Lecture 14 - Random Graphs

Thursday, March 17, 2022 7:55 AM

recall:

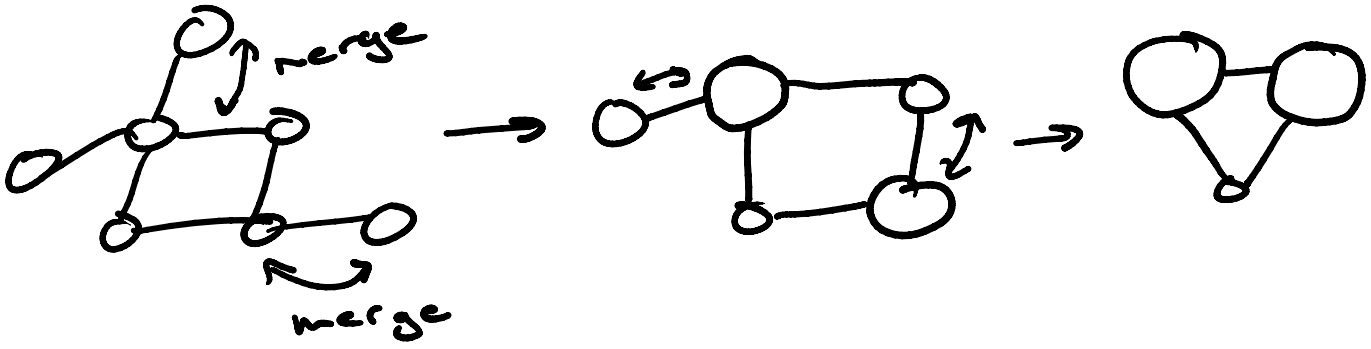
- graph Laplacian => $L = D - A$
- $Lv = \lambda v$ ← eigenvalues
  ↑ all eigenvectors
  ↳ we can use these to cut our graph into clusters (equal sized or not)

Coarsening

Generally: graph partitioning is hard (NP-hard)

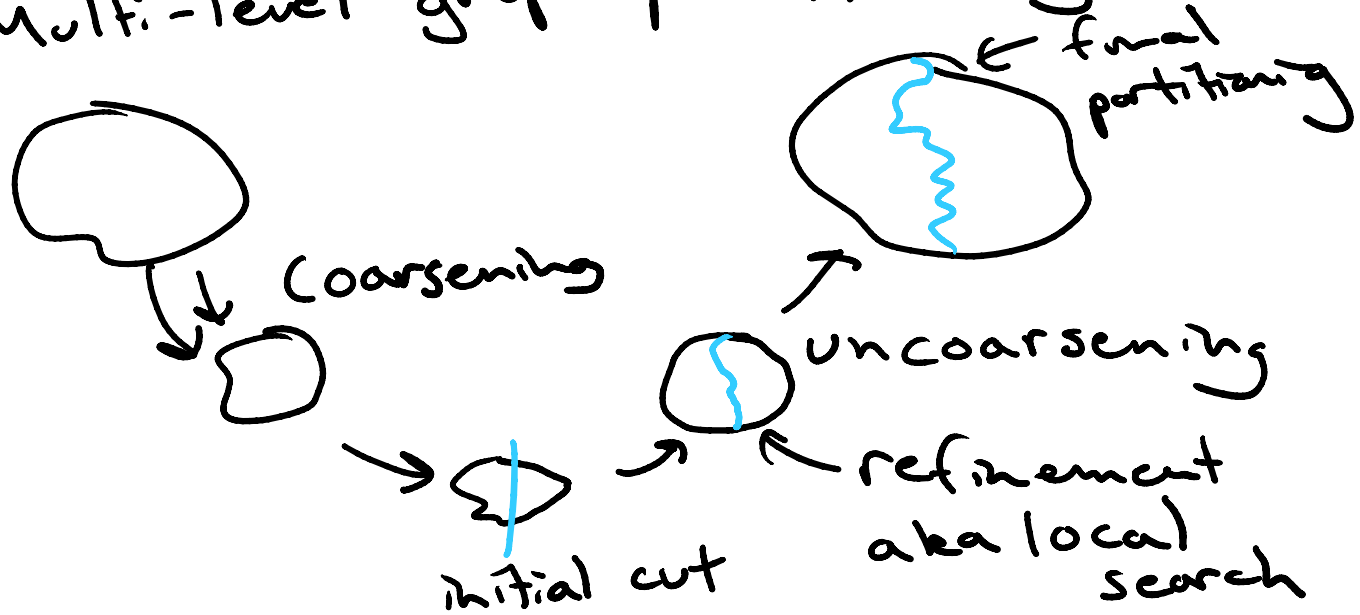
What we want to do: transform our input to a smaller/easier problem

