Lecture 3

Estimating Average Degree
Approximating Average Degree

\[
\overline{d} = \frac{\sum_{u \in V} d(u)}{n}
\]

Assume: 
- \( G \) simple (no parallel edges, self-loops)
- \( \Omega(n) \) edges (not "ultra-sparse")

Representation: adjacency list + degrees

\[
\begin{align*}
\text{d(v)} & \quad \text{node v} \\
3 & \quad 1 \\
1 & \quad 2, 3, 7 \\
\vdots & \quad \vdots \\
2 & \quad n-1, n \\
\end{align*}
\]

- Degree queries: on \( v \) return \( d(v) \)
- Neighbor queries: for \( (v_i, j) \) return \( j \)th nbr of \( v_i \)

Noise Sampling:

Pick \( \tilde{\mathcal{N}} \) sample nodes \( v_1, \ldots, v_\tilde{\mathcal{N}} \)

Output \( \frac{1}{\tilde{\mathcal{N}}} \sum_{i=1}^{\tilde{\mathcal{N}}} d(v_i) \) (ave degree of sample)

Using straightforward Chernoff/Hoeffding \( \Rightarrow \Omega(n) \) samples needed
Degree sequences are special?

\((n-1, 0, 0, 0, \ldots, 0)\) not possible

\((n-1, 1, 1, \ldots, 1)\) is possible

Some lower bounds:

"Ultrasparse case":

\[
\begin{align*}
\text{need linear time to get any multiplicative approx} \\
\text{graph with 0 edges vs. graph with 1 edge} \\
\text{ave deg = 0} \\
\text{ave deg = } \frac{1}{n}
\end{align*}
\]

\[\text{need } \Omega(n) \text{ queries to distinguish}\]

\[
\begin{align*}
\text{ave deg } \geq 2:\ \\
n\text{-cycle } d = 2 \\
n\text{-}cn^{1/2}\text{ cycle } d = 2 + c^2 \\
+ cn^{1/2}\text{-clique}
\end{align*}
\]

\[\text{need } \Omega(n^{1/2}) \text{ queries to find clique node}\]
Algorithm idea:

- group nodes of similar degrees
- estimate average within each group
- each group has bounded variance
- doesn't work for estimating average of arbitrary numbers, why should it work here?

Bucketing:

- set parameters $\beta = \frac{\varepsilon}{C}$
- $t = O\left(\log n / \varepsilon\right)$
- $\# buckets$

$B_i = \{ v \mid (1+\beta)^{i-1} \leq d(v) \leq (1+\beta)^i \}$

for $i \in \{0 \ldots (t-1)\}$

Note:

- total degree of nodes in $B_i$
  
  $$(1+\beta)^{i-1} |B_i| \leq d_{B_i} \leq (1+\beta)^i |B_i|$$

- total degree of graph
  
  $$\sum_{i} (1+\beta)^{i-1} |B_i| \leq d_{total} \leq \sum_{i} (1+\beta)^i |B_i|$$
First idea for algorithm:

- Take sample $S$ of nodes
- $S_i = S \cap B_i$ (samples that fall in $i$th bucket use degree queries to determine this)
- Estimate average degree contribution from $B_i$ using $S_i$
  - i.e. $\rho_i = \frac{|S_i|}{|S|}$
  - Output $\sum_i \rho_i (1+\beta)^{i-1}$

\[ E[\rho_i] = E\left[\frac{|S_i|}{|S|}\right] = \frac{|B_i|}{|S|} \]

Problem:

- $i$ st. $|S_i|$ is small likely come from $i$ st. $|B_i|$ small
  - Example of problem:

\[ \xrightarrow{3 \text{ nodes, deg } n-3} \]
\[ \xrightarrow{n-3 \text{ nodes, deg } 3} \]

- $a$ in $i$ st. $(1+\beta)^{i-1} \leq 3 \leq (1+\beta)^i$
  - $|B_a| = n-3$ contributes $(n-3)3$ edges
- $b$ in $i$ st. $(1+\beta)^{i-1} \leq n-3 \leq (1+\beta)^i$
  - $|B_b| = 3$ contributes $3(n-3)$ edges

- $c \neq a,b \quad |B_c| = 0$

Still, maybe good enough for $2$-approximation?
Next idea: use "0" for small buckets

