Introduction to Computational Graph Analytics

Lecture 1

CSCI 4974/6971

29 August 2016

Graph, networks, and characteristics of real-world data

Slides from Marta Arias & R. Ferrer-i-Cancho, Intro to Complex and Social Networks

So, let's start! Today, we'll see:

1. Examples of real networks
2. What do real networks look like?
   - real networks exhibit small diameter
     - .. and so does the Erdös-Rényi or random model
   - real networks have high clustering coefficient
     - .. and so does the Watts-Strogatz model
   - real networks' degree distribution follows a power-law
     - .. and so does the Barabasi-Albert or preferential attachment model

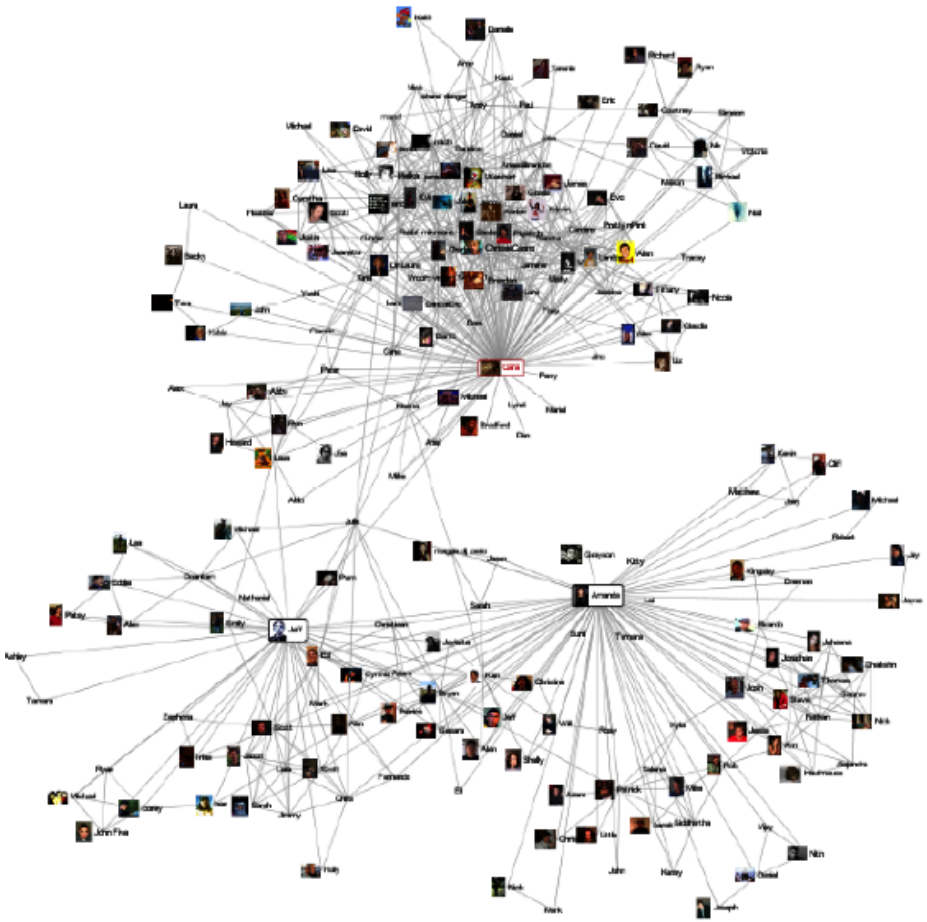
Examples of real networks

- Social networks
- Information networks
- Technological networks
- Biological networks

Social networks

Links denote social "interactions"

