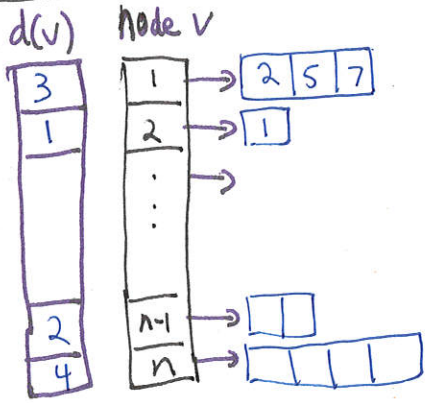
Approximating the Average degree

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

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

Representation of G:



Adjacency list + degrees:

- degree queries: on v return $d(v)$
- neighbor queries: for (v, j) return jth nbr of v

Naive sampling:

Pick ?? random nodes $v_1 \dots v_s$

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

straightforward use of Chernoff/Hoeffding needs $\Omega(\frac{1}{n})$ samples

Degree sequences are special?

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

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

Some lower bounds for approximation:

"Ultra sparse" case: need linear time to get any multiplicative approx:

graph with 0 edges ave deg = 0 vs. graph with 1 edge ave deg = 1/n

requires $\Omega(n)$ queries to distinguish

Ave deg ≥ 2 case :

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

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

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

Algorithm

idea: group nodes of similar degrees } doesn't work for estimating ave of arbitrary numbers, why should it work here?
estimate average w/in each group

buckets!

set $\beta = \varepsilon/c$

$t = O\left(\frac{\log n}{\varepsilon}\right)$ # buckets

$B_i = \{ v \mid (1+\beta)^{i-1} < d(v) \leq (1+\beta)^i \}$ for $i \in \{0 \ldots t-1\}$

Note that 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 $\sum_i (1+\beta)^{i-1} |B_i| \leq d_{\text{total}} \leq \sum_i (1+\beta)^i |B_i|$

ave deg(G)

First idea:

- Take sample S
  - $S_i \leftarrow S \cap B_i$ (samples that fell in i^{th} bucket)
  - estimate average degree of B_i using S_i
    i.e. $\rho_i \leftarrow \frac{|S_i|}{S}$

    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: i s.t. $|S_i|$ is small (" " $|B_i|$ " ") } for these, estimate of ρ_i will be "off"

Next idea: use "0" for small buckets

Algorithm:

- Sample S
- $S_i \leftarrow S \cap B_i$
- For all i

  if $S_i \geq \frac{\varepsilon}{\sqrt{n}} \cdot \frac{|s|}{c \cdot t}$ ← how big? we'll see in a minute...

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

  else $\rho_i \leftarrow 0$

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

← so $|S|$ will need to be atleast bigger than $t\sqrt{\frac{n}{\varepsilon}}$

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

Analysis

Output not too large:

ideal (unrealistic) case → suppose $\forall i \quad \rho_i = \frac{|B_i|}{n} \Rightarrow \sum_i \rho_i (1+\beta)^{i-1} = \sum_i \frac{|B_i|}{n} (1+\beta)^{i-1}$

$\leq \bar{d}$

realistic case → suppose $\forall i \quad \rho_i \leq \frac{|B_i|}{n} \cdot (1+\gamma)$ (sampling error)

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

for small i, $\rho_i \leq \frac{|B_i|}{n} \cdot (1+\gamma)$ by def

large i, $\rho_i \leq \frac{|B_i|}{n} \cdot (1+\gamma)$ whp by sampling bounds

So output not too large!

But are we undercounting by a lot?

For large i,

by sampling, $\rho_i \geq \frac{|B_i|}{n} \cdot (1-\gamma)$

so $$\sum_i \rho_i (1+\beta)^{i-1} \geq \sum_i \frac{|B_i|}{n} (1-\gamma)(1+\beta)^{i-1}$$

$$\geq (1-\beta)(1-\gamma) \sum_v \frac{d(v)}{n}$$

$$= (1-\beta)(1-\gamma)(\bar{d})$$

For small i ???

Undercounting on small buckets:

3 types of edges:

small/large determined by algorithm

1) large-large - both endpts in large buckets — counted twice
2) large-small - one endpt in large bucket, one in small — counted once
3) small-small - both endpts in small buckets — never counted

How many small-small edges?

good news: small buckets don't have many nodes

Assume for all small buckets, $|B_i| \leq \sqrt{\frac{\varepsilon}{n}} \cdot \frac{2n}{c\varepsilon} = \frac{2\sqrt{\varepsilon n}}{c\varepsilon}$

else would have had more samples!

expected # samples is $\sqrt{\frac{\varepsilon}{n}} \frac{2|S|}{c\varepsilon}$

total # small-small edges:

$$\leq \left(\frac{2\sqrt{\varepsilon n}}{c\varepsilon}\right)^2 = O\left(\frac{\varepsilon n}{c^2}\right) = O(\varepsilon n)$$

so if we ignore them, they affect approx of $\bar{d}$ by $\leq (1+\varepsilon)$ multiplicative factor, $\leq \varepsilon n$ additive.

when graph has degree $\Omega(n)$

First Claim:

Algorithm gives factor 2 mult approx since large-small underestimated by at most ½ factor.

$\Rightarrow$ $2+\varepsilon$- multiplicative approximation

↑ large/small ↑ small/small

Improving Further

Need to do better on "large-small"

idea: estimate fraction of "large-small" & correct for them

how?

Plan: standard sampling?

pick random edge

or pick "almost" random edge ???

New query:

random neighbor query (v):

given v, return random nbr of v

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

Pick almost random edge in a bucket:

pick random edge by picking any node that falls in that bucket + random nbr query from that node.

Algorithm to estimate fraction large-small in B_i:

repeat $O(1/\delta)$ times:
  pick random node $u \in B_i$
  $e \leftarrow$ random nbr of u
  set a_j to $\begin{cases} 1 & \text{if } e \text{ is "large-small"} \\ 0 & \text{o.w.} \end{cases}$

output α_i = average a_j

Analysis

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

let T_i = number of "large-small" edges in B_i

$\Pr$["large small" edge e in B_i chosen] $= \frac{1}{d\,|B_i|}$

e_i can only touch B_i from one endpt since B_i either "large" or "small" but not both!

$E[a_j] = \Pr$[any "large small" edge in B_i chosen]

$= \frac{T_i}{d\,|B_i|}$

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

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

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

algorithm estimates $E[a_j]$ to $(1+\varepsilon)$-multiplicative factor, giving $(1+\varepsilon)(1+\beta)$ estimate of $\frac{T_i}{n}$ via $\alpha_i \cdot \underbrace{\rho_i(1+\beta)^{i-1}}_{\text{undercount of \# edges in } B_i}$

running algorithm for each bucket gives:

Final Algorithm:

- Sample S s.t. $|S| \ge \sqrt{\frac{n}{\varepsilon}}\, t$
- $S_i \leftarrow S \cap B_i$
- For all i
  if $S_i \ge \sqrt{\frac{\varepsilon}{n}}\,\frac{|S|}{Ct}$, use $\rho_i \leftarrow \frac{|S_i|}{|S|}$
  For all $v \in S_i$
  - pick random nbr u of v
  - $\chi(v) \leftarrow \begin{cases} 1 & \text{if } u \text{ small} \\ 0 & \text{o.w.} \end{cases}$

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

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

(α_i: large-small correction; $(1+\beta)^{i-1}$: includes large-large + one side of large-small)