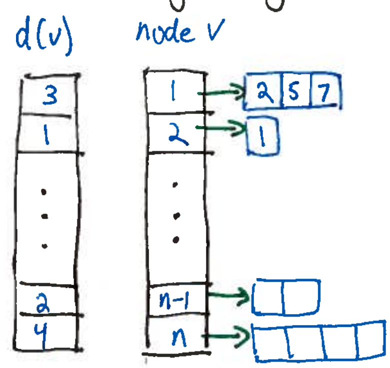
Approximating Average Degree

def Average degree $\bar{d} = \frac{\sum_{u \in V} d(u)}{n}$

Assume: G simple (no parallel edges, self-loops)

$\Omega(n)$ edges (not "ultra-sparse")

representation: adjacency list + degrees



- degree queries: on v return $d(v)$
- neighbor queries: for (v,j) return j^{th} nbr of v

Naive sampling:

Pick ?? sample nodes $v_1 \cdots v_s$

output $\frac{1}{s} \sum_i d(v_i)$ (ave degree of sample)

using straightforward Chernoff/Hoeffding $\Rightarrow \Omega(n)$ samples needed

Degree sequences are special?

$(n-1, 0, 0, 0, \dots, 0)$ not possible

$(n-1, 1, 1, 1, \dots, 1)$ is possible

Some lower bounds:

"ultrasparse case":

need linear time to get any multiplicative approx

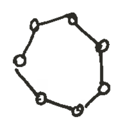
graph with 0 edges vs. graph with 1 edge

ave $\deg = 0$ ave $\deg = \frac{1}{n}$

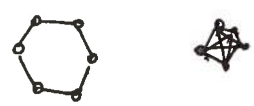
need $\Omega(n)$ queries to distinguish

ave deg ≥ 2:

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



$n - cn^{1/2}$ cycle + $cn^{1/2}$-clique $\bar{d} \approx 2 + c^2$



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

Algorithm idea:

group nodes of similar degrees

estimate average w/in each group

+ each group has bounded variance

- doesn't work for estimating ave of arbitrary numbers, why should it work here?

Bucketing:

set parameters $\beta = \varepsilon/c$

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

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

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

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:

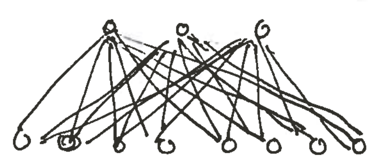
- Take sample S of nodes
- $S_i \leftarrow S \cap B_i$ (samples that fall in ith bucket use degree queries to determine this)
- estimate average degree contribution from B_i using S_i

  i.e. $\rho_i \leftarrow \frac{|S_i|}{|S|}$ $\leftarrow$ note: $\forall i$ $E[\rho_i] = E\left[\frac{|S_i|}{|S|}\right] = \frac{|B_i|}{n}$
- Output $\sum_i \rho_i (1+\beta)^{i-1}$

Problem:

$\underbrace{i \text{ s.t. } |S_i| \text{ is small}}_{\text{likely come from } i \text{ s.t. } |B_i| \text{ small}}$ $\leftarrow$ $\boxed{\text{for these, our estimate of } |S_i| \text{ could be terrible}}$

example of problem:



$\leftarrow$ 3 nodes, deg $n-3$

$\leftarrow$ $n-3$ nodes, deg 3

$a \leftarrow i$ s.t. $(1+\beta)^{i-1} \leq 3 \leq (1+\beta)^i$ $\quad |B_a| = n-3$ contributes $(n-3)\cdot 3$ edge

$b \leftarrow i$ s.t. $(1+\beta)^{i-1} \leq n-3 \leq (1+\beta)^i$ $\quad |B_b| = 3$ $\leftarrow$ contributes $3\cdot(n-3)$ edge

Never sampled but contributes ½ edges!!

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

Still, maybe good enough for 2-approximation?

ave deg 5
2017

Next idea: use "0" for small buckets

Algorithm:

- sample S ← how big?
- $S_i \leftarrow S \cap B_i$
- For all i

  if $|S_i| \geq \sqrt{\frac{\varepsilon}{n}} \cdot \frac{|S|}{c \cdot t}$ ← so $|S| > t\sqrt{\frac{n}{\varepsilon}}$

  let $|S| = \Theta(\sqrt{n}\ \text{polylog}\, n \times \text{poly}\, 1/\varepsilon)$

  $\Rightarrow |S_i| \geq \Omega(\text{poly}\log n \times \text{poly}\, 1/\varepsilon)$

  use $\rho_i \leftarrow \frac{|S_i|}{|S|}$ ⇐ call i "big"

  else $\rho_i \leftarrow 0$ ⇐ call i "small"