How? Coarsening



merge

Multi-level graph partitioning



Coarsening: merging vertices via edge contractions to construct a smaller representation of the input

→ many application-specific ways to do this

→ for partitioning, we can preserve spectral properties

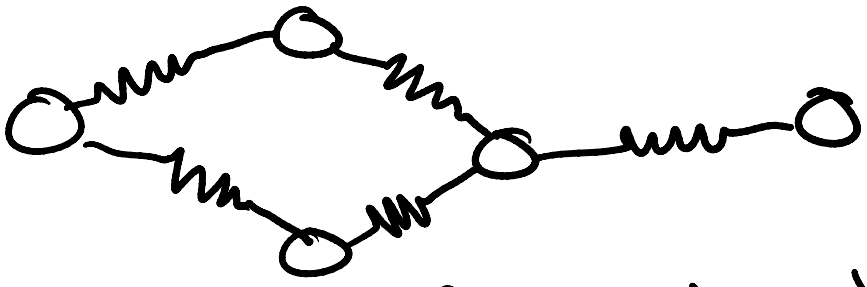
* How? selecting vertices to merge that minimizes change in the graph spectrum (vertices with similar eigenvector values)

Another application: visualization

Another application: visualization

Visualization: plotting a graph in 2D (or 3D) space

Common approach: model edges as springs



too close: force push vertices apart

too far: force pulls vertices together

Another approach: using the spectrum

recall: spectrum is essentially mapping from discrete to continuous space

Using two eigenvector values, we can map a graph into 2D space by interpreting the eigenvector values as coordinates

One major issue: large inputs

↳ hairball

Mitigating: can use coarsening
e.g. can coasen communities
to expose underlying structure

We can use spectral coasening and visualization

Coasen:
Find n pairs of vertices with closest spectral values and coasen them

Visualization:
Interpret the two largest eigenvector values for each vertex as (x,y) coordinates in 2D space

---

Random graphs

What is a random graph?
How many |V| ?
How many |E|?
How are edges created?
(attachment probability)

(attachment probability)

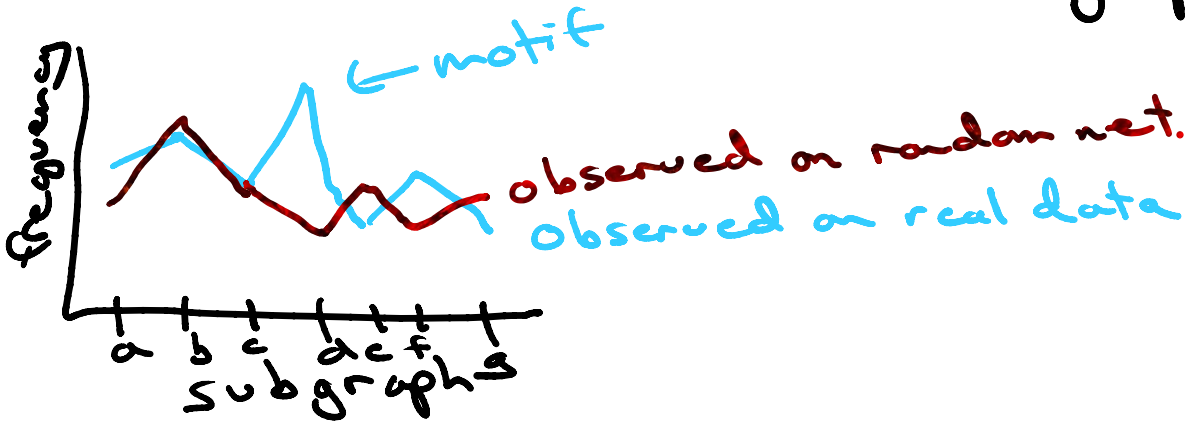
Big idea:
- Randomly configured network
- Edges determined randomly

