Lecture 3 - Network Measures

Thursday, January 20, 2022 7:50 AM

Plan for today:
- HW1 released 'soon' ✓
- Keep thinking about project ✓
- Review last class ✓
- k-edge-connectivity ✓
- Strong connectivity ✓
- Mining connectivity structure ✓
- Degree distributions ✓
- Power-law distributions ✓
- Shortest paths and small worldness ✗
- Code mode

Review last class

k-connectivity
- related to network resilience

→ how many vertices to remove to disconnect a network?

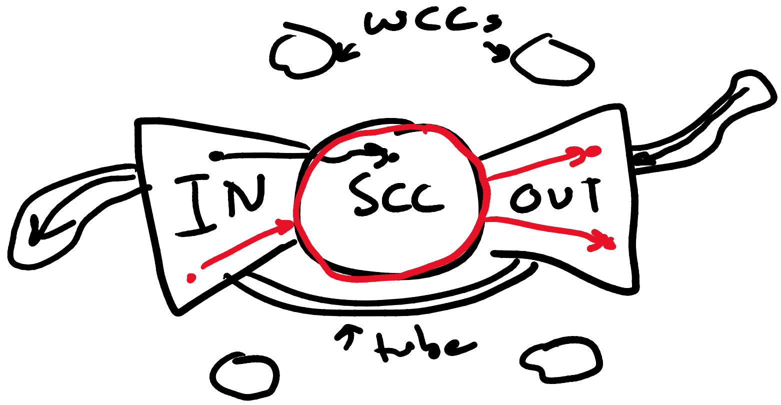
directed graphs
- assume directivity

~~v,u-path~~

strong/weak connectivity
- strong relates to notion of connectivity $\underline{\forall} v,u \in \underline{V(G)}: \underline{\exists}\, u,v\text{-path}$
- weak: as above but ignoring

- weak: as above but ignoring edge direction

→ connectivity in general gives insight into underlying structure of reachability

