Last class:

- many random graphs discussed

Issue: not representative of real graphs in some measurable way

Q: Can we actually capture all the relevant properties?

degree distribution, triangles, clustering, communities

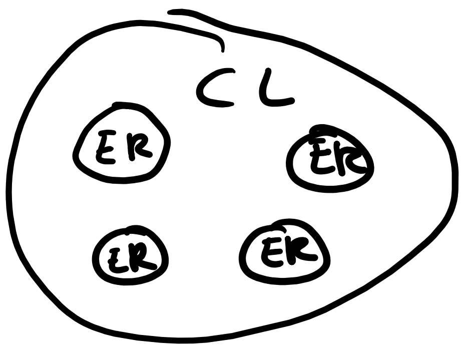
A: Yes?

Example: B-TER (block two-level Erdös-Renyí)

We construct a block model w/ communities of varying sizes (function of $\langle k \rangle$)

sizes (function of $\langle k \rangle$)

We layer a Chung-Lu model on top to connect communities



→ can tune probabilities to match C.C., D.D. and capture all real properties

Problem: tougher to analytically study

→ properties are a function of D.D., C.C. and graph construction

Takeaway:

complexity of model <=> complexity of analysis of model of analysis

$\iff$ difficulty to use as a null model

Other random graph models

- Defined by matrix products
  - → RMAT, Krönecker
  - → fast to generate
- Other block models

  $\begin{array}{|c|c|} \hline p_1 & p_2 \\ \hline p_3 & p_4 \\ \hline \end{array}$ (a superclass of eg., C-L and E-R)
- Benchmark graphs
  - → LFRs et al (C.D.)
  - → Matching, coloring, and

→ Matching, coloring, and
other basic graph problem

Null Models

aka null graph models

→ Graph that is <u>uniformly randomly</u>

configured given a set of

properties (p, n, m, D.D., CC)

Why: hypothesis testing

- We measure something
  on a real G
- We measure it on multiple
  random graphs
- Is the difference
  significant?

Null hypothesis

↪ Is what we're observing simply the result of random chance?

Q: What graphs can we use for this?

A: almost any we've discussed

(though they might not be "good")

↪ uniformly randomness

One specific example:

- graphs with a known D.D.

→ Chung-Lu probabilities

$$\frac{d_u d_v}{2m} = p_{uv}$$

Issues → don't fit the D.D.

