Focus for today:

Null models (random graphs)

→ How can we best explicitly or implicitly model real networks to give us a baseline for observations/measurements

No measurements are done in a vacuum

Basis of science: hypothesis testing

Q: How likely is it that our measured observations did not emerge randomly?

did not emerge randomly?

Review

$G(n, m) \to$ n verts
m edges randomly selecting
2 vertices with replacement

Erdös-Renyi

$G(n, p) \to$ n verts
$\forall x, y \in V(G)$: (x, y) exists
with prob. p

Configuration Model $\to$ given degree dist.
create edges
w/ $p_{u,v} = \frac{d_u d_v}{2m}$
(effectively
$\hookrightarrow$ using stubs)

Chung-Lu $\to$ explicitly use $p_{u,v}$
probability for edge
creation

Issues:

E-R don't have a realistic

E-R don't have a realistic degree distribution

E-R and CM/CL graphs don't have clustering/triangles

G(n, m) and CM graphs don't define simple graphs

MOAR: Chung-Lu Model

generalization: $p_{u,v} = \dfrac{w_u w_v}{\sum_{i \in V} w_i}$

weights

Recall generation:

For $v \in V(G)$:

create edges for all $u \in V(G)$ with probability $p_{u,v}$ above

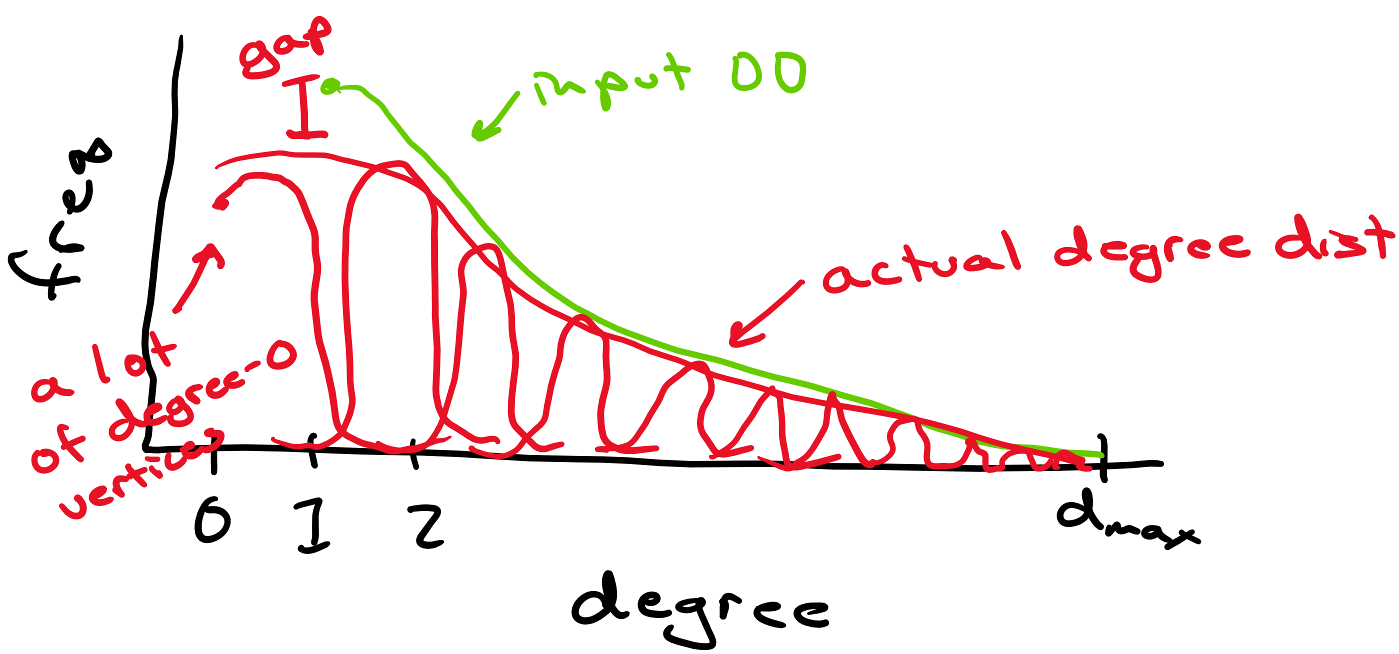
→ Note similarity to $G(n, p)$

Note 2: we won't match the degree dist. exactly

=> Theorists: but in expectation we will

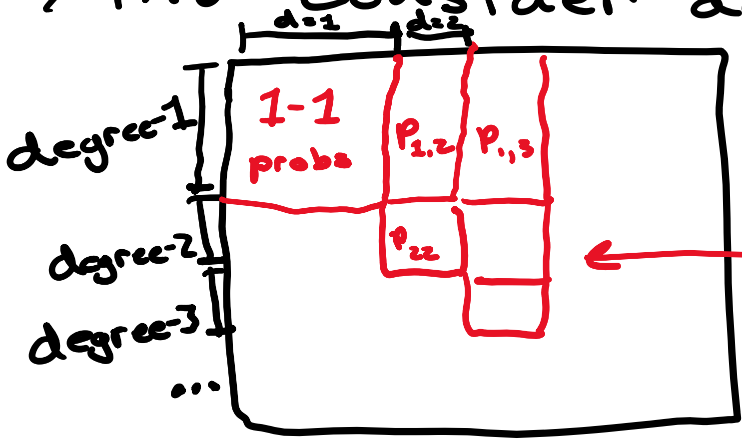
Unfortunately: we actually don't

Reality: our output distribution is a sum of Poisson dist. for all $d_i d_j$ degree pairs and vertices