Why do we care:
- Mirror properties of real networks to analytically study
- Use as a "null model" for hypothesis testing or otherwise

E.g. in modularity we compare our observation relative to a randomly configured network with the same degree sequence $\left(\frac{d_i d_j}{2m}\right)$

Eg. motif finding

motif = a frequently occuring subgraph



subgraphs

How do we define a random graph?

Classic model: Erdös-Rényi

O.G.: $G(n, m)$ (n = $|V|$, m = $|E|$) $\langle k \rangle = \frac{2m}{n}$ ($\langle k \rangle$ = avg. degree)

Issue: generates loopy multigraphs

Newer model: $G(n, p)$ $\langle k \rangle = p(n-1)$ (p = attachment probability)

→ This is a Bernoulli process

Hence, this gives us a binomial distribution for the degree distribution

→ How many vertices of degree k

(degree k, # vertices w/ degree k)

Degree distribution

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

($P(k)$: prob that vertex [illegible] degree k)

prob that vertex has degree k

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

Binomial → poisson

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

Why: approximation is good
$n >> \langle k \rangle$ in real networks

1 fewer parameter

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

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

$\langle k \rangle = 0$ → completely disconnected
$\langle k \rangle = n-1$ → completely connected

In reality, around $\langle k \rangle = 1$ we observe emergence of giant component

(see B. section 3.C)

↳ "critical point"

→ ... networks ✓

→ critical p...

This mirrors real-world networks ✓

What about other real-world properties

- small-world
- low diameter
- Hubbedness
- degree skew

Can we quantify these for an E-R graph?

- consider vertex v
- v has degree of $\langle k \rangle$
- each of v's neighbors has $\langle k \rangle$ neighbors
- 2-hop neighborhood around v has $\langle k \rangle \langle k \rangle = \langle k \rangle^2$
- 3-hop neighborhood $= \langle k \rangle^3$

$$|N_d(v)| = \langle k \rangle + \langle k \rangle^2 + \langle k \rangle^3 \dots \langle k \rangle^d$$

$|N_d(v)|$: d-hop neighborhood

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

To get an estimate for diameter, we can find how many hops in which we would expect to have all vertices

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

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

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

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

as $n >> \langle k \rangle \rightarrow d \simeq \ln(n)$

SO IN OTHER WORDS

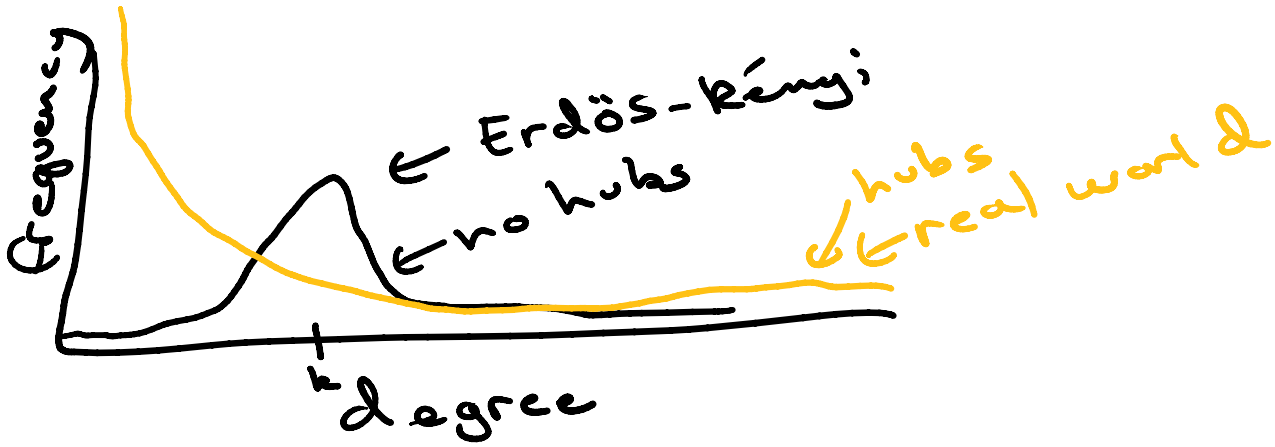
our diameter grows logarithmically as a function of graph order

