Lecture 3:

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

\[ \text{Average degree } \bar{d} = \frac{\sum d(u)}{n} \]

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

Representation via adj list + degrees:

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

Pick $O(n^2)$ sample nodes $V_1 \ldots V_s$

Output ave degree of sample:

$$\frac{1}{s} \sum_{i} d(V_i)$$

Straightforward Chernoff/Hoeffding needs $\Omega(n)$ samples

Lower bound?

\[
\begin{array}{cccc}
  d(1) & d(2) & \ldots & d(n) \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

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

Not a possible degree sequence!!

\[
\begin{array}{cccc}
  n & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Is possible
Some lower bounds:

"Ultrasparsse" case:

0 edges vs. 1 edge

need $\Omega(n)$ queries to distinguish

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

ave deg $\geq 2$:

n-cycle $\bar{d}=2$

$\quad$ vs.

$n-C\cdot \sqrt{n}$ cycle $\bar{d} \sim 2+c^2$

+ $\sqrt{n} - $ clique

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

$\Rightarrow$ need $\Omega(n^{\frac{1}{2}})$ queries for constant multip approx
Algorithm idea:

group nodes of similar degrees

estimate average within each group

why does this help?

recall Chernoff:

\[ X_1, \ldots, X_r \text{ iid } X_i \in [0,1] \]
\[ S = \sum_{i=1}^{r} X_i \quad p = \mathbb{E}[X_i] = \mathbb{E}[S]/r \]

Then
\[ \Pr \left[ \left| \frac{S}{r} - p \right| \geq \delta p \right] \leq e^{-\Omega(\delta^2 p \log(p))} \]

\[ \Rightarrow r \text{ needs to be so } p \text{ very small is not good!} \]

\[ \Omega \left( \frac{1}{p \delta^2} \right) \]

\[ \text{lets assume } \delta \text{ is a constant} \]

\[ X_i \text{ needs to be in } [0,1] \]

\[ \text{so if } X_i \leq \frac{\text{deg}(i)}{n} \]
\[ \text{then } p \text{ can be as small as } \frac{1}{n} \]

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

\[ \text{but if } b \leq \text{deg}(i) \leq (1+\varepsilon)b \]
\[ \text{can set } X_i \leq \frac{\text{deg}(i)}{(1+\varepsilon)b} \]

\[ \text{as at least a constant } \Rightarrow p \geq \frac{1}{1+\varepsilon} \]

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

Bucketing:

Set parameters $\beta = \frac{\epsilon}{2}$

$$t = O\left(\log n \left| \epsilon \right|\right) \text{ # buckets}$$

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

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

(can add bucket for deg 0 nodes)

(or assume none)

Note: total degree of nodes in $B_i$

$$(1+\beta)^{i-1} |B_i| \leq \sum_{v \in B_i} d_v \leq (1+\beta)^i |B_i|$$
total degree of graph: \[
\sum_{t=1}^{t} (1+\beta)^{i-1} |B_i| \leq d_{\text{total}} \leq \sum_{i=1}^{t} (1+\beta)^{i-1} |B_i|
\]

First idea for algorithm: \[B_i = \sum_{v \mid (1+\beta)^{i-1} < d(v) \leq (1+\beta)^{i}}\]

- Take sample \( S \) of nodes
- \( S_i \leftarrow S \cap B_i \) - use degree queries to partition
- estimate \( |B_i| \):
  \[p_i \leftarrow \frac{|S_i|}{|S|}
  \]
- Output \( \sum_i p_i (1+\beta)^{i-1} \)

Problem: \( i \) s.t. \( |S_i| \) is small

\[\Rightarrow p_i \] is a bad approx
Example:

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

\[ \leftarrow 3 \text{ nodes each deg } n-3 \]

\[ \leftarrow n-3 \text{ nodes each deg } 3 \]

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

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

\[ \forall c \neq a, b \quad \text{or } B_c = \emptyset \]

\[ |B_a| = n-3 \quad \exists \quad \text{both contribute} \]

\[ |B_b| = 3 \quad (n-3) \cdot 3 \text{ edges} \]

\[ B_a \text{ contributes } (n-3) \cdot 3 \text{ edges} \]

\[ B_b \text{ contributes } 3 \cdot (n-3) \text{ edges} \]

\[ \text{not seen in sample of size } o(n) \]

Next idea: use \( \Theta \) for small buckets

(\( \text{helps ''variance''} \))
Old algorithm:

- Take sample $S$
- $S_{x} \leftarrow S \land B_{x}$

- estimate $|B_{x}|$
  
  $p_{x} \leftarrow \frac{|S_{x}|}{|S|}$

- Output $\sum_{i} p_{x} (1+\beta)^{i-1}$

New algorithm:

- Take sample $S$
- $S_{x} \leftarrow S \land B_{x}$

- estimate $|B_{x}|$
  
  for all $i$
  
  if $|S_{x}| \geq \sqrt{\frac{3}{n}} \cdot \frac{|S|}{C \cdot t}$
  
  use $p_{x} \leftarrow \frac{|S_{x}|}{|S|}$

  else $p_{x} \leftarrow 0$

- Output $\sum_{i} p_{x} (1+\beta)^{i-1}$

- how big is $S$?