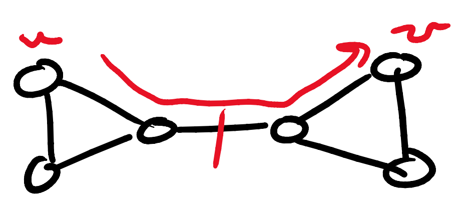


graph mining connectivity structure

---

k-edge-connectivity

- similar to k-connectivity BUT how many edges to cut to disconnect a network



Algorithmic approaches

- Network flow max flow = min cut $O(n^4)$

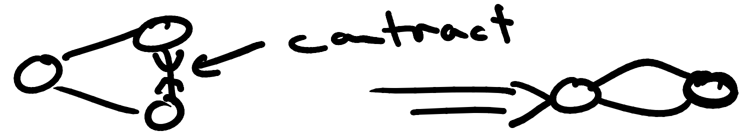
- Network Flow

cut $O(n^4)$

- Faster algorithms

→ randomized algos ~ $O(n^2)$

edge contraction



auxiliary graph construction

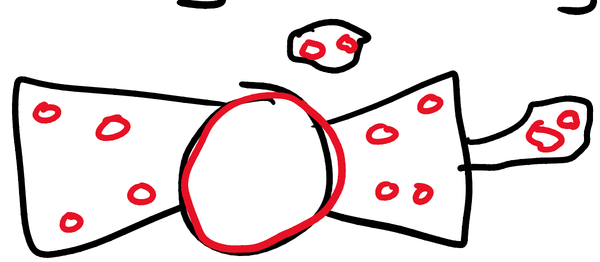
transformation

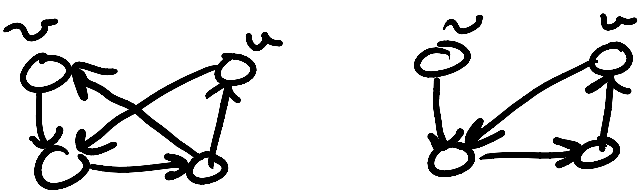
---

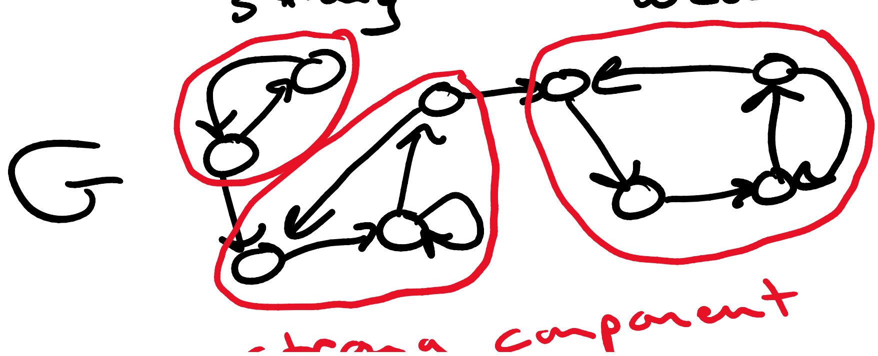
strong connectivity

Recall: strong connected iff $\forall u,v \in V(G)$: $\exists$ u,v-path

Usually: strong connectivity decomposition







maximal components

→ can't be made larger

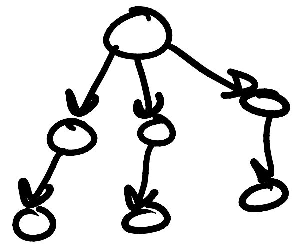
strong component decomposition

made larger

maximum → largest of all possibilities

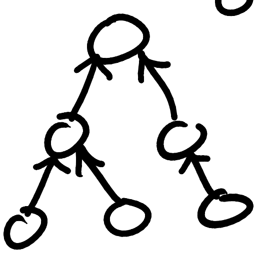
Algorithms:

- Our boi Tarjan (DFS)
- FWBW algo. (BFS)



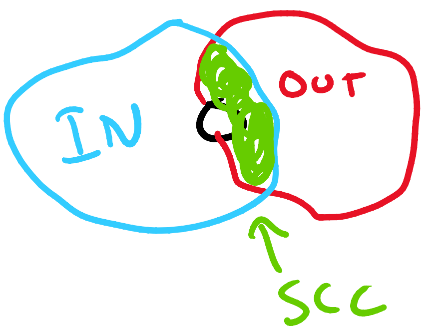
↳ select a start vertex

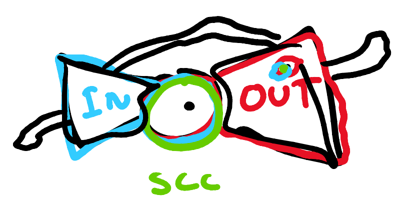
OUT = perform a BFS following out edges, note what vertices we reach



IN = perform a BFS following in edges, note what vertices we reach

SCC containing start vertex = OUT $\wedge$ IN





Mining connectivity structure

How???!!?!?!

How? ??!!?!?!

- Identifying cut edges (sets)
  cut vertices

- k-(edge-)connected components
  decompositions k-(CC)

- distributions of conn. comp. sizes
  * sizes vs. counts of CCs

  size 1 count 80
  size 2 count 60
  size 3 count 30
  ...
  size 40 count 1

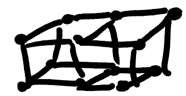
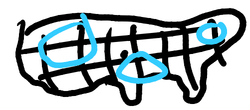
→ reachability between vertices and/or k-CCs

Degree distributions

→ how many vertices of each degree

we consider degree distributions in graphs in several ways:

graphs in several ways.

- Quantifying skew or irregularity
  (gini coefficient, power-law exponent fit)
  - * meshes → not skewed 
  - * social net → skewed 

- Random graph generation
  - * null model hypothesis testing

Power-law degree distributions

$$\underbrace{P(k) \sim k^{-\gamma}}$$ ← power-law exponent

probability of degree k

→ frequency of degree decreases exponentially as degree increases

If we want to compare skewedness between networks

- Fit power-law exponent
  - * many ways to do this
- We'll consider maximum likelihood estimator (MLE)

likelihood estimator (MLE)

* assume input is integers

$$\cancel{\gamma = \frac{1}{n} \sum_{d(v)\in G} \frac{d(v)}{d_{min}}}$$ (will double check)

γ = power law exponent

$n = |V(G)|$ = # verts in G

$d(v)$ = degree of v

d_{min} = min degree in G
=> 1

- Most real-world graphs

$$\underbrace{1 \le \gamma \le 3}$$

less skewed — more skewed

=> Generally, most graphs exhibit skew not just in degree distributions
- connectivity distributions
- cluster sizes

- cluster sizes
- shortest path lengths

$\Longrightarrow$ $\underbrace{\text{scale-free}}$ property