Algorithm:
- sample $S$
- $S_i = S \cap B_i$
- For all $i$
  - if $|S_i| > \frac{\varepsilon}{\ln n} \cdot \frac{|S|}{C\cdot t}$
    - use $p_i = \frac{|S_i|}{|S|}$
    - $\Rightarrow$ Call $i$ "big"
  - else $p_i = 0$
    - $\Rightarrow$ Call $i$ "small"
- output $\sum \limits_i p_i (1 + \beta)^{i-1}$

Analysis:
1) Output not too large

idealistic (but unrealistic) case
- Suppose $\forall i$ $p_i = \frac{|B_i|}{n}$, then $\sum \limits_i p_i (1 + \beta)^{i-1} = \sum \limits_i \frac{|B_i|}{n} (1 + \beta)^{i-1}$
- $\sum \limits_i \frac{|B_i|}{n} (1 + \beta)^{i-1} \leq d (1 + \gamma)$

realistic case
- Suppose $\forall i$ $p_i \leq \frac{|B_i|}{n} (1 + \gamma)$
- $\Rightarrow \sum \limits_i p_i (1 + \beta)^{i-1} \leq d (1 + \gamma)$
2) Can output be too small?

\[
\forall i \quad \rho_i = \frac{|B_i|}{n} \quad \text{then} \quad \sum_i \rho_i (1 + \rho)^i = \sum_i \frac{|B_i|}{n} (1 + \beta)^i \\
\text{(since multiply by} \quad \frac{1}{(1 + \rho)(1 - \rho)} \leq 1) \\
\geq (1 - \beta) \sum_i \frac{|B_i|}{n} (1 + \beta)^i \\
\geq (1 - \beta) \frac{d}{\beta_i} \quad \text{deg of node in} \quad \beta_i
\]

By sampling, for big \( i \), \( \rho_i \geq \frac{|B_i|}{n} (1 - \delta) \)

For small \( i \) ????

**How much undercounting?**

Divide edges into 3 types:

1) **big-big** - both endpoints in big buckets counted twice
2) **big-small** - one endpt in big bucket counted once
3) **small-small** - both endpoints in small buckets never counted

[See example]

**Note:** big-big & big-small get counted (off by factor of two) but small-small can be a real problem
Example

ave deg 5
bucket a
big

ave deg 7
bucket b
small

ave deg 4
calendar
small

Total degree
LHS of bipartite $\mathcal{R}_5$ of bipartite $\mathcal{S}_5$-clique
$5(n-8) + (n-8)(3) + 4.5 = 8(n-8)+30$
ave deg $\approx 5n$
Algorithm will output $\approx 5n$

Samples

$6, 109, 037$
$157, 74$
$+1, ...$
bucket a

$\phi$
bucket b

$\phi$
bucket c

$\Rightarrow (\text{whp})$ bucket a is big, in fact,
$\text{whp } p_a < 1$

$\Rightarrow (\text{whp})$ bucket b, c are small
$p_b \approx 0, p_c \approx 0$

$\Rightarrow$ most nodes here

# big-small edges (of: $3(n-8)$)
Fraction of big-big + big-small $\approx \frac{3(n-8)}{5(n-8)} = \frac{3}{5}$

$E[a_j] = \frac{3}{5}$
Output $\approx 1 \cdot (1 + \frac{3}{5}) \cdot (1 + \frac{8}{5}) \approx 5$
Good news: Small buckets could have too many nodes

\[ |B_i| > \frac{2 \sqrt{\pi n}}{ct} \]

\[ \implies \text{can bound total # small-small edges} \]

If \( |B_i| > \frac{2 \sqrt{\pi n}}{ct} \)

then Expected size of \( S_i \) is \( \geq |S_i| \cdot \frac{|B_i|}{n} \)

\[ \geq |S_i| \cdot 2 \cdot \frac{\sqrt{\pi}}{ct} \]

So, very likely algorithm will decide via Chernoff bounds that \( i \) is "big"