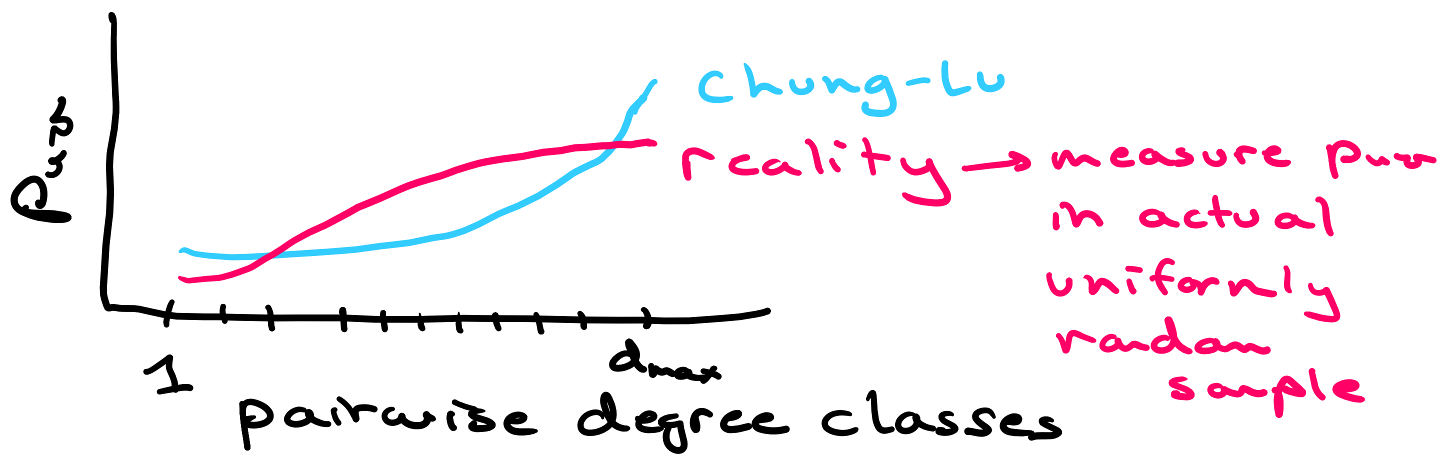
can use for simple graphs

✗ ↪ but biased probs (not uniformly random)

* → but biased probs
(not uniformly random)

Note: C-M fit the D.D. but
are not simple

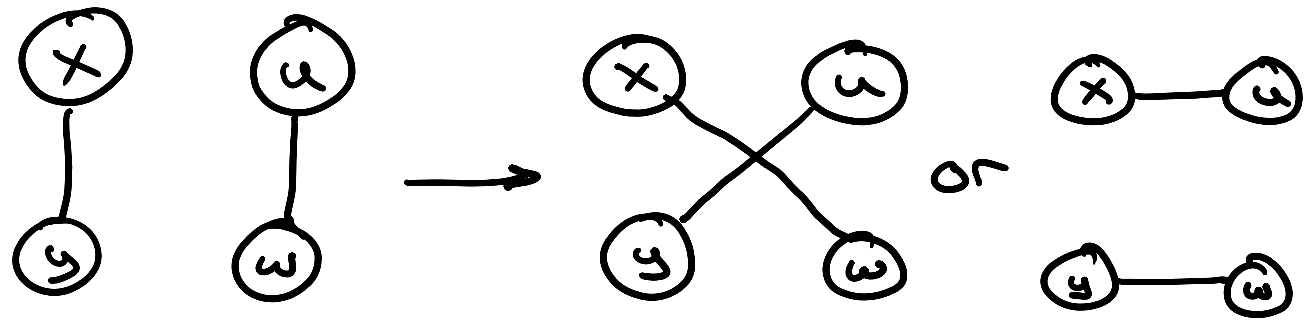
* In observation:



So: How can we actually generate a simple graph with a known D.D. in an unbiased way?

A: we can take any simple graph with the D.D. and do "some number" of

do "some number" of double-edge swaps



randomly select edges w/o bias

s.t. we don't make the graph non-simple

→ we get a Markov process that explores the entire graph topological space

- space is connected
- it is very very large

How many swaps to get sample?

- What are its properties when interpreted as a graph?

In practice, how many swaps?

→ Mixing time: unknown in general

In practice $\underline{O(n)}$?
good enough

Usage:

Motif/antimotifs

Algo benchmarking

Other measurement studies

Recall: Graph Mining is...

- Link prediction
- Centrality
- Clustering
- Vertex classification

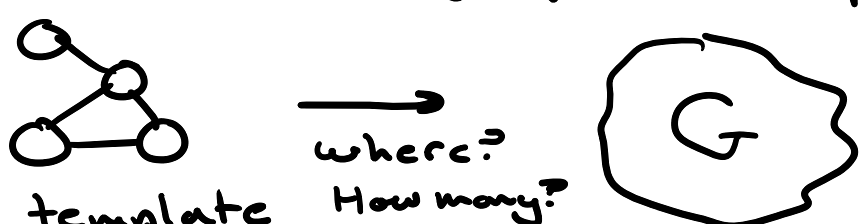
Now: $\underbrace{\text{subgraph mining}}$

you can tell it's relevant to graph mining

Many different problems fit under this umbrella

Triangle counting: Clustering, C.C., triadic closure, etc.

