Lecture 3:

- Estimate average degree
  - recap
  - 2-approximation
  - $1+\varepsilon$ - approximation

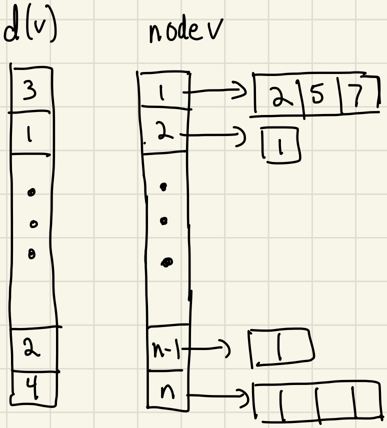
Estimating the average degree of a graph

<u>def</u> Average degree $\bar{d} = \dfrac{\sum_{u \in V} d(u)}{n}$

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} nbr of v

Naive sampling:

Pick $O(??)$ sample nodes $v_1 \cdots v_s$

output ave degree of sample:

$$\frac{1}{s} \sum_i d(v_i)$$

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

lower bound?

$d(1)$	$d(2)$	$\cdots$				$d(n)$
0	0	0	$n-1$	0	0	0

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

not a possible degree sequence!!

$n-1$	1	1	1	1	1	1

is possible

Some lower bounds:

"ultrasparse" case:

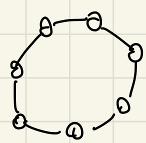
0 edges vs. 1 edge

need $\Omega(n)$ queries to distinguish

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

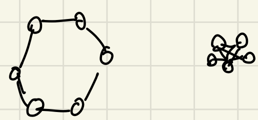
ave deg ≥ 2:

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



vs.

$n - c\cdot\sqrt{n}$ cycle $\bar{d} \approx 2+c^2$
$+ \, c\sqrt{n}$ - clique



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

Algorithm idea:

group nodes of similar degrees

estimate average w/in each group

why does this help?

recall Chernoff:

$X_1 \cdots X_r$ iid $X_i \in [0,1]$

$S = \sum_{i=1}^{r} X_i \quad p = E[X_i] = E[S]/r$

Then $\Pr[|\frac{S}{r} - p| \geq \delta p] \geq e^{-\Omega(rp\delta^2)}$

$\Rightarrow$ r needs to be

$\Omega(\frac{1}{p\delta^2})$

let's assume δ is a constant

X_i needs to be in $[0,1]$

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

then p can be as small as $\frac{1}{n}$

$\Rightarrow$ r needs to be $\Omega(1/p) = \Omega(n)$

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

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

then $p \geq \frac{1}{1+\varepsilon}$

$\Rightarrow$ r needs to be only $\Omega(1)$. Much better!!!

+ each group has bnded variance
− doesn't work for arbitrary θ's
why here?

Bucketing:

set parameters $\beta = \frac{\varepsilon}{c}$

$t = O(\log n / \varepsilon)$ # buckets

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

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

(can add bucket for deg 0 nodes
or
* assume none)

note: total degree of nodes in B_i

$$(1+\beta)^{i-1} |B_i| \leq d_{B_i} \leq (1+\beta)^i |B_i|$$

total degree of graph:

$$\sum_i (1+\beta)^{i-1} |B_i| \le d_{total} \le \sum_i (1+\beta)^i |B_i|$$

First idea for algorithm:

$\sigma_j^{(i)} = \begin{cases} 1 & \text{if sample } j \text{ falls in bucket } i \\ 0 & \text{o.w.} \end{cases}$

- Take sample S of nodes
- $S_i \leftarrow S \cap B_i$ (samples that fall in ith bucket. Use degree queries to determine)
- estimate $|B_i|$:

$$\rho_i \leftarrow \frac{|S_i|}{|S|}$$

note: $E[\rho_i] = E\left[\frac{|S_i|}{|S|}\right] = \frac{E\left[\sum_{j=1}^{|S|} \sigma_j^{(i)}\right]}{|S|}$

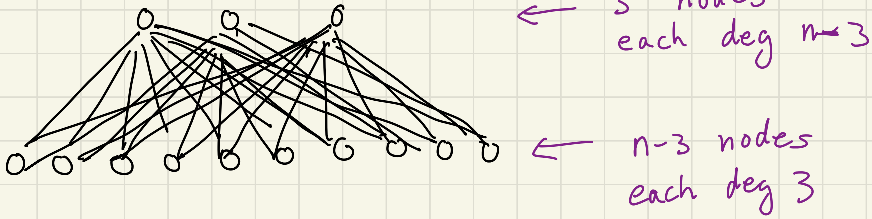
$$= \frac{|S|}{|S|} \cdot \frac{|B_i|}{n}$$