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

Analysis:

1) Output not too large

idealistic (but unrealistic) case ⇒ Suppose $\forall i$ $\rho_i = \frac{|B_i|}{n}$, then $\sum_i \rho_i (1+\beta)^{i-1} = \sum_i \frac{|B_i|}{n} \underbrace{(1+\beta)^{i-1}}_{\leq \text{deg of node in } B_i}$

$\leq \bar{d}$

realistic case Suppose $\forall i$ $\rho_i \leq \frac{|B_i|}{n} (1+\gamma)$ ← bound on sampling error when $|S_i|$ is big (note that trivial when $|S_i|$ not big since $\rho_i \leftarrow 0$)

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

✓

ave deg 6
2017

2) Can output be too small?

if $\forall i \quad \rho_i = \frac{|B_i|}{n}$ then $\sum_i \rho_i (1+\beta)^{i-1} = \sum_i \frac{|B_i|}{n} (1+\beta)^{i-1}$

$$\geq (1-\beta) \sum_i \frac{|B_i|}{n} \underbrace{(1+\beta)^i}_{\geq \text{ deg of node in } B_i}$$

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

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

For small i ????

How much undercounting?

divide edges into 3 types:

type is determined by run of algorithm

1) big-big - both endpts in big buckets — counted twice
2) big-small - one endpt in big bucket — counted once
   " " " small "
3) small-small - both endpts 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 <u>real</u> problem

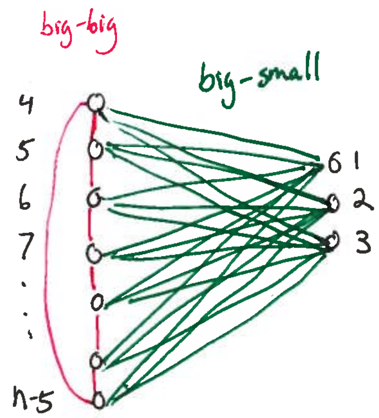
ave deg 7
2017

Example

ave deg 5

bucket a

big

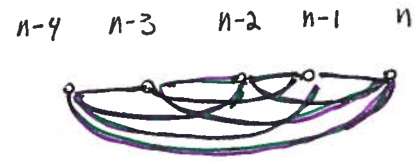


ave deg $n-5$

bucket b

small

Small-small



5-clique

ave deg 4

bucket c

small

Total degree

$$5(n-\cancel{8}) + (n-8)(3) + 4\cdot 5 = 8(n-\cancel{8})+20$$

ave deg $\approx 8n$

Samples

bucket a	bucket b	bucket c
6, 104, 22, 157, 74, 41, ...	$\emptyset$	$\emptyset$

most nodes here

$\Rightarrow$ (whp) bucket a is big, in fact,

whp $\rho_a \leftarrow 1$

output ≈ 5

very few nodes in these buckets so unlikely to see any samples

$\Rightarrow$ (whp) b & c are small

$\rho_b \leftarrow 0$ $\quad$ $\rho_c \leftarrow 0$

---

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

Fraction of big-big + big-small : $\dfrac{3(n-8)}{\cancel{5}(n-8)} = 3/\cancel{5}$

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

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

Good news: Small buckets can't have too many nodes

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

if $|B_i| > \frac{2\sqrt{\varepsilon n}}{ct}$ then Expected size of S_i is $\geq |S| \cdot \frac{|B_i|}{n}$

$O\left(\frac{\log n}{\varepsilon}\right)$

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

twice the threshold for being "big"

so very likely algorithm will decide that i is "big"

via Chernoff bnds

So assume $|B_i| \leq \frac{2\sqrt{\varepsilon n}}{ct}$ for all i "small"

then total # small-small edges

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

if we ignore them, they affect approx of

$\bar{d}$ by $\leq (1+\varepsilon)$ multiplicative factor $\leftarrow$ here we assume graph has ave degree ≥ 1

$\leq \varepsilon n$ additive factor

First Claim:

Algorithm almost gives factor 2 mult approx

since large-small underestimated by $\leq$ factor $1/2$

we get $(2+\varepsilon)$ - multiplicative approx

large-small error

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 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 $\begin{cases} 1 & \text{if } e \text{ is "big-small"} \\ 0 & \text{o.w. } (e \text{ is "big-big"}) \end{cases}$

Output α_i = average a_j

ave deg 10
2017

Analysis:

Easy case: All nodes in B_i have same degree

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

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

$e=(u,v)$ only one of u,v is big since e is "big-small"

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

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

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

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

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

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

ave deg 11
2017

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

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

  else use $\rho_i \leftarrow 0$

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

includes big-big & one side of big-small

other side of big-small correction