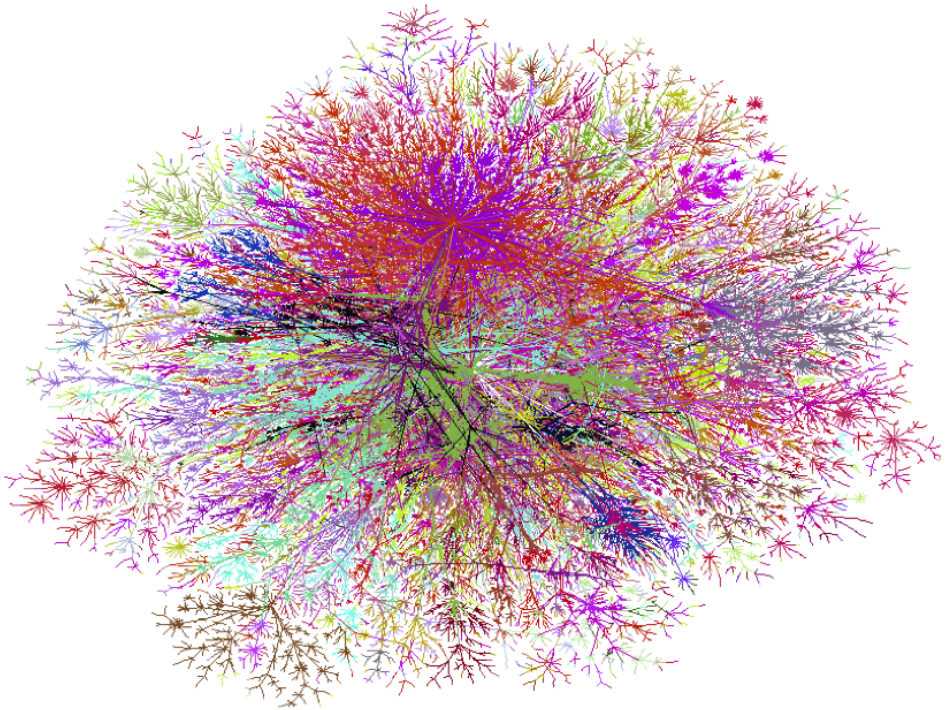
- friendship, collaborations, e-mail, etc.



Information networks

Nodes store information, links associate information

- citation networks, the web, p2p networks, etc.



Technological networks

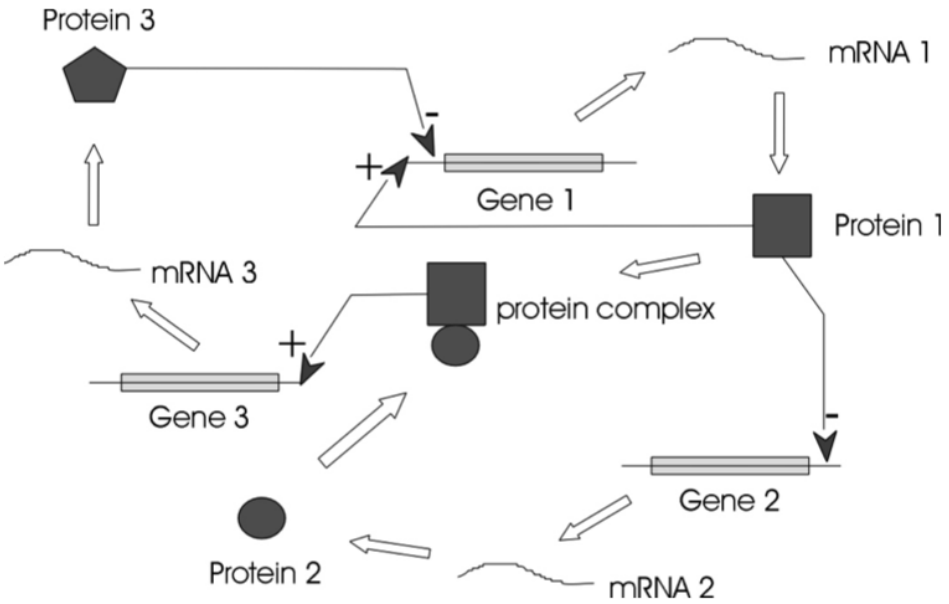
Man-built for the distribution of a commodity

- telephone networks, power grids, transportation networks, etc.