- Output $\sum_i \rho_i (1+\beta)^{i-1}$ ← undercounting

Problem:

if $\underbrace{i \text{ is s.t. } |S_i| \text{ small}}$, will need lots of samples to approx

these likely come from B_i s.t. $|B_i|$ is small example:



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

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

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

$$\left.\begin{array}{l} |B_a| = n-3 \\ \underbrace{|B_b| = 3} \end{array}\right\}$$ both contribute $(n-3)\cdot 3$ 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 $|B_i|$:

$$p_i \leftarrow \frac{|S_i|}{|S|}$$

- Output $\sum_i p_i (1+\beta)^{i-1}$

New algorithm:

- Take sample S (how big?)
- $S_i \leftarrow S \cap B_i$
- estimate $|B_i|$:

for all i

if $|S_i| \geq \sqrt{\frac{\varepsilon}{n}} \cdot \frac{|S|}{C \cdot t}$ — i "big"

use $p_i \leftarrow \frac{|S_i|}{|S|}$

else $p_i \leftarrow 0$ — i "small"

- Output $\sum_i p_i (1+\beta)^{i-1}$

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

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

then $|S_i| \geq \sqrt{\frac{\varepsilon}{n}} \frac{|S|}{C \cdot t} \Rightarrow |S_i| \geq \Omega(\text{polylog} n \times \text{poly} \frac{1}{\varepsilon})$

$\underset{\text{union bnd + Chernoff}}{\Longrightarrow}$ $\forall i \quad (1-\gamma) \frac{|B_i|}{n} \leq p_i \leq (1+\gamma) \frac{|B_i|}{n}$

for $\gamma \sim \Theta(\varepsilon)$

Why these settings of S? (ignore dependence on ε for now)

* each bucket that has at least $\approx \frac{1}{\sqrt{n}}$ fraction of <u>nodes</u> 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 <u>edges</u> so it is ok to zero them out

  - remember the clique lower bound example? if we set the "small" threshold to bigger than $\frac{1}{\sqrt{n}}$ 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 $\frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}}$ 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

Chernoff bound

union bound over $\log n$ buckets

so need prob of having bad estimate "δ" set to $\ll \frac{1}{\log n}$ per bucket

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

So if we set $S \approx \underbrace{\sqrt{n}}_{\text{to get buckets with } \frac{1}{\sqrt{n}} \text{ fraction of nodes}} \cdot \underbrace{\text{poly}\left(\frac{1}{\varepsilon}\right)}_{\text{this comes in everywhere}} \underbrace{\text{poly}(\log n)}_{\text{to satisfy Chernoff + union bnds}}$

we should be more than ok

Analysis:

1) Output not too large:

idealistic case

Suppose $\forall i$ $p_i = \frac{|B_i|}{n}$,

then $\sum_i p_i (1+\beta)^{i-1} = \sum_i \frac{|B_i|}{n} \underbrace{(1+\beta)^{i-1}}_{\leq \text{ deg of nodes in } B_i}$

$\leq \bar{d}$

realistic case

Suppose $\forall i$ $p_i \leq \frac{|B_i|}{n}(1+\gamma)$ e.g. when i is big

$\Rightarrow \sum_i p_i (1+\beta)^{i-1} \leq \bar{d}(1+\gamma)$

✓

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 $(1+\beta)(1-\beta) < 1$ $\longrightarrow$ $\geq (1-\beta)\sum_i \frac{|B_i|}{n}(1+\beta)^i$

$\geq (1-\beta)\bar{d}$

by sampling, for big i, $p_i \gtrsim \frac{|B_i|}{n}(1-\gamma)$

for small i ????

How much undercounting?

divide edges into 3 types

type determined by run of algorithm

1) big-big: both endpts in big buckets — counted twice

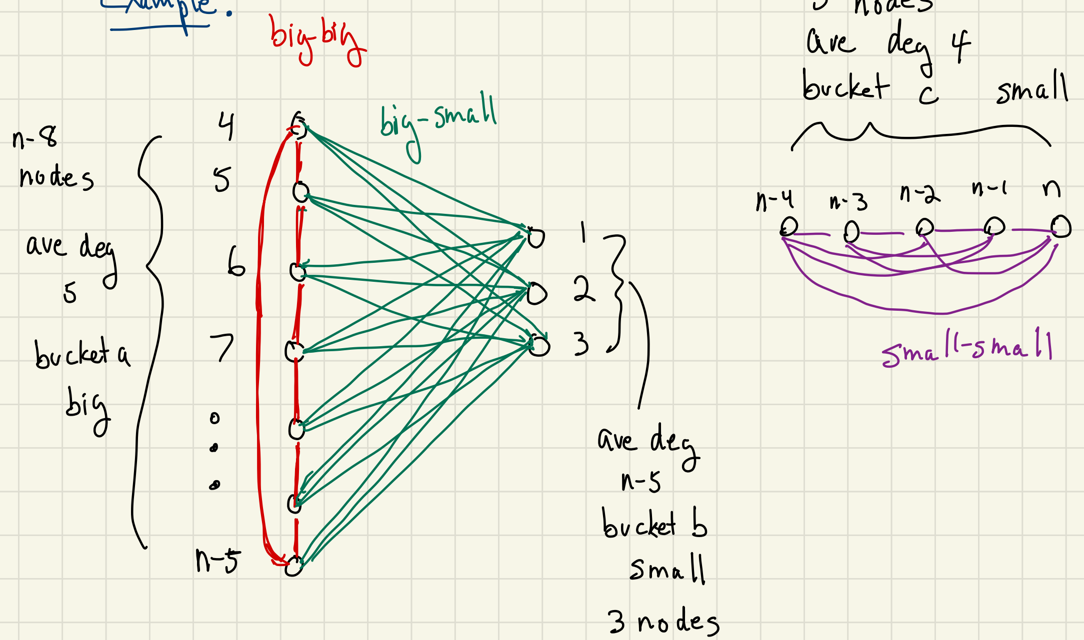
2) big-small: one endpt in big bucket
" " " small " — counted once