Template matching:
aka subgraph isomorphism



how many

aka subgraph counting
or subgraph enumeration

where

Motif/antimotif detection

Motif/antimotif detection

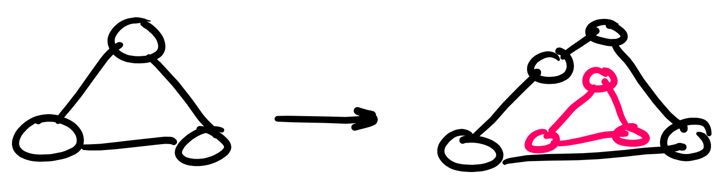
What subgraphs show up more/less frequently than expected?

Comparative analysis

Compare real networks based on subgraph counts, etc.

graph alignment

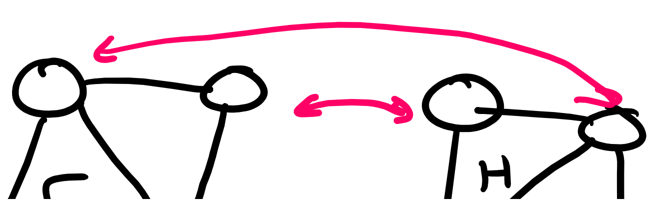
approximate subgraph isomorphism

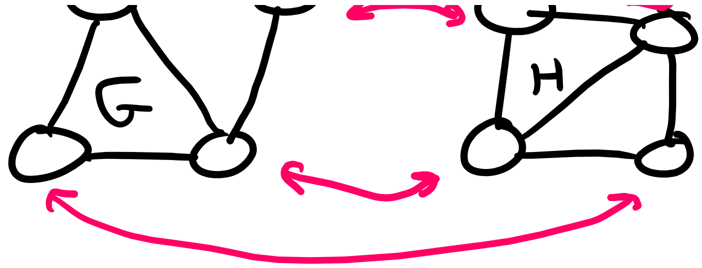


w/ error for edge additions/subtracts subdivisions, etc.

Subgraph Isomorphism

Graph isomorphism





Does a mapping of $V(G) \leftrightarrow V(H)$
s.t. edge relationships are retained?

→ Tougher than you think

(complexity: ~~unknown~~)
(NPI)

quasipolynomial $2^{O((\log n)^c)}$

Usually: algorithms turn G
and H into long structured
strings s.t. two graphs
have the same string
iff
they're isomorphic

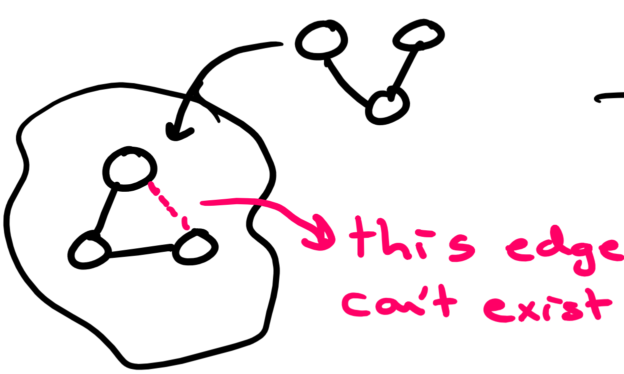
Useful in (bio)chemistry

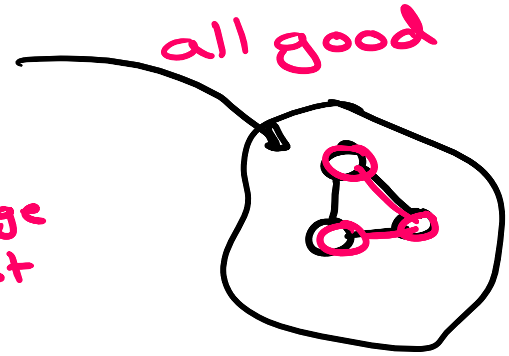
What we care more about:

subgraph isomorphism

Good	Bad
ton of applications	NP-complete
	$O(n^k)$, $k = \lvert V(T) \rvert$, $n = \lvert V(G) \rvert$
approximation algos work well	Generally need counts for multiple templates (blows up exponentially w/ template size)

Induced vs. non-induced





Next class:

triangle counting and algorithms

template matching algorithms

Hopefully: graph alignment