Biological networks

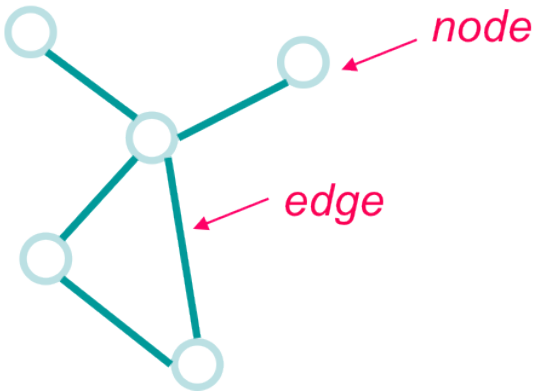
Represent biological systems

- protein-protein interaction networks, gene regulation networks, metabolic pathways, etc.



Representing networks

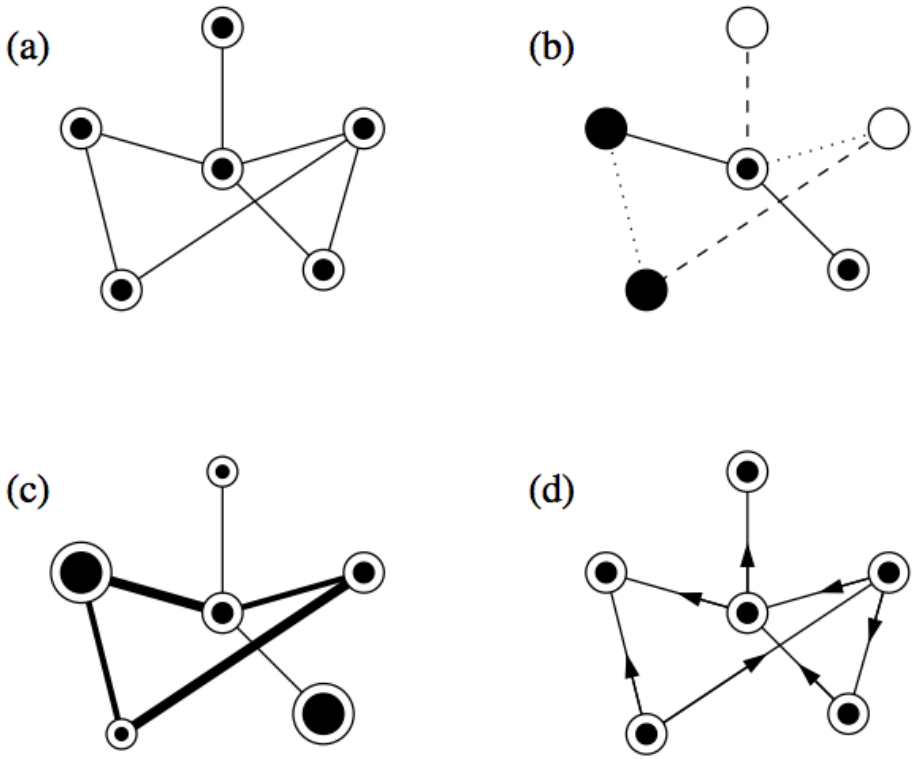
- Network ≡ Graph
- Networks are just collections of "points" joined by "lines"



points	lines	
vertices	edges, arcs	math
nodes	links	computer science
sites	bonds	physics
actors	ties, relations	sociology

Types of networks

From [Newman, 2003]