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:

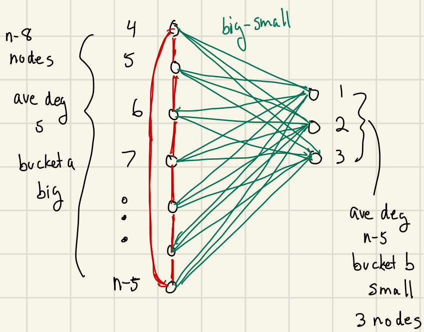


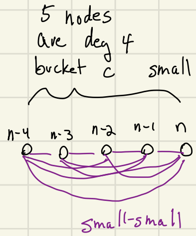
Total degree: $5\cdot(n-8) + (n-8)\cdot 3 + 4\cdot 5 = 8(n-8)+20$

ave degree ≈ 8

algorithm will likely output ≈ 5

Example:





New algorithm:

- Take sample S (how big?)
- $S_i \leftarrow S \cap B_i$
- estimate $|B_i|$:

  for all i

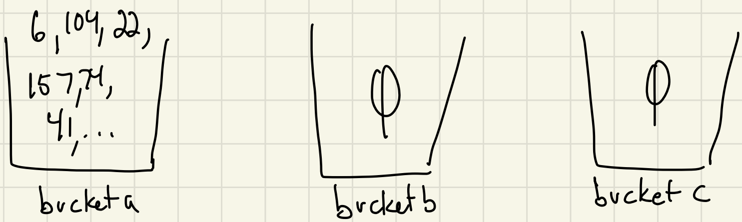
  if $|S_i| \geq \sqrt{\frac{\varepsilon}{n}} \cdot \frac{|S|}{c \cdot t}$ $\quad$ i "big"

  use $\rho_i \leftarrow \frac{|S_i|}{|S|}$

  else $\rho_i \leftarrow 0$ $\quad$ i "small"

- Output $\sum_i \rho_i (1+\beta)^{i-1}$

Samples:



most nodes here

whp $\rho_a \leftarrow 1$

few, if any, in these buckets

$\Rightarrow$ whp $b + c$ are small so likely that $\rho_b = \rho_c = 0$

output $\approx 1 \cdot 5$

Good news:

Small buckets can't have many nodes

$\Rightarrow$ bound on total # small-small edges

if $|B_i| > \frac{2\sqrt{\varepsilon n}}{ct}$ then expected size of S_i

is $\geq |S| \cdot \frac{|B_i|}{n}$

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

$$\leq \left(\underbrace{\frac{2\sqrt{\varepsilon n}}{ct}}_{\text{\# nodes per small bucket}} \cdot \underbrace{t}_{\text{\# buckets}} \right)^2 = O\left(\frac{\varepsilon n}{c^2}\right) = O(\varepsilon n)$$

if ignore small-small edges,

they affect approx of $\bar{d}$

by $\leq \frac{\varepsilon n}{n} = \varepsilon$ additive factor

$\leq (1+\varepsilon)$ multiplicative factor

assume $\bar{d} \geq 1$

First Claim:

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

large-small underestimated by factor ≤ 2

small-small error

Improving further:

need to improve on "big-small" edges

Can we estimate fraction of them

+ 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 $i \in [1..\deg(v)]$
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 (big):

repeat $O(1/\delta)$ times

pick random node $u \in B_i$

$e \leftarrow$ random nbr of u

set a_j to be $\begin{cases} 1 & \text{if } e \text{ "big-small"} \\ 0 & \text{o.w.} \end{cases}$

(e is "big-big")