- why $\sqrt{\frac{3}{n}} \cdot \frac{|S|}{C \cdot t}$?

  one of these comes from $t = O\left(\frac{\log n}{\delta_{\varepsilon}}\right)$

  let $|S| = \Theta\left(\sqrt{n} \cdot \text{poly log } n \cdot \text{poly } \frac{1}{\varepsilon}\right)$

  $\Rightarrow \bigg|S_{x}\bigg| = \sqrt{\frac{3}{n}} \cdot \frac{|S|}{C \cdot t} \geq \Omega\left(\text{poly log } n \times \text{poly } \frac{1}{\varepsilon}\right)$

  $\Rightarrow$ by union bound + Chernoff bound

Assume thus

$\forall i$ big $(1-\delta) \frac{|B_{x}|}{n} \leq p_{x} \leq (1+\delta) \frac{|B_{x}|}{n}$
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 "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 and shouldn’t be missed, but represents only $\frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}}$ 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

so need prob of having bad estimate \( \delta \) set to \( \leq \frac{1}{\log n} \) per bucket

Chernoff will also depend on accuracy parameter \( \beta = \frac{\epsilon}{c} \)

So if we set \( S \approx \sqrt{n} \cdot \text{poly} \left( \frac{1}{\log n} \right) \) we should be more than ok
Analysis:

1) Output not too large:

**Idealistic case**

Suppose \( \forall i \quad p_i \leq \frac{|B_i|}{n} \), then

\[
\sum_{i} p_i (1 + \beta)^{-i} = \sum_{i} \frac{|B_i|}{n} (1 + \beta)^{-i} \\
\leq \bar{d} 
\]

(\( \leq \deg \text{ of any node in } B_i \))

**Realistic case**

Suppose \( \forall i \quad p_i \leq \frac{|B_i|}{n} (1 + \delta) \)

\[
\Rightarrow \sum_{i} p_i (1 + \beta)^{-i} \leq \bar{d} (1 + \delta) 
\]

(note that we are assuming this for all big \( i \))

\( \forall \) for all small \( i \) we set \( p_i = 0 \)

2) Can output be too small?

if \( \forall i \quad p_i = \frac{|B_i|}{n} \) then

\[
\sum_{i} p_i (1 + \beta)^{-i} = \sum_{i} \frac{|B_i|}{n} (1 + \beta)^{-i} \\
\geq (1 - \beta) \sum_{i} \frac{|B_i|}{n} (1 + \beta) \\
\geq (1 - \beta) \bar{d} \geq \deg \text{ of any node in } B_i \]
by sampling, for big $i$, $p_i \geq \frac{|B_i|}{n} (1 - \delta)$

for small $i$???

$\beta \approx o(\log n/\delta)$

big

$$\frac{\log |B_i|}{n} \geq \frac{1}{\epsilon} \cdot \frac{1}{ct}$$

How much undercounting?

divide edges into 3 types

1) big-big: both endpoints in big buckets counted twice

2) big-small: one endpoint in big bucket counted once " " " small "

3) small-small: both endpoints in small buckets never counted

big-big ok

big-small undercounted by $\frac{1}{2}$ factor 2 approx

small-small not counted at all
Example:

\[ \begin{align*}
\text{n-8 nodes} & \quad \text{ave deg} 5 \\
\text{bucket a} & \quad \text{big} \\
\text{n-5} & \\
\end{align*} \]

\[ \begin{align*}
\text{big-big} & \\
4 & \text{big-small} \\
5 & \\
6 & \\
7 & \\
\vdots & \\
n & \\
\end{align*} \]

\[ \begin{align*}
\text{5 nodes} \\
\text{ave deg 4} \\
\text{bucket c small} \\
\end{align*} \]

\[ \begin{align*}
\text{n-4} & \quad \text{n-3} \\
\text{n-2} & \quad \text{n-1} \\
n & \\
\end{align*} \]

\[ \text{small-small} \]

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

Average degree \( \approx 8n \)

