < \frac{2m}{\sqrt{2m}} = \sqrt{2m}$$

Analysis: Expectation

Algorithm (**Input:** ε, n; vertex and neighbor query access to $G=(V,E)$)

1. Set $k = \frac{12}{\varepsilon^2} \cdot \sqrt{n}$ and initialize $X_i = 0$ for all $i \in [k]$
2. For $i = 1$ to k **do**
   a. Sample a vertex $u \in V$ u.i.r. and query its degree $d(u)$
   b. Sample a vertex $v \in N(u)$ u.i.r. by making a neighbor query to v.
   c. If $u \prec v$, set $X_i = 2d(u)$
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i\in[k]} X_i$

- Let $d^+(u)$ denote the number of neighbors v of u with $u \prec v$.
- Let X denote one of the variables X_i. (They all have the same distribution.)
- Let U denote the random variable equal to the node u sampled in Step 2a.

$$\mathbb{E}[X] = \mathbb{E}\big[\mathbb{E}[X|U]\big]$$

By the compact form of the Law of Total Expectation

$$\mathbb{E}[X|U] = \frac{d^+(U)}{d(U)} \cdot 2d(U) = 2d^+(U).$$

$d^+(U)$ is # of neighbors v of U for which $X = 2d(U)$

$$\mathbb{E}[X] = \mathbb{E}[2d^+(U)] = 2\sum_{u\in V} \frac{1}{n} \cdot d^+(u) = \frac{2m}{n} = \bar{d}$$

Analysis: Variance

Reminders:
$d^+(u) =$ the # of neighbors v of u with $u \prec v$.
RV X denotes X_i.
RV $U =$ the node u sampled in Step 2a.

- $\mathrm{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 < \mathbb{E}[X^2]$
- $\mathbb{E}[X^2] = \left[\mathbb{E}[X^2|U]\right]$

By the compact form of the Law of Total Expectation

- $\mathbb{E}[X^2|U] = \frac{d^+(U)}{d(U)} \cdot \left(2d(U)\right)^2 = 4d^+(U) \cdot d(U)$.

Outdegree Lemma

$\forall v, \quad d^+(v) < \sqrt{2m}$.

- $$\begin{aligned} \mathbb{E}[X^2] &= \mathbb{E}[4d^+(U) \cdot d(U)] \\ &< \mathbb{E}\left[4 \cdot \sqrt{2m} \cdot d(U)\right] \\ &= 4\sqrt{2m} \cdot \mathbb{E}[d(U)] \\ &= 4\sqrt{2m} \cdot \bar{d}. \end{aligned}$$

By linearity of expectation

By definition of expectation

We get that $\mathrm{Var}[X] < 4\sqrt{2m} \cdot \bar{d}$.

Analysis: Putting It All Together

Algorithm (Input: ε, n; vertex and neighbor query access to $G=(V,E)$)

1. Set $k = \frac{12}{\varepsilon^2} \cdot \sqrt{n}$ and initialize $X_i = 0$ for all $i \in [k]$
2. For $i = 1$ to k **do**
   a. Sample a vertex $u \in V$ u.i.r. and query its degree $d(u)$
   b. Sample a vertex $v \in N(u)$ u.i.r. by making a neighbor query to v.
   c. If $u \prec v$, set $X_i = 2d(u)$
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i \in [k]} X_i$

$d^+(u) =$ the # of neighbors v of u with $u \prec v$.
RV X denotes X_i.
RV $U =$ the node u sampled in Step 2a.

- $\mathbb{E}[\hat{d}] = \mathbb{E}[X] = \bar{d}$
- $\mathrm{Var}[\hat{d}] = \frac{\mathrm{Var}[X]}{k} \le \frac{4\sqrt{2m} \cdot \bar{d}}{k}$
- $\Pr[|\hat{d} - \bar{d}| \ge \varepsilon \cdot \bar{d}] = \Pr[|\hat{d} - \mathbb{E}[\hat{d}]| \ge \varepsilon \cdot \bar{d}] \le \frac{\mathrm{Var}[\hat{d}]}{(\varepsilon \cdot \bar{d})^2}$ (By Chebyshev)

$$\le \frac{4\sqrt{2m} \cdot \bar{d}}{k \cdot \varepsilon^2 \cdot \bar{d}^2} = \frac{4\sqrt{2m} \cdot n}{k \cdot \varepsilon^2 \cdot 2m} = \frac{4n}{k \cdot \varepsilon^2 \cdot \sqrt{2m}} = \frac{4\sqrt{n}}{k \cdot \varepsilon^2 \cdot \sqrt{\bar{d}}} = \frac{1}{3\sqrt{\bar{d}}} \le \frac{1}{3}$$

(Our choice of k) (Assumption $\bar{d} \ge 1$)

Approximating the Average Degree: Run Time

Algorithm (Input: ε, n; vertex and neighbor query access to $G=(V,E)$)

1. Set $k = \frac{12}{\varepsilon^2} \cdot \sqrt{n}$ and initialize $X_i = 0$ for all $i \in [k]$
2. For $i = 1$ to k **do**
   a. Sample a vertex $u \in V$ u.i.r. and query its degree $d(u)$
   b. Sample a vertex $v \in N(u)$ u.i.r. by making a neighbor query to v.
   c. If $u \prec v$, set $X_i = 2d(u)$
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i \in [k]} X_i$

Running time:

$$O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$$

to get $\Pr\left[\left|\hat{d} - \bar{d}\right| \geq \varepsilon \cdot \bar{d}\right] \leq \frac{1}{3}$