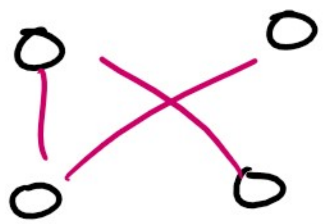
What is a random graph?



How big is $|V|$?
or $|E|$?
or p = attachment probability

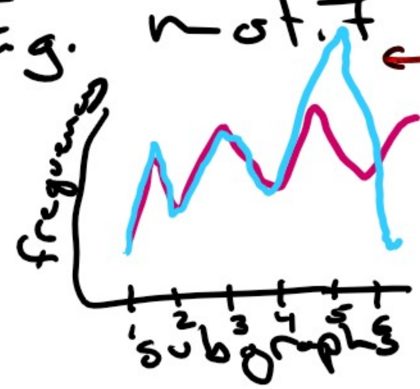
Big idea

- Randomly configured network
  - Edges determined randomly

Why do we care?

- Mirror properties of existing graphs to analytically
- Use as a "null model" for hypothesis testing

Eg. motif finding



motif
null model = random graph
graph we care about

How do we define a random

How do we define a random graph?

Classic model: Erdös-Rényi

O.G.: $G(n, m)$ ← # edges $\langle k \rangle = \frac{2m}{n}$

n: # verts ← avg. degree

newer: $G(n, p)$ $\langle k \rangle = p(n-1)$

p: attachment probability = probability that edge (i,j) exists

→ consider this as a Bernoulli process

- This gives us degrees as a binomial distribution

→ degree distribution

How many vertices of degree k

(degree k, # vertices w/ degree k)

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

→ probability of degree k

→ probability of degree k

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

Note: as $n \to \infty$ and k is fixed

Binomial distribution → Poisson distribution

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

why: approxmation is good

→ real networks have $n >> \langle k \rangle$ (much much bigger)

1 fewer parameter to care about

Let's consider connectivity of Erdös-Rényi graphs in terms of a GIANT component

- what $\langle k \rangle$ do we need?

$\langle k \rangle = 0$ → everything is disconnected

$\langle k \rangle = n-1$ → we have K_n clique

In reality, occurs around $\langle k \rangle = 1$

(see ref 1 section 3.C)

(see ref 1 section 3.C)

As $\langle k \rangle$ approaches and increases past 1, the giant component emerges → "critical point"

This mirrors reality → most real graphs have a giant component ✓

What about the other real-world properties such as:

- Small-world (low average shortest paths)
- Low diameter

Can we quantify these for an E-R graph? (Erdös-Rényi)

- Consider vertex v
- v has $\langle k \rangle$ neighbors, $|N(v)| = \langle k \rangle$
- Each of v's neighbors has on average $\sim \langle k \rangle$ neighbors
- 2-hop neighborhood of v is of size $\langle k \rangle \langle k \rangle = \langle k \rangle^2$
- 3-hop neighborhood → $\langle k \rangle^3$

2-hop neighborhood → $\langle k \rangle$

In general

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

d-hop neighborhood

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

To get an estimate of our graph diameter, consider how many hops it takes to visit ALL vertices from v

Set $|N_d(v)| = n$, solve for d

$$\frac{\langle k \rangle^{d+1} \cancel{-1}}{\cancel{\langle k \rangle - 1}} = n$$

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

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

as $n \gg \langle k \rangle$

SO IN OTHER WORDS:

Our diameter grows logarithmically as a function of our graph size

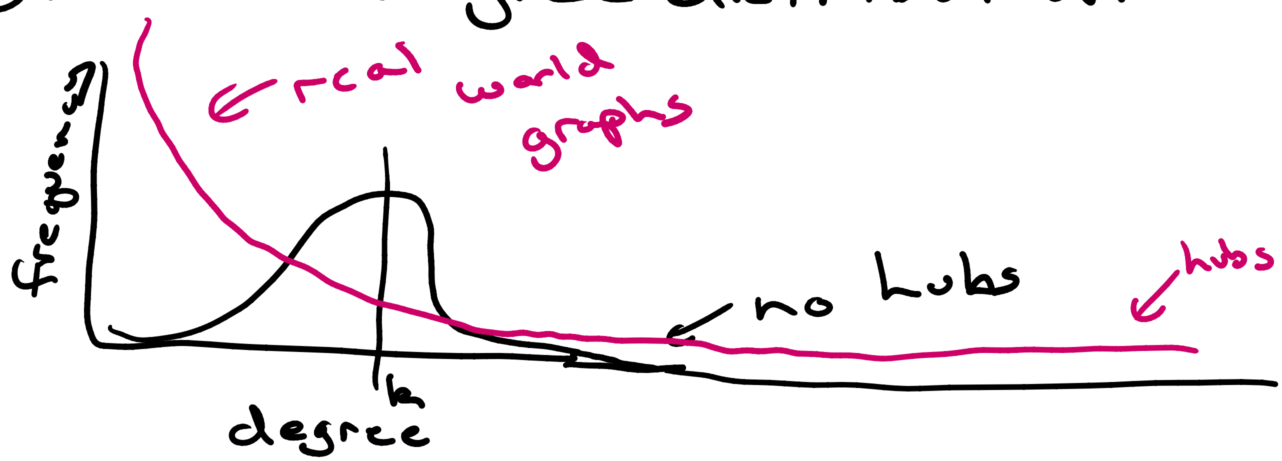
→ That's a small-world graph ✓

→ That's a small-world graph ✓

→ Similarly, our avg. shortest paths lengths grow log. as well ✓

Big issue: lack of hubs

Our E-R degree distribution



In reality $\langle k \rangle \ll k_{max}$

Introducing:

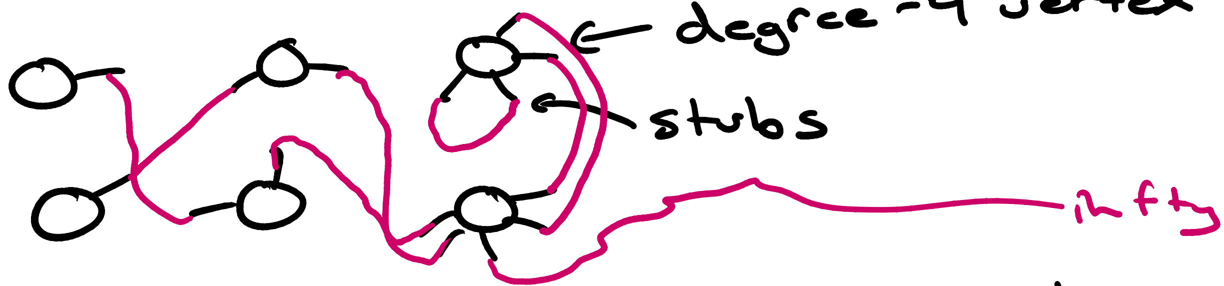
The configuration model

- We can generate a random graph with any arbitrary degree distribution (if its realizable)

Basic idea:

- We have n vertices with some number of stubs

- We have n vertices with a number of stubs
- # of stubs follows the degree distribution



- We select two stubs at random and connect them

- what are our attachment probabilitie?

Consider

Note: more likely to selects stubs from large degree vertices

Attachment probabity p_{ij} is a function of $d(i)$ and $d(j)$

probability of creating edge (i,j)

= (prob selecting stub from i)

*

(prob selecting stub from i)

$*$

(prob selecting stub from j)

$* 2 \leftarrow (i,j) = (j,i)$

$* m \leftarrow$ total attempts

$$\rightarrow p_i \, p_j$$

$$\approx \frac{d(i)}{2m} * \frac{d(i)}{\cancel{2m}} * \cancel{2*m}$$

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

Note: a lot of the properties we saw with E-R graphs still hold, but we also now can model an arbitrarily degree distribution

Big issue: tough to generate simple graphs

Also: still no clustering :(

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

And a Bernoulli process

For $v \in V(G)$

- create edges for all $u \in V(G)$ with probability $p = \frac{d(v)d(u)}{2m}$

$\Rightarrow$ Chung-Lu model

Note: we won't hit the degree distribution exactly

→ but in expectation, we will (not really though)

In reality: our output degree distribution is a sum of poisson distribution

→ we're really layering a bunch of Erdős-Rényi graphs

defined by p_{ij} for all possible

defined by p_{ij} for all possible degrees i, j

→ we end with up with for all target degrees in our degree distribution → Poisson

In effect: almost no input distributions can be generated

We also can't modify the input distribution to get output our target distribution

(Brissette and Slota, 2021)

Regardless → still no clustering

---

defining clustering

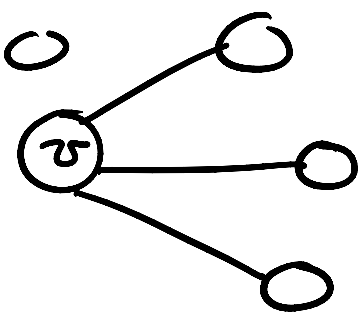
Generally → clustering coefficient

$$C_v = \frac{\text{triangles containg } v}{\text{total \# triangles that } v \text{ could be in}}$$

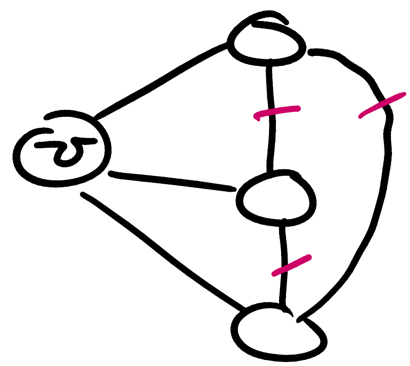
↗ clustering coefficient of v

$$C_v = \frac{2|e=(x,y)\in E(G) : x,y \in N(v)|}{d(v)(d(v)-1)}$$

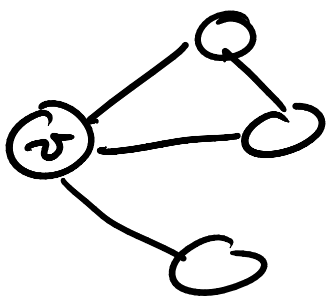
$C_v = 0$



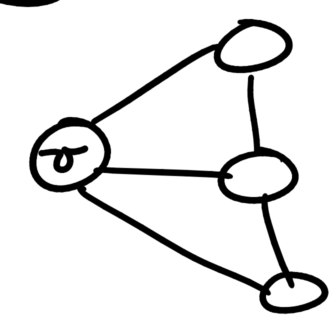
$C_v = 1$



$C_v = \frac{1}{3}$



$C_v = \frac{2}{3}$



Consider Erdős-Rényi

$$C_v = p\,\frac{d(v)(d(v)-1)}{\cancel{2}}$$

$$C_v = \frac{p \, \frac{d(v)(d(v)-1)}{\cancel{2}}}{\frac{d(v)\,d(v-1)}{\cancel{2}}}$$

$$C_v = p = \frac{\langle k \rangle}{n-1}$$

as $\langle k \rangle \ll (n-1)$

$C_v \rightarrow$ small

as $n \rightarrow \infty$, $C_v \rightarrow 0$

In reality: C_v is <u>not</u> zero

For social networks: $C_v \simeq 0.25$
$C_v \simeq 0.33$

Why? Think of how many of your friends are friends with each other

---

How can we generate a graph that exhibits clusering?

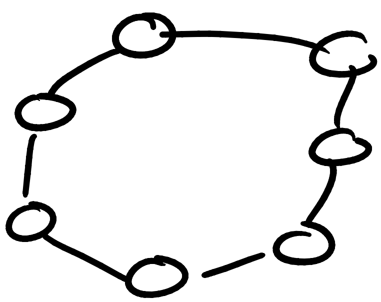
Introduction:

Introducing:

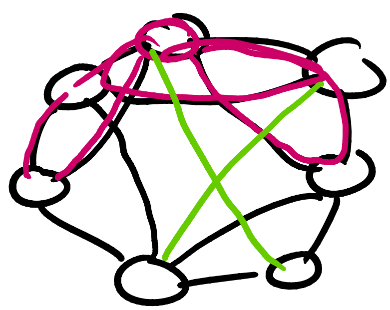
Watts-Strogatz model

→ small-world model

Basic model: ring graph with random rewiring



ring graph
$k = 2$



$k = 4$

→ On each side of a cycle, connect to $\frac{k}{2}$ neighbors

→ For each edge:

- Randomly select one new endpoint with probability β

Note: as $\beta \to 0$

Note: as $\beta \to 0$

we have a tightly clustered graph, but a high diameter

as $\beta \to 1$

we lose our clustering but greatly decrease our diameter

→ approach Erdös-Rényi

Note: initially $C_v \approx \frac{3}{4} \frac{(k-1)}{(k-2)}$

with $\beta = 0$

→ too high for real networks

$C_v(\beta) = C_v(0)(1-\beta)^3$

→ Each time we rewire, we lose a triangle

Unfortunately:

- No hubs ☹
- Don't match a degree dist. ☹

Everything so far → not exactly attempting to model the growth process of real networks

---

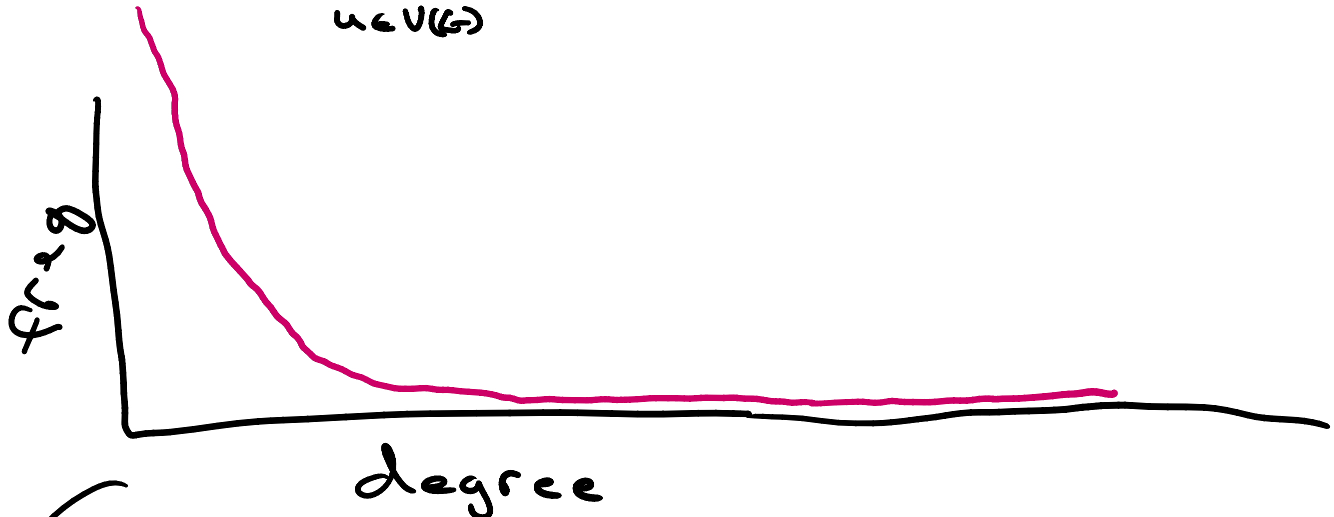
Another class of random graphs → model growth

Barabasi-Albert model
→ we add a new vertex, and attach it to existing vertices
→ Built on the phenomena of "preferential attachment"
- we bias towards high degree vertices
- rich-get-richer
→ creates hubs

- Our attachment probability for new vertex v to attach to existing vertex $\bar{i}$

to existing vertex i

$$p_i = \frac{d(i)}{\sum_{u \in V(G)} d(u)}$$



$\rightarrow$ power-law degree distribution

$$p_k \sim k^{-\gamma}$$

$\rightarrow$ very common in reality

$$2 < \gamma < 3$$

For Barabasi-Albert

$$\gamma \simeq 3$$

Power-law graphs are also referred to as "scale-free"

$\gamma \simeq 3$

Power-law graphs are also referred to as "scale-free"

→ as many real network do

- Degree dist ✓
- Small world ✓
- hubs ✓
- clustering ×
- growth ✓

→ Is there a model thats covers these real-world graph properties?

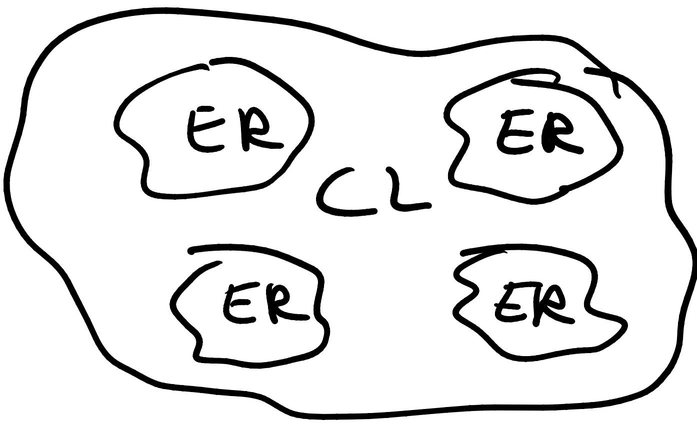
Yes

→ BTER: Block two-level Erdös Renyi

Erdös Renyi

Basic idea

- create cluster via dense Erdös-Renyi graphs
- Layer a Chung-Lu graph over all vertices



- we can model degree and clustering coefficient distributions

Problem: Tough to analytically study

- attachment probabilities are a function of degrees and clustering coefficients

---

What else for random graphs

→ RMAT and Krönecker

[illegible] via matrix products

- defined via matrix products

→ Benchmark graphs

- LFR for benchmarking community detection algorithms