Lecture 16 - Random Graphs

Sunday, March 20, 2022 2:01 PM

What is a random graph?

- randomly configured graph in "some way"
- usually: $|V(G)|$, $|E(G)|$, D, p

degree dist.

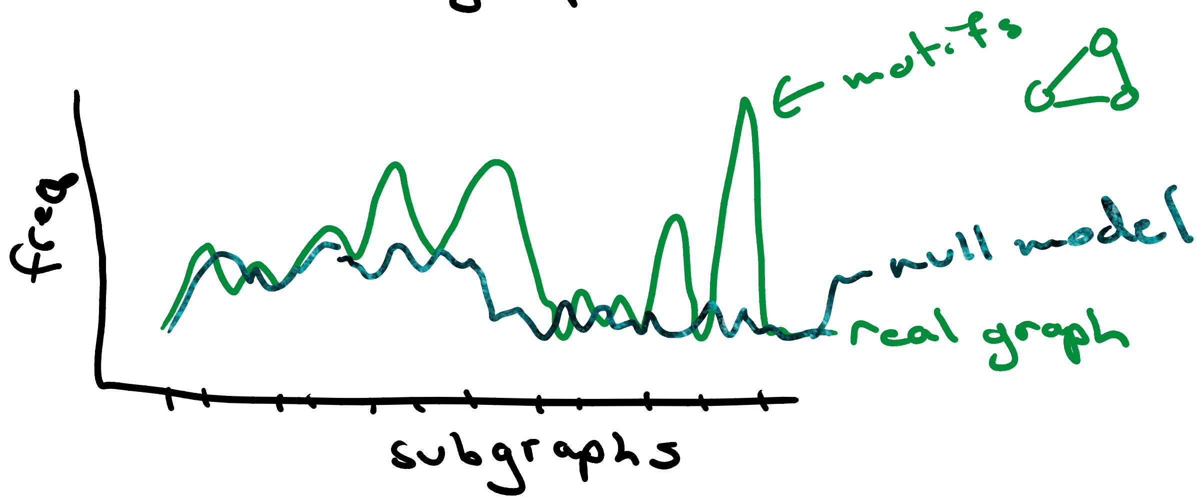
attachment probability

Why do we care?

- Mirror properties of existing real graphs for analytical/theoretical study
- Use random graphs as <u>null models</u>

E.g. motif finding

→ identifying "frequently-occurring" subgraph structures



Q: How do we define random graphs?

Q:

A: in various models

Classic model

$\Rightarrow$ Erdös-Rényi

O.G.: $G(n, m)$ $\qquad \langle k \rangle = \dfrac{2m}{n}$

n: # verts $\quad$ m: # edges $\quad$ $\langle k \rangle$: avg. degree

newer: $G(n, p)$

p: attachment probability

$\rightarrow$ prob. that u,v edge exists

$\hookrightarrow$ generation by evaluating all u,v pairs and flipping a coin

$\hookrightarrow$ Bernoulli process which outputs edge and degree distribution

$$P(k) = \binom{n-1}{k} p^k (1-p)^{(n-1)-k}$$

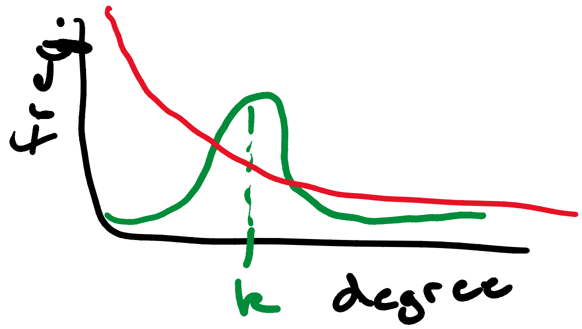
$P(k)$: prob. of degree k

mean value $\Rightarrow \langle k \rangle = \sum_{k=0}^{n-1} k p_k = p(n-1)$

As $n \to \infty$ and k is fixed

Binomial $\to$ Poisson

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^{k}}{k!}$$ ← mean value



Using our model to study real-world graph properties

- consider vertex v
- v has $\langle k \rangle$ neighbors, $|N(v)| = \langle k \rangle$
- each of v's neighbors has $\langle k \rangle$ neighbors as well

→ 2-hop neighborhood of v

$\Rightarrow \langle k \rangle^2$

In general:

$$|N_d(v)| = \langle k \rangle + \langle k \rangle^2 \ldots + \langle k \rangle^d$$

↑ d-hop neighborhood

$$\approx \frac{\langle k \rangle^{d+1} - 1}{}$$

$$\approx \frac{\langle k\rangle^{d+1} - 1}{\langle k\rangle - 1}$$

↑
d-hop neighborhood

consider as $|N_d(v)| \to n$

we can take

$$|N_d(v)| = n \approx \frac{\langle k\rangle^{d+1} \cancel{-1}}{\cancel{\langle k\rangle - 1}}$$

$$n \approx \langle k\rangle^{d}$$

$$d \approx \frac{\ln(n)}{\ln(\langle k\rangle)} \qquad \langle k\rangle << n$$

$$\underbrace{d \approx \ln(n)}$$

expected diameter grows logarithmically with $|V(G)|$

Issue: we don't model an explicit degree sequence

Introducing:

the configuration model

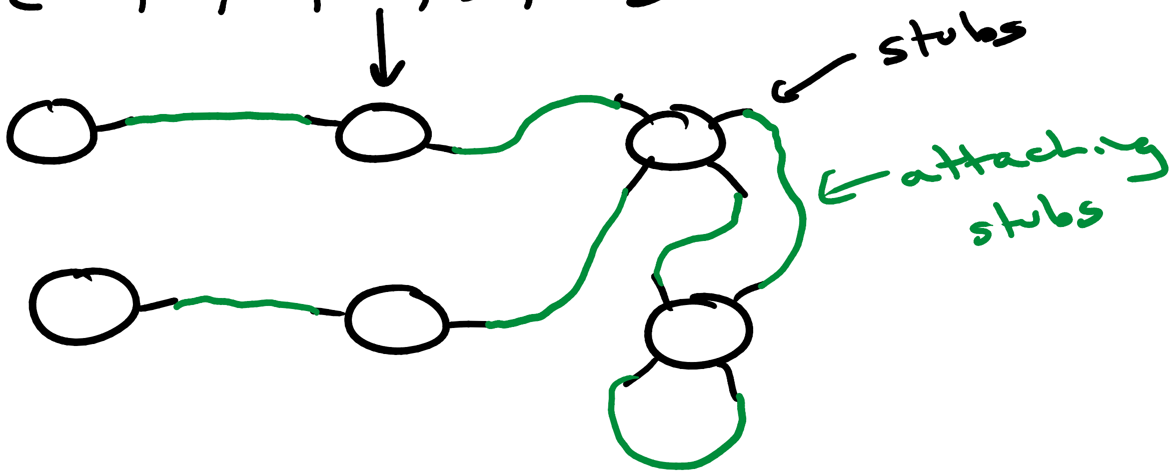
→ this will generate a random graph with ... sequence (assuming realizable)

→ this will generate a random graph with same degree sequence (assuming realizable)

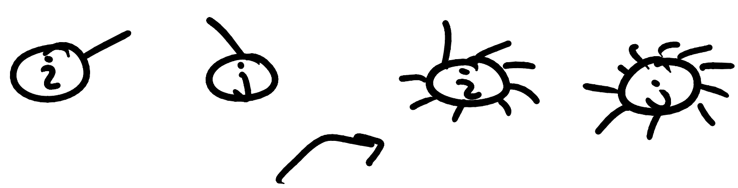
Basic idea:

- we have a bijection from our degree sequence to n vertices each with $d(i)$ number of <u>stubs</u>

$S = \{4, 4, 2, 2, 1, 1\}$



What about attachment probabilities?

consider 

Note: more likely to select stubs from higher degree vertex

Consider attachment of i, j's stubs

→ we know it'll probably be a function of $d(i), d(j)$

of $d(i), d(j)$

Probability of edge (i,j)

= (prob. of selecting i's stub)

$*$

(prob. of selecting j's stub)

$*$

2 ← we can select (i,j) or (j,i)

$*$

m ← we select m total edges

prob. of i's stub = $\dfrac{d(i)}{2m}$

$$p_{i,j} = \frac{d(i)}{2m} * \frac{d(j)}{\cancel{2m}} * \cancel{2 * m}$$

↑ attachment probability

$$p_{i,j} = \frac{d(i)\,d(j)}{2m}$$

Configuration model → attachment probs

attachment probs. → <u>another model</u>

→ Chung-Lu

$p_{i,j} = w_i w_j$ ← weights associated with each vertex

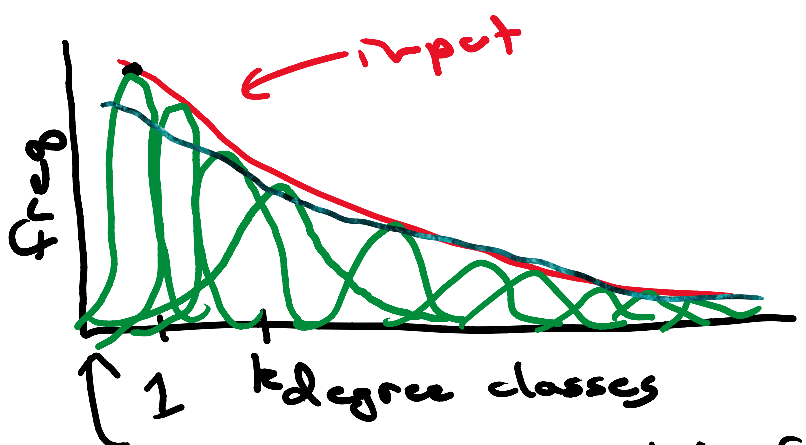
$$P_{i,j} = \frac{w_i w_j}{\sum_k w_k}$$ ← weights associated with each vertex aka degrees

We can generate a graph by evaluating this prob. for all u,v pairs

→ Note: we won't be hitting the degree distribution exactly (in expectation we are (not really))

Reason: we're actually layering a bunch of G(n,p) graphs

So: a vertex's degree is a sum of degrees for these E-R graphs
=> it's expected degree is a sum of Poisson's → Poisson



we end up with a lot of degree-0 verts

One more issue: $P_{i,j} = \frac{d(i)\,d(j)}{\quad}$

One more issue: $p_{ij} = \dfrac{d(i)\,d(j)}{2m}$

what happens when $d(i)\,d(j) > 2m$?

multi-graph $\rightarrow$ $p_{i,j}$ is just expected # of (i,j) edges

simple graph $\rightarrow$ nonsense

Null models

$\rightarrow$ a graph with some properties that is selected or implicitly defined from all possible graph topologies that fit the given properties

For loopy multigraphs: our configuration model and its probs. match a null model

For simple graphs: Chung-Lu graphs but the attachment probabilities are wrong and it's not an unbiased sample

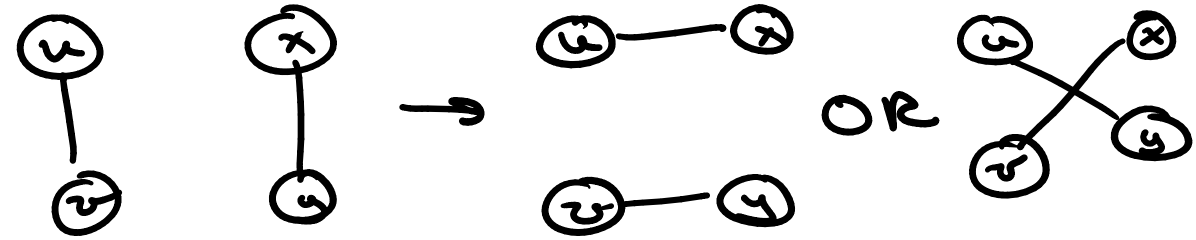
Q: how can we get an unbiased sample?

A1: not via attachment probability

A1: not via attachment probability

A2: via double-edge swaps

Double-edge swap:



Note: degrees are unmodified

An approach for simple null model generation:

Generate graph via H-H → perform "some number" of d.e.s. that don't create loops or multi-edges

Note: this is a Markov process

"Mixing time" = "some number"

= unknown in the general case

To get our attachment probabilities:

- We generate a large number of unbiased samples
- We just measure actual attachment

- we just measure actual attachment rates
  - → we can use these to calculate empirical attachment probs for simple graphs □