LECTURE 17

Last time

• Lower bound for testing triangle-freeness
• Canonical testers for the dense graph model

Today

• Approximating the average degree
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 \(\Delta\) (degree bound)
• Access: Neighbor queries: What is the \(i\)th neighbor of vertex \(v\)?
• For property testing, distance is normalized by \(\Delta n\)

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 $\varepsilon, n$, 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, w.p. at least $2/3$, an estimate $\hat{d}$ for 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

- Need $\Omega(n)$ queries to get a $c$-approximation to the average of numbers $x_1, \ldots, x_n \in \{0,1,\ldots,n-1\}$ 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 n-1 n-1 is not a degree sequence
Simple Lower Bounds

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

we need $\Omega\left(\frac{\sqrt{n}}{\sqrt{c}}\right) = \Omega(\sqrt{n})$ queries
Average: Degree Approximation Guarantee

- \( \Pr[|\hat{d} - \bar{d}| \geq \varepsilon \cdot \bar{d}] \leq \frac{1}{3} \)

- In particular, \( \hat{d} \) is an unbiased estimator: \( \mathbb{E}[\hat{d}] = \bar{d} \)

- The approximation guarantee is equivalent to \((1 + \varepsilon)\)-approximation
  \[
  (1 - \varepsilon) \cdot \bar{d} \leq \hat{d} \leq (1 + \varepsilon) \cdot \bar{d}
  \]
  \[
  \bar{d} \leq \frac{\hat{d}}{1 - \varepsilon} \leq \frac{1 + \varepsilon}{1 - \varepsilon} \cdot \bar{d}
  \]

  \[
  \frac{1 + \varepsilon}{1 - \varepsilon} \leq 1 + \frac{2\varepsilon}{1 - \varepsilon} \leq 1 + 4\varepsilon \text{ for } \varepsilon \leq 1/2
  \]

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

- Amplification of success probability: If we want error probability \( \delta \),
  we repeat the algorithm \( \Theta \left( \log \frac{1}{\delta} \right) \) and output the median answer.
Main idea: To reduce variance, we will count each edge towards its endpoint with smaller degree.

- Define ordering on $V$: for $u, v \in V$, we say $u < v$ if $d(u) < d(v)$ or if $d(u) = d(v)$ and $id(u) < id(v)$.
- "Orient" the edges towards higher-ID nodes.
- Define $N(v)$ to be the set of neighbors of $v$.

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

1. Set $k = \frac{12}{\epsilon^2} \cdot \sqrt{n}$ 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 $u < v$, set $X_i = 2d(u)$
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i \in [k]} X_i$
Analysis: Expectation

Algorithm (Input: $\varepsilon, n$; 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 $u < v$, 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 $u < v$.
- 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.

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

By the compact form of the Law of Total Expectation
\[ \mathbb{E}[X|U] = \frac{d^+(U)}{d(U)} \cdot 2d(U) = 2d^+(U). \]
\[ \mathbb{E}[X] = \mathbb{E}[2d^+(U)] = 2 \sum_{u \in V} \frac{1}{n} \cdot d^+(u) = \frac{2m}{n} = \bar{d} \]

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

- Let $d^+(u)$ denote the number of neighbors $v$ of $u$ with $u < v$.
- Let $H \subseteq V$ be the set of the $\sqrt{2m}$ vertices with highest rank according to $<$. 
- Let $L = V \setminus H$.

### Observations

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. For all $v \in H$, $d^+(v) &lt; \sqrt{2m}$.</td>
<td>vertices with the highest degree</td>
</tr>
<tr>
<td>2. For all $v \in L$, $d(v) &lt; \sqrt{2m}$.</td>
<td></td>
</tr>
</tbody>
</table>

**Proof:**

1. $d^+(v)$ is the number of neighbors of $v$ of rank higher than $v$.
   
   If $v \in H$, it 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 $\sqrt{2m} \cdot d(v)$.

   That is, $d(v) < \frac{2m}{\sqrt{2m}} = \sqrt{2m}$
Analysis: Variance

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

By the compact form of the Law of Total Expectation

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

\[
\mathbb{E}[X^2] = \mathbb{E}[4d^+(U) \cdot d(U)] = 4 \sum_{u \in V} \frac{1}{n} \cdot d^+(u) \cdot d(u)
\]

\[
= \frac{4}{n} \left( \sum_{u \in H} d^+(u) \cdot d(u) + \sum_{u \in L} d^+(u) \cdot d(u) \right)
\]

\[
\leq \frac{4}{n} \left( \sum_{u \in H} \sqrt{2m} \cdot d(u) + \sum_{u \in L} d^+(u) \cdot \sqrt{2m} \right)
\]

\[
\leq \frac{4\sqrt{2m}}{n} \left( \sum_{u \in H} d(u) + \sum_{u \in L} d(u) \right) = 4\sqrt{2m} \cdot \bar{d}
\]

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

Observation:
- \( \forall v \in H, d^+(v) < \sqrt{2m} \).
- \( \forall v \in L, d(v) < \sqrt{2m} \).
Analysis: Putting It All Together

Algorithm (Input: $\varepsilon, n$; 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 $u < v$, set $X_i = 2d(u)$.
3. Return $\hat{d} = \frac{1}{k} \cdot \sum_{i \in [k]} X_i$.

- $\mathbb{E}[\hat{d}] = \mathbb{E}[X] = \tilde{d}$
- $\text{Var}[\hat{d}] = \frac{\text{Var}[X]}{k} \leq \frac{4\sqrt{2m} \cdot \tilde{d}}{k}$
- $\Pr[|\hat{d} - \tilde{d}| \geq \varepsilon \cdot \tilde{d}] = \Pr[|\hat{d} - \mathbb{E}[\hat{d}]| \geq \varepsilon \cdot \tilde{d}] \leq \frac{\text{Var}[\hat{d}]}{(\varepsilon \cdot \tilde{d})^2}$

\[
\frac{4\sqrt{2m} \cdot \tilde{d}}{k \cdot \varepsilon^2 \cdot \tilde{d}^2} = \frac{4\sqrt{2m} \cdot n}{k \cdot \varepsilon^2 \cdot 2m} = \frac{4n}{k \cdot \varepsilon^2 \cdot \sqrt{2m}} = \frac{4\sqrt{n}}{k \cdot \varepsilon^2 \cdot \sqrt{d}} \leq \frac{1}{3} \leq \frac{1}{3}
\]
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 $\Pr[|\hat{d} - \bar{d}| \geq \varepsilon \cdot \bar{d}] \leq \frac{1}{3}$