So, assume \( |B_i| \leq \frac{2 \sqrt{\pi n}}{ct} \) for all \( i \) "small"

then total # small-small edges

\[ \leq \left( \frac{2 \sqrt{\pi n}}{ct} \right)^2 \]

\[ \leq \frac{4 \pi n}{c^2} \]

\[ \leq O\left( \frac{\pi n}{c^2} \right) = O\left( \frac{\pi n}{c^2} \right) \]

if we ignore them, they affect approx of \( \mathcal{A} \) by \( \leq (1+\epsilon) \) multiplicative factor

\[ \leq \epsilon n \] additive factor

First Claim:

Algorithm almost gives factor 2 multi approx

since large-small underestimated by \( \leq \) factor \( \frac{1}{2} \)

we get \( (2+\epsilon) \) multiplicative approx

\[ \uparrow \text{large-small error} \]

\[ \rightarrow \text{small-small error} \]
Improving further:

need to do better on "big-small" edges...

can we estimate the fraction of them & correct for them?

Can do via sampling if we can pick a "random" edge

New queries:

random neighbor query \( (v) \):
- given \( v \), return random nbr of \( v \)
- implementation:
  1. degree query for \( v \)
  2. pick random \( i \in [1..\text{deg}(v)] \)
  3. neighbor query \( (v,i) \)

pick (almost) random edge in (big)bucket \( i \):
- pick random edge by sampling nodes until one falls in bucket \( i \)
- return random nbr query from that node

Estimate fraction big-small in \( B_\alpha \text{ (big)} \):

repeat \( O(1/\delta) \) times:
- pick random node \( u \in B_\alpha \)
- \( e \leftarrow \text{random nbr of } u \) \( l \) if \( e \) is "big-small"
  - set \( a_l \) to be \( 1 \) if \( e \) is "big-small"
  - 0 o.w. (\( e \) is "big-big")

Output \( \alpha_\alpha = \text{average } a_l \)
Analysis:

Easy case: All nodes in $B_i$ have same degree $d$

\[ T_i < \# \text{ "big-small" edges in } B_i \]

\[ \Pr[\text{ "big-small" edge } e \text{ in } B_i \text{ chosen}] = \frac{1}{|B_i|} \cdot \frac{1}{d} \]

so \[ \Pr[a_j = 1] = E[a_j] = \frac{T_i}{d \cdot |B_i|} \]

general case: all nodes in bucket $B_i$ have degree within $(1+\beta)$ factor of each other

\[ \frac{1}{|B_i|(1+\beta)^{x}} \leq \Pr[\text{ "big small" edge } e \text{ in } B_i \text{ chosen}] \leq \frac{1}{|B_i|(1+\beta)^{x-1}} \]

\[ \frac{T_i}{|B_i|(1+\beta)^{x}} \leq E[a_j] \leq \frac{T_i}{|B_i|(1+\beta)^{x-1}} \Rightarrow E[a_j] |B_i|(1+\beta)^{x-1} \leq T_x \leq E[a_j] |B_i| (1+\beta)^{x-1} \]

\[ \frac{T_i}{n} \] estimate to $(1+\varepsilon)$-multi factor to get $(1+\varepsilon)(1+\beta)^{x-1}$ estimate of $\frac{T_i}{n}$ via $\frac{T_i}{n} \cdot (1+\varepsilon)^{x-1}$
Final Algorithm:

- Sample $\Theta\left(\frac{n}{e^2}\right)$ nodes + place in $S \leftarrow O\left(\frac{n}{e^2}\right)$ samples
- $S_1 \leftarrow S \cap B$,
- For all $i$
  
  if $S_i \geq \frac{1}{\epsilon^3} \frac{|S_1|}{ct}$
  
  use $p_i \leftarrow \frac{|S_i|}{|S_1|}$

  For all $v \in S_i$
  - Pick random nbr $u$ of $v$
  - $\omega(v) \leftarrow \frac{1}{0.1}$ if a small
  
  $\omega_i \leftarrow \frac{1}{|S_i|} \sum_{v \in S_i} \omega(v)$

  else use $p_i \leftarrow 0$

- Output $\sum_{i \text{ large}} p_i (1 + \omega_i) (1 + \rho)^{i-1}$

includes big-big + one side of big-small
other side of big-small

Where do errors come from?

- estimating $p_i$'s ~ multiplicative $(1+\epsilon)$ factor
- estimating $\omega_i$'s ~ additive
- small - small edges ~ additive $\epsilon \cdot n$ additive error