Algorithm will likely output \( \approx 5 \)
New algorithm:

- Take sample $S$ (how big?)
- $S_i \in S \cap B_i$
- estimate $|B_i|$
  - for all $i$
    - if $|S_i| \geq \frac{3}{9} \cdot \frac{|S|}{c-t}$
      - use $p_i \leq \frac{|S_i|}{|S|}$
    - else $p_i = 0$
  - Output $\leq \sum \frac{p_i}{(1-p_i)}^{i-1}$

$\begin{align*}
\text{Samples}: & \\
\{(6, 104, 22), 15, 7, 7, 4, 1, \ldots\} & \text{bucket a}
\end{align*}$

$\begin{align*}
\text{most nodes} & \\
\Rightarrow \text{whp bucket a} & \\
p_a \leq 1 & \\
\text{output} = 5
\end{align*}$
Good news:

Small buckets can't have many nodes

\[ \Rightarrow \text{bound on total } \pm \text{ small-small edges} \]

if \( |B_i| > \frac{2\sqrt{\epsilon n}}{ct} \) then expected size of \( S_i \) is

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

\[ \geq \frac{|S_i| \cdot 2 \cdot \frac{\sqrt{\epsilon n}}{n} \cdot \frac{1}{ct}}{n} \]

= twice the threshold for being big

so very likely that algorithm will decide \( i \) is big

(healing Chernoff + union bound)

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

then total \( \pm \) small-small edges is:

\[ \leq \left( \frac{2\sqrt{\epsilon n}}{ct} \right)^2 = O(\epsilon n) \]
if ignore small-small edges, they affect approx of $d$

$\exists \frac{d}{\geq 1}$

First Claim:

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

so far, all we have used are degree queries!
Improving further:

need to improve on "big-small" edges

Main idea:

double count from the big side!
New queries:

**Random neighbor query (v):**

- given v, return random nbr of v

Implementation:

1. degree query to v.
2. pick random \( i \in [1, \deg(v)] \)
3. neighbor query \((v, i)\)

1st use of nbr queries!

pick (almost) random edge in (big) bucket \( i \):

Sample nodes until fall into bucket \( i \)

Random nbr query from 1st node that falls in \( i \)
Estimate fraction big-small in $B_i (\text{big})$:

repeat $O(1/\delta)$ times:

pick random node $u \in B_i$

pick random nbr of $u$

set $a_j$ to be $\{1$ if $e$ "big-small" $\}$ $\{0$ o.w. $e$ "big-big" $\}$

Output $d_i = \text{average } a_j$

Analysis:

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

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

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

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

\[
\frac{\Gamma_i}{|B_i| (1 + \beta)^i} \leq E[c_{ij}] \leq \frac{\Gamma_i}{|B_i| (1 + \beta)^{i-1}}
\]
Example:

5 nodes

big-big

n-8 nodes

ave deg 5

bucket a

big

b

5

4

big-small

6

7

n-5

Small

3 nodes

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

ave degree \approx 8

algorithm will likely output \approx 5

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

Fraction of big-small over total: \approx \frac{3(n-8)}{5(n-8)} = \frac{3}{5} \quad E[A_j] = \frac{3}{5} \quad Output 1 \cdot \left(1 + \frac{3}{5}\right) \cdot \left(1 + \frac{3}{5}\right) \approx 8 \approx 5
Final Algorithm:

- Sample $\Theta(n^{1/2} \log n)$ nodes + place in $S$

- $S_1 \leftarrow S \cap B_i$

- For all $i$
  - If $|S_i| \geq \sqrt{\frac{3}{n}} \frac{|S_i|}{\log t}$
    - Use $p_i \leftarrow \frac{|S_i|}{|S_i|}$
    - For all $v \in S_i$
      - Pick random nbr $u$ of $v$
      - $x(v) \leftarrow \begin{cases} 1 & \text{if } u \text{ is small} \\ 0 & \text{otherwise} \end{cases}$
      - $\alpha_i \leftarrow \frac{\left| \{ u \in S_i \mid x(u) = 1 \} \right|}{|S_i|}$
    
  - Else use $p_i \leftarrow 0$

- Output $\sum_{\text{large } \lambda} p_i \left( 1 + \alpha_i \right) (1 + \beta)^{\lambda-1}$

  $\uparrow$ correction to get other side of big-small

  $\leftarrow$ big-big + one side of big-small
Where do errors come from?

\[ \text{estimate } \pi^3 \approx \text{mult } 1 + \varepsilon \]

\[ \alpha^3 \]

\[ \pm \text{ small small edges } \approx \text{ add } \varepsilon \]