(a) unweighted, undirected

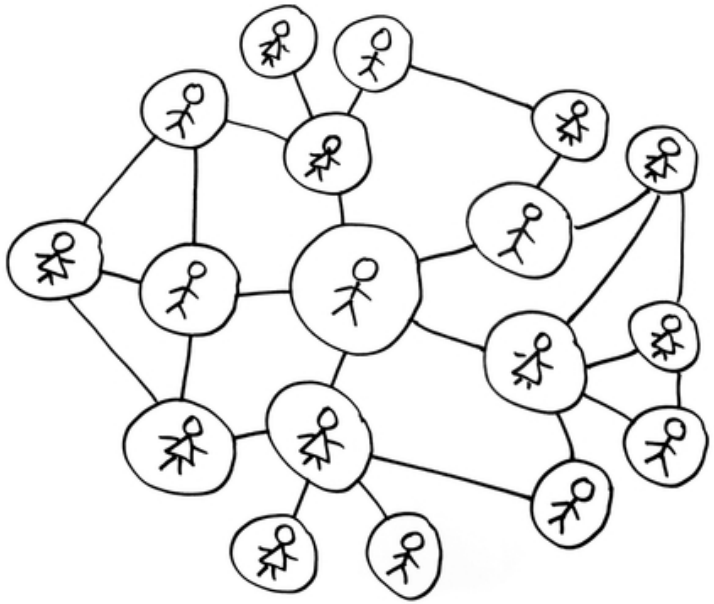
(b) discrete vertex and edge types, undirected

(c) varying vertex and edge weights, undirected

(d) directed

Small-world phenomenon

- A friend of a friend is also frequently a friend
- Only 6 hops separate any two people in the world



Measuring the small-world phenomenon, I

- Let d_{ij} be the shortest-path distance between nodes i and j
- To check whether "any two nodes are within 6 hops", we use:
  - The diameter (longest shortest-path distance) as

$$d = \max_{i,j} d_{ij}$$

  - The average shortest-path length as

$$l = \frac{2}{n\,(n+1)} \sum_{i>j} d_{ij}$$

  - The harmonic mean shortest-path length as

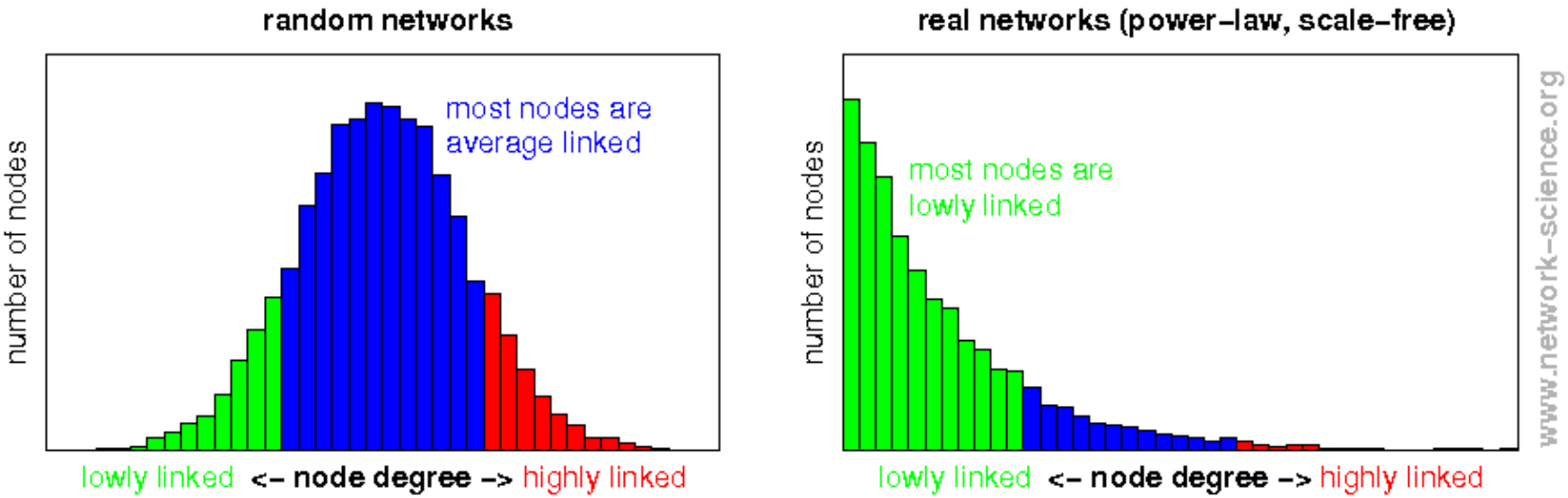
$$l^{-1} = \frac{2}{n\,(n+1)} \sum_{i>j} d_{ij}^{-1}$$