↳ that's a small world graph ✓

(and avg. shortest paths grow $\ln(N)$ as well)

Big Issue w/ E-R

is the degree distribution



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

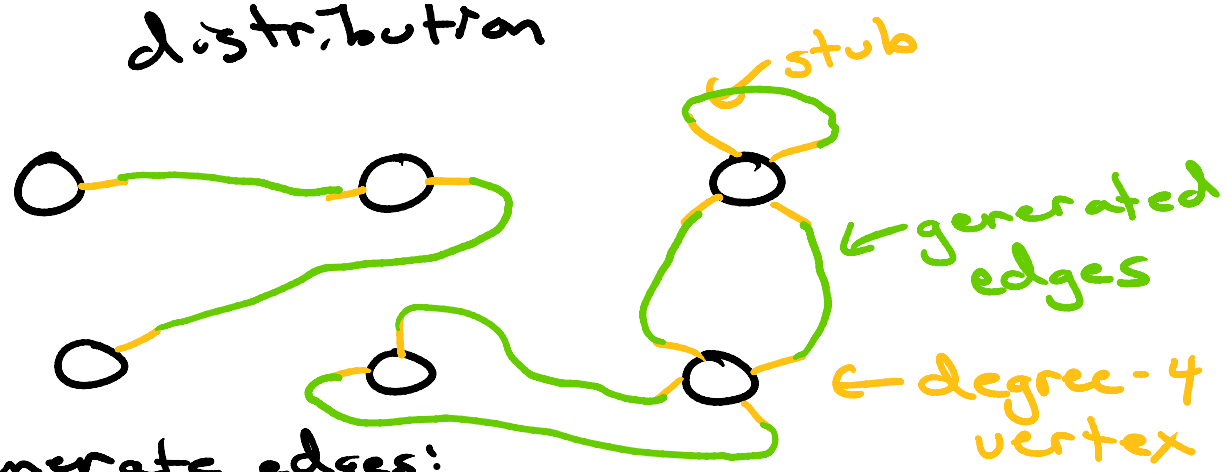
Introducing: the configuration model

- we can generate a random network with an arbitrary degree distribution

Basic idea

- We have n vertices each with some number of "stubs"
- # of stubs follows the degree distribution

distribution



to generate edges:
- We select two stubs randomly and add an edge between them

What are the attachment probabilities?

consider   or  

Note: more likely to select high degree vertex's stubs

Attachment probability is a function of $d(i)$ and $d(j)$ and the sum of the degrees

prob. of creating edge(i,j)

= (prob of selecting i's stub)
* (prob of selecting j's stub)

$*$ (prob of selecting j's stub)

$*$ 2 $(i,j), (j,i)$

$*$ m (total attempts)

$\rightarrow p_i p_j \, 2m$

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

$= \frac{d(i)\, d(j)}{2m}$ ← as we've seen with modularity

Note: a lot of the same properties hold as with E-R graph, but we also can model skewed distributions and hubs ✓

Issue 1: tough to generate simple graphs

Issue 2: no clustering