Output α_i = average a_j

Analysis:

easy case: all nodes in B_i have same degree d

$T_i \leftarrow$ # big-small edges in B_i

$\Pr[\text{"big-small" edge } e \text{ in } B_i \text{ chosen}] = \underbrace{\frac{1}{|B_i|}}_{\text{prob } u \text{ is chosen}} \cdot \underbrace{\frac{1}{d}}_{\text{prob } (u,v) \text{ output given } u \text{ chosen}}$

$e = (u,v)$

only one of u,v big

wlog assume u big

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

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

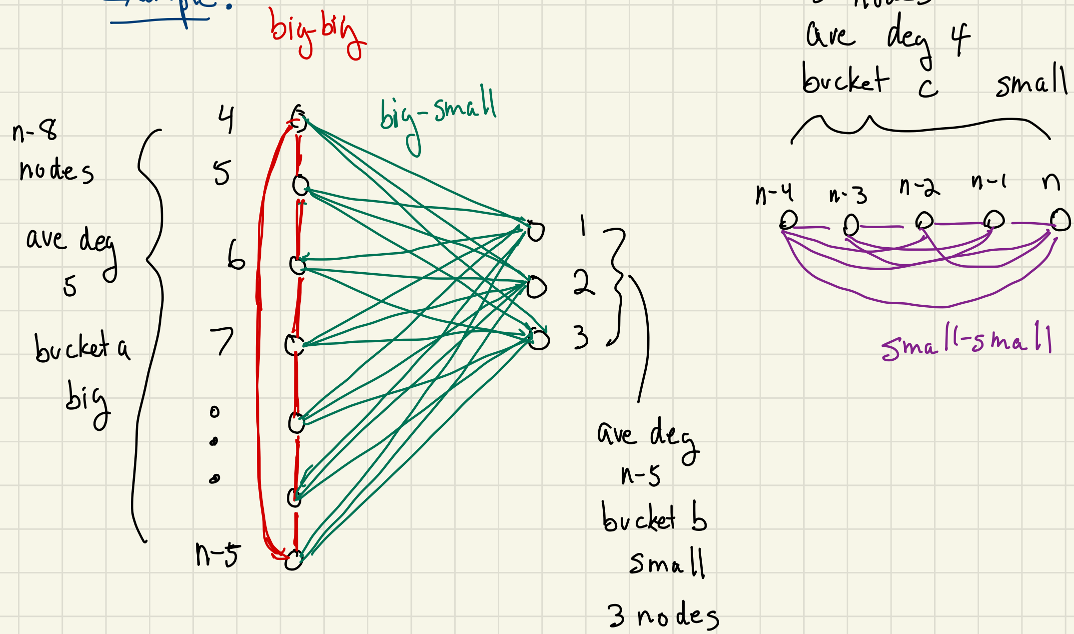
$$\frac{1}{|B_i|(1+\beta)^i} \leq \Pr[\text{"big-small" edge } e \text{ in } B_i \text{ chosen}] \leq \frac{1}{|B_i|(1+\beta)^{i-1}}$$

$$\left.\frac{T_i}{|B_i|(1+\beta)^i} \leq \underbrace{E[a_j]}_{\substack{\text{estimate to}\\ 1+\varepsilon\text{-mult factor}}} \leq \frac{T_i}{|B_i|(1+\beta)^{i-1}}\right\} \Longrightarrow$$

to get $(1+\varepsilon)(1+\beta)$ estimate of $\frac{T_i}{n}$ via $\alpha_i \rho_i \underbrace{(1+\beta)^{i-1}}_{\substack{\text{undercount}\\ \text{of \# edges in } B_i}}$

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

Example:



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: $\approx \dfrac{3(n-8)}{5(n-8)} = \dfrac{3}{5}$

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

Output $1\cdot\left(1+\frac{3}{5}\right)\underbrace{(1+\beta)^a}_{\approx 5} \approx 8$

Final Algorithm:

- sample $\Theta\left(\frac{\sqrt{n}}{\varepsilon}t\right)$ nodes + place in S
- $S_i \leftarrow S \cap B_i$
- For all i

  if $|S_i| \geq \sqrt{\frac{\varepsilon}{n}} \frac{|S|}{ct}$

  use $\rho_i \leftarrow \frac{|S_i|}{|S|}$

  for all $v \in S_i$

  - Pick random nbr u of v
  - $\chi(v) \leftarrow \begin{cases} 1 & \text{if } u \text{ is small} \\ 0 & \text{o.w.} \end{cases}$

  $\alpha_i \leftarrow \frac{|\{v \in S_i \mid \chi(v) = 1\}|}{|S_i|}$

  else use $\rho_i \leftarrow 0$

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

big-big + one side of big-small

Correction to get other side of big-small

Where do errors come from?

estimating p_i's
estimating α_i's $\Big\}$ mult $1 \pm \varepsilon$ - factor

small-small edges $\}$ additive εn error