From [Newman, 2003]

	network	type	n	m	z	ℓ	α	$C^{(1)}$	$C^{(2)}$	r	Ref(s).
social	film actors	undirected	449 913	25 516 482	113.43	3.48	2.3	0.20	0.78	0.208	20, 416
	company directors	undirected	7 673	55 392	14.44	4.60	–	0.59	0.88	0.276	105, 323
	math coauthorship	undirected	253 339	496 489	3.92	7.57	–	0.15	0.34	0.120	107, 182
	physics coauthorship	undirected	52 909	245 300	9.27	6.19	–	0.45	0.56	0.363	311, 313
	biology coauthorship	undirected	1 520 251	11 803 064	15.53	4.92	–	0.088	0.60	0.127	311, 313
	telephone call graph	undirected	47 000 000	80 000 000	3.16		2.1				8, 9
	email messages	directed	59 912	86 300	1.44	4.95	1.5/2.0		0.16		136
	email address books	directed	16 881	57 029	3.38	5.22	–	0.17	0.13	0.092	321
	student relationships	undirected	573	477	1.66	16.01	–	0.005	0.001	−0.029	45
	sexual contacts	undirected	2 810				3.2				265, 266
information	WWW `nd.edu`	directed	269 504	1 497 135	5.55	11.27	2.1/2.4	0.11	0.29	−0.067	14, 34
	WWW Altavista	directed	203 549 046	2 130 000 000	10.46	16.18	2.1/2.7				74
	citation network	directed	783 339	6 716 198	8.57		3.0/–				351
	Roget's Thesaurus	directed	1 022	5 103	4.99	4.87	–	0.13	0.15	0.157	244
	word co-occurrence	undirected	460 902	17 000 000	70.13		2.7		0.44		119, 157
technological	Internet	undirected	10 697	31 992	5.98	3.31	2.5	0.035	0.39	−0.189	86, 148
	power grid	undirected	4 941	6 594	2.67	18.99	–	0.10	0.080	−0.003	416
	train routes	undirected	587	19 603	66.79	2.16	–		0.69	−0.033	366
	software packages	directed	1 439	1 723	1.20	2.42	1.6/1.4	0.070	0.082	−0.016	318
	software classes	directed	1 377	2 213	1.61	1.51	–	0.033	0.012	−0.119	395
	electronic circuits	undirected	24 097	53 248	4.34	11.05	3.0	0.010	0.030	−0.154	155
	peer-to-peer network	undirected	880	1 296	1.47	4.28	2.1	0.012	0.011	−0.366	6, 354
biological	metabolic network	undirected	765	3 686	9.64	2.56	2.2	0.090	0.67	−0.240	214
	protein interactions	undirected	2 115	2 240	2.12	6.80	2.4	0.072	0.071	−0.156	212
	marine food web	directed	135	598	4.43	2.05	–	0.16	0.23	−0.263	204
	freshwater food web	directed	92	997	10.84	1.90	–	0.20	0.087	−0.326	272
	neural network	directed	307	2 359	7.68	3.97	–	0.18	0.28	−0.226	416, 421

Degree distribution

Histogram of nr of nodes having a particular degree



f_k = fraction of nodes of degree k

Scale-free networks

The degree distribution of most real-world networks follows a power-law distribution

