Lecture 3:

- Estimate average degree
  - recap
  - 2-approximation
  - 1+3-approximation
Estimating the average degree of a graph

def Average degree \( \bar{d} = \frac{\sum d(u)}{n} \)

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

Representation via adj list + degrees:

\[
\begin{align*}
\text{d(v)} & \quad \text{node v} \\
3 & \quad 1 \quad \rightarrow \quad [2, 5, 7] \\
1 & \quad 2 \quad \rightarrow \quad \{1\} \\
\vdots & \quad \vdots \\
2 & \quad \vdots \\
1 & \quad \vdots \\
\end{align*}
\]

- degree queries: on \( v \) return \( d(v) \)
- neighbor queries: on \((v, j)\) return \( j \)th \( \text{nbr} \) of \( v \)
Naive sampling:

Pick \( O(\log n) \) sample nodes \( V_1, \ldots, V_s \)

Output avg degree of sample:

\[
\frac{1}{s} \leq d(V_i)
\]

Straightforward Chernoff/Howlett needs \( \Omega(n) \) samples

Lower bound?

\[
d(1) \  \delta(2) \ \ldots \ \delta(k) \\
0 \ 0 \ 0 \ \delta(n) \ 0 \ 0 \ 0
\]

Need \( \Omega(n) \) samples to find "needle in haystack"

Not a possible degree sequence!!

\[n-1\underbrace{1\quad 1\quad 1\quad 1\quad 1\quad 1\quad 1}_{\text{is possible}}\]
Some lower bounds:

"Ultra-sparse" case:

0 edges vs. 1 edge

need $\Omega(n)$ queries to distinguish

$\Rightarrow$ multiplicative approx needs $\Omega(n)$

ave $\text{deg} \geq 2$:

\[ n - \text{cycle} \quad \overline{d} = 2 \]

vs.

\[ n - C \cdot \sqrt{n} \text{ cycle} \quad \overline{d} \sim 2 + c^2 \]

$+ \sqrt{n} - \text{clique}$

need $\Omega(n^{\frac{1}{2}})$ queries to find clique node
Algorithm idea:

group nodes of similar degrees
estimate average within each group

why does this help?
recall Chernoff:

\[ X_1, \ldots, X_r \text{ iid } X_i \in \{0,1\} \]
\[ S = \sum_{i=1}^{r} X_i \quad p = E[X_i] = E[S]/r \]
\[ \text{Then } \quad \Pr\left[\left| \frac{S}{r} - p \right| \geq \delta p \right] \leq e^{-\Omega(r\delta^2)} \]

\[ \Rightarrow \quad r \text{ needs to be } \Omega\left(\frac{1}{\delta^2 p}\right) \]

 lets assume \( \delta \) is a constant

\( X_i \) needs to be in \([0, 1]\)

so if \( X_i \leq \frac{\deg(i)}{n} \)

than \( p \) can be as small as \( \frac{1}{n} \)

\[ \Rightarrow \quad r \text{ needs to be } \Omega\left(\frac{1}{p}\right) = \Omega(n) \]

but if \( b \leq \deg(i) \leq (1+\varepsilon)b \)

can set \( X_i = \frac{\deg(i)}{(1+\varepsilon)b} \)

then \( p = \frac{1}{1+\varepsilon} \)

\( \Rightarrow \) \( r \) needs to be only \( \Omega(1) \). Much better!!!
+ each group has bounded variance
- doesn’t work for arbitrary #’s
  why here?

**Bucketing:**

Set parameters

\[ \beta = \frac{\varepsilon}{c} \]

\[ t = O(\log n/\varepsilon) \quad \# \text{buckets} \]

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

\[ \text{for } i \in \{0, \ldots, (t-1)\xi\} \]

(can add bucket for deg 0 nodes

or assume none)

Note: total degree of nodes in \( B_\xi \)

\[ (1+\beta)^\xi \cdot |B_\xi| \leq d_{B_\xi} \leq (1+\beta)^\xi \cdot |B_\xi| \]
Total degree of graph:
\[ \sum_{i} (1+\beta)^{-i} |B_{i}| \leq d_{\text{total}} \leq \sum_{i} (1+\beta)^{-i} |B_{i}| \]

First idea for algorithm:

- Take sample \( S \) of nodes
  \[ S_{i} \leftarrow S \cap B_{i} \]
- Estimate \( |B_{i}| \):
  \[ \rho_{i} \leftarrow \frac{|S_{i}|}{|S|} \]
- Output \( \sum_{i} \rho_{i} (1+\beta)^{-i} \)

Problem:

If \( i \) is s.t. \( |S_{i}| \) small, will need lots of samples to approximate. These likely come from \( B_{i} \) s.t. \( |B_{i}| \) is small.
Example:

\[ a \leftarrow \text{i.s.t. } (1+\beta)^{i-1} \leq 3 \leq (1+\beta)^i \]

\[ b \leftarrow \text{i.s.t. } (1+\beta)^{i+1} \leq n-3 \leq (1+\beta)^i \]

\[ \forall c \neq a, b \quad |B_c| = 0 \]

\[ |B_a| = n-3 \]

\[ |B_b| = 3 \]

both contribute (n-3):3 edges

but these are not likely to be sampled

Still, maybe good enough for 2-approximation?

Next idea:

Use "0" for small buckets
**Old algorithm:**

- Take sample $S$
- $S_i \leftarrow S \cap B_i$
- estimate $|B_i|$
  $$\rho_i \leftarrow \frac{|S_i|}{|S|}$$
- Output $\sum_i \rho_i (1 + \beta)^{i-1}$

**New algorithm:**

- Take sample $S$
- $S_i \leftarrow S \cap B_i$
- estimate $|B_i|$
  for all $i$
  if $|S_i| \geq \frac{\sqrt{\frac{3}{n}} \cdot |S|}{C \cdot t}$
    use $\rho_i \leftarrow \frac{|S_i|}{|S|}$
  else $\rho_i \leftarrow 0$
- Output $\sum_i \rho_i (1 + \beta)^{i-1}$

**Why $\sqrt{\frac{3}{n}} \cdot \frac{|S|}{C \cdot t}$?**

let $|S| = \Theta (\sqrt{n \text{ polylog } n \cdot \text{ poly } \epsilon})$

then $|S_i| \geq \frac{\sqrt{\frac{3}{n}} \cdot |S|}{C \cdot t} \Rightarrow |S_i| \geq \Omega (\text{polylog } n \cdot \text{poly } \epsilon)$

\[
\Rightarrow \forall i \quad (1 - \delta) \frac{|B_i|}{n} \leq \rho_i \leq (1 + \delta) \frac{|B_i|}{n} \quad \text{for } \delta \approx \Theta(\epsilon)
\]
Why these settings of $S$? (ignore dependence on $\varepsilon$ for now)

* each bucket that has at least $\approx \frac{1}{\sqrt{n}}$ fraction of nodes should have *enough* samples to be able to estimate the fraction.

* why $\approx \frac{1}{\sqrt{n}}$?

- we will want to argue that "small" buckets represent a very small fraction of the edges so it is OK to zero them out.

- remember the clique lower bound example? if we set the $\frac{1}{\sqrt{n}}$ "small" threshold to bigger than $\sqrt{n}$ we might miss lots of edges (e.g. a clique on $\sqrt{n}$ nodes will have $\Theta(n)$ edges & shouldn't be missed but represents only $\frac{1}{\sqrt{n}}$ fraction of nodes).
- Why is $\frac{1}{\sqrt{n}}$ small enough? See later!

* What is "enough" samples for each bucket?

- We will need to argue that we are getting good estimates of $\frac{|B_i|}{n}$ for each big bucket.

  \[ \text{union bound over log n buckets} \]

  So need prob of having bad estimate $\delta$ set to $\epsilon < \frac{1}{\text{log n}}$ per bucket.

  Chernoff will also depend on accuracy parameter $\beta = \frac{\epsilon}{c}$.

  \[ \text{So if we set } S \sim \sqrt{n} \cdot \text{poly}(\frac{\delta}{\epsilon}) \cdot \text{poly}(\text{log n}) \]

  \[ \text{we should be more than ok} \]

  \[ \text{this comes in everywhere to satisfy Chernoff \& union bounds} \]
Analysis:

1) Output not too large:

**Idealistic case**
Suppose \( \forall i \ \ p_i = \frac{|B_i|}{n} \),
then \( \sum_i p_i (1+\beta)^i \cdot \frac{|B_i|}{n} \leq \deg \text{ of nodes in } B_i \)

**Realistic case**
Suppose \( \forall i \ \ p_i \leq \frac{|B_i|}{n} (1+\beta) \), e.g. when \( i \) is big
\[ \Rightarrow \sum_i p_i (1+\beta)^{i-1} \leq d(1+\beta) \]

2) Can output be too small?

if \( \forall i \ \ p_i = \frac{|B_i|}{n} \) then \( \sum_i p_i (1+\beta)^{i-1} = \sum_i \frac{|B_i|}{n} (1+\beta)^{i-1} \)
multiply by \( \frac{(1+\beta)(1-\beta)}{\beta} \leq 1 \)
\[ \geq (1-\beta) \sum_i \frac{|B_i|}{n} (1+\beta)^i \]
\[ \geq (1-\beta) d \]
by sampling, for big i, \( p_i \geq \frac{|B_i|}{n} (1 - \delta) \)

for small i ?? ??

How much undercounting?

divide edges into 3 types

1) big-big: both endpts in big buckets Counted twice

2) big-small: one endpt in big bucket " " " small " Counted once

3) small-small: both endpts in small buckets "never" counted

Note: small-small can be a big problem

big-small only undercounted by a factor of 2
Example:

Total degree: \(5 \cdot (n-8) + (n-8) \cdot 3 + 4 \cdot 5 = 8(n-8) + 20\)

Average degree \(\approx 8\)

Algorithm will likely output \(\approx 5\)
Example:

- **big-big**
  - 5 nodes
  - Ave degree 4
  - bucket a
    - big
  - bucket b
    - small
  - 3 nodes

New algorithm:

- Take sample $S$ (how big?)
  - $S_i \in S \cap B_i$
- Estimate $|B_i|$:
  - For all $i$
    - if $151 \geq \frac{3}{n} \cdot \frac{|B_i|}{c+1}$
      - use $p_i \leq \frac{|B_i|}{151}$
    - else $p_i = 0$
- Output $\leq \prod_i p_i (1 - p_i)^{-1}$

Samples:

- $\{6, 104, 22\}$
- $157, 74, 41, \ldots$

- bucket a
  - most nodes here
  - $whp$ $p_a \approx 1$
- bucket b
  - $\emptyset$
  - Few, if any, in these buckets
  - $whp$ $b + c$ are small so likely
  - that $p_b = p_c = 0$
- bucket c
  - $\emptyset$

Output $\approx 1.5$
Good news:

Small buckets can't have many nodes

⇒ bound on total # small-small edges

\[ |B_i| > \frac{2\sqrt{\varepsilon n}}{ct} \quad \text{then expected size of } S_i \]

\[ \geq |S| \cdot \frac{|B_i|}{n} \]

\[ \geq |S| \cdot 2\sqrt{\varepsilon n} \cdot \frac{1}{ct} \]

3/2 twice threshold for "big"

so likely algorithm will decide that i "big"

Assume for all i "small" that \( |B_i| \leq \frac{2\sqrt{\varepsilon n}}{ct} \)

then total # small-small edges

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

\[ \approx O(\frac{\varepsilon n}{c^2}) = O(\varepsilon n) \]
If ignore small-small edges, 
they affect approx of $\bar{d}$ 
by $\leq \frac{3 \varepsilon n}{n} = 3 \varepsilon$ additive factor 
$\leq (3+1) = 4$ multiplicative factor 

First Claim:

Algorithm gives factor $(2+\varepsilon)$-mult approx

Improved further:

need to improve on "big-small" edges
Can we estimate fraction of them and correct for them?

e.g. by sampling random edges?
New queries:

**Random neighbor query (v):**
- given v, return random nbr of v

implementation:
1. degree query to v
2. pick random i ∈ [1..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 $B_i$:

repeat $O(1/n)$ times

pick random node $u \in B_i$

$e \leftarrow$ random nbr of $u$

set $a_j$ to be $\{1$ if $e$ "big-small"

$\{0$ o.w.

(e is "big-big")

Output $\bar{a}_i =$ average $a_j$

Analysis:

easy case: all nodes in $B_i$ have same degree

$T_i = \# \text{ big-small edges in } B_i$

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

only one of $uv$ big

$\Pr[a_j=1] = \frac{T_i}{d \cdot |B_i|}$
general case: all nodes in $B_i$ have degrees within $(1+\beta)$ factor of each other

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

$$\frac{T_i}{|B_i| (1+\beta)^i} \leq E[a_j] \leq \frac{T_i}{|B_i| (1+\beta)^{i-1}}$$

estimate to $1+\varepsilon$-mult factor

to get

$(1+\varepsilon)(1+\beta)$ estimate

of $\frac{T_i}{n}$ via $\lambda_i p_i (1+\beta)^{i-1}$

undercount of # edges in $B_i$
Example:

- **big-bi**
- **big-sm**
- **small-s**

**Nodes:**
- 5 nodes
- Ave. deg: 4
- Bucket C: small

**Nodes:**
- n-4
- n-3
- n-2
- n-1
- n

**Small-Small**

**Total degree:** $5 \cdot (n-8) + (n-8) \cdot 3 + 4 \cdot 5 = 8(n-8) + 20$

**Ave. degree ≈ 8**

Algorithm will likely output ≈ 5

\# big-small edges slots: $3 \cdot (n-8)$

Fraction of big-big over big-small $\approx \frac{3(n-8)}{5(n-8)} = \frac{3}{5}$

$E[a_j] = \frac{3}{5}$

Output $1 \cdot \left(1 + \frac{3}{5}\right)^2 \approx 8 \approx 5$
Final Algorithm:

- Sample \( \Theta(\sqrt{n}/\epsilon^2) \) nodes and place in \( S \)

- \( S_i \leftarrow S \setminus B_i \)

- For all \( i \):
  
  - If \( |S_i| \geq \sqrt{\frac{3}{n}} \frac{|S|}{\epsilon} \) use \( p_i \leftarrow \frac{|S_i|}{|S|} \)
  
  - For all \( r \in S_i \):
    
    - Pick random nbr \( u \) of \( r \)
    
    - \( \chi(r) \leftarrow \begin{cases} 1 & \text{if } u \text{ is small} \\ 0 & \text{otherwise} \end{cases} \)

- \( \alpha_i \leftarrow \frac{\sum_{r \in S_i} |r\setminus |S_i|} {|S_i|} \)

- Else use \( p_i \leftarrow 0 \)

- Output \( \sum_{\text{large } i} p_i (1 + \alpha_i) (1 + \beta)^{i-1} \)

  - \( \epsilon \) correction to get other side of big-small
  
  - One side of big-small
Where do errors come from?

- Estimating $p_i$ and $x_i$'s
- Estimating $d_i$'s
- Small small edges

$\exists$ multiplicative factor
$\exists$ additive error