And consider adjacency matrix

And consider adjacency matrix



→ sort verts by degree

each block is an E-R-style graph

In general: this is a block model

Takeaway: almost no actual degree distributions can actually be realized

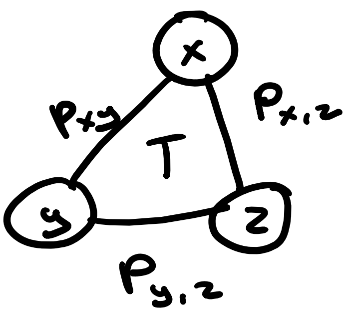
→ problematic to use as a null model in general

AKA theory ↔ practice doesn't always work out

Our one big issue

thus for: clustering

(or triangles)



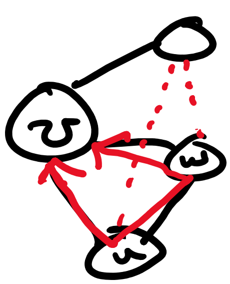
$$P_T = p_{xy}\, p_{xz}\, p_{yz} \approx p^3 \text{ for E-R}$$

$$\approx \frac{d_x^2\, d_y^2\, d_z^2}{(2m)^3} \text{ for CM/CL}$$

these $\to 0$ as $n \to \infty$

Clustering coefficient?

Recall: $C_v = \dfrac{|\text{triangles with } v|}{\text{total \# of triangles possible for } v}$



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

Consider an E-R graph

prob neighbors are connected

$$C_v = \frac{p\,\dfrac{d_v(d_v-1)}{2}}{\dfrac{d_v(d_v-1)}{2}}$$

same here

total unique

total unique combinations of neighbors

$$\underbrace{C_v = p}_{\text{very low}} = \frac{\langle k \rangle}{n-1}$$

Reality $C_v \approx 0.33$

(observed in reality)

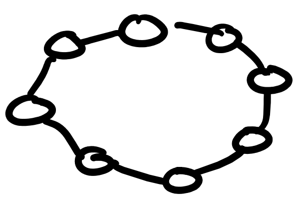
Note: generalize to CM/CL

Introducing:

Watts - Strogatz

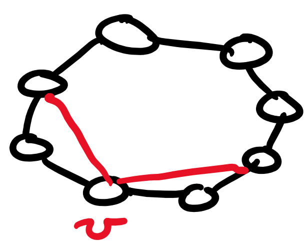
aka "small world" model

Basic idea: $WS(k, B, n)$

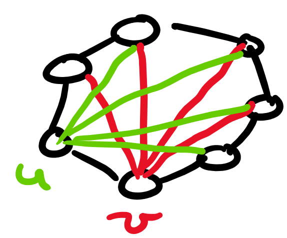


ring graph

$k=2$



$k=4$



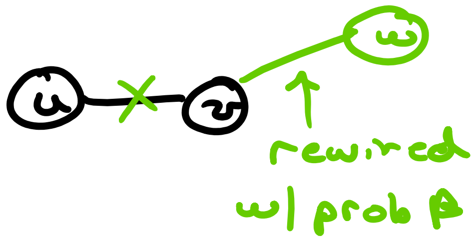
$k=6$

↳ k is initial degree, attach each v to $\frac{k}{2}$ neighbors

adjacent on the ring

2 neighbors
adjacent on the ring

WS: also rewires each edge with probability β to any of $n-1$ vertices



→ this gives us a low diameter and avg. shortest paths (small world)

$\beta \to 0$ we have the original ring (not small world but highly clustered)

$\beta \to 1$ we have an E-R variant (no clustering, low diameter)

$0 < \beta < 1$ we have a clustered graph w/a low diameter

Unfortunately:

w/o hubs (degree distributions)

No hubs (degree distributions)

No communities

One other approach

→ Modeling network growth

Barabasi-Albert:

→ we add a new vertex and attach it preferentially based on degrees of existing vertices

(AKA preferential attachment)

new vert, existing

$$P_{u,v} = \frac{d_v}{\sum_{i \in V} d_i}$$ ← evaluate for attachment

But no clustering ☹

Holme & Kim: BA + triangles

Do a B-A step (add v, preferentially attach it to w)

Then → we select another neighbor of w (if possible) to create a triangle

Can tune frequency of this step

→ result is scale-free, with clustering / some communities (large deg. verts) and a $\underbrace{\text{realistic DD}}_{\text{exact}}$

AND

Newman: Configuration + triangles

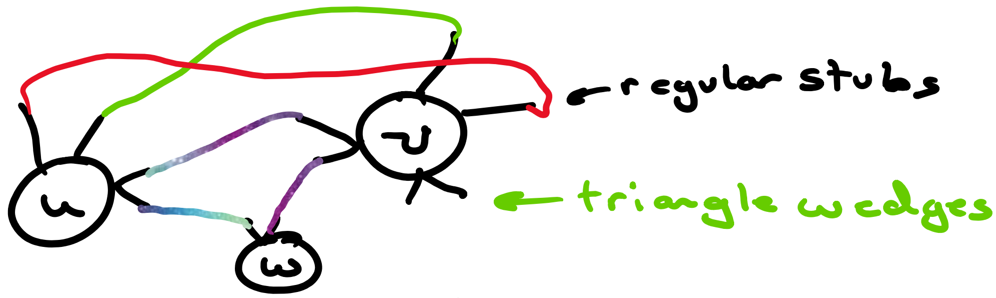
→ Input is distribution of (degree, # triangle) pairs

degree: must sum to even

triangle: must divide by 3

Example: $v \to$ (degree = 2, #T = 2)

Example: $v \to$ (degree = 2, #T = 2)



Approach: randomly select 2 stubs (edge creation)

randomly select 3 wedges

Actual degree: (#stubs) + 2(#T)

One issue still:

defining communities

Next class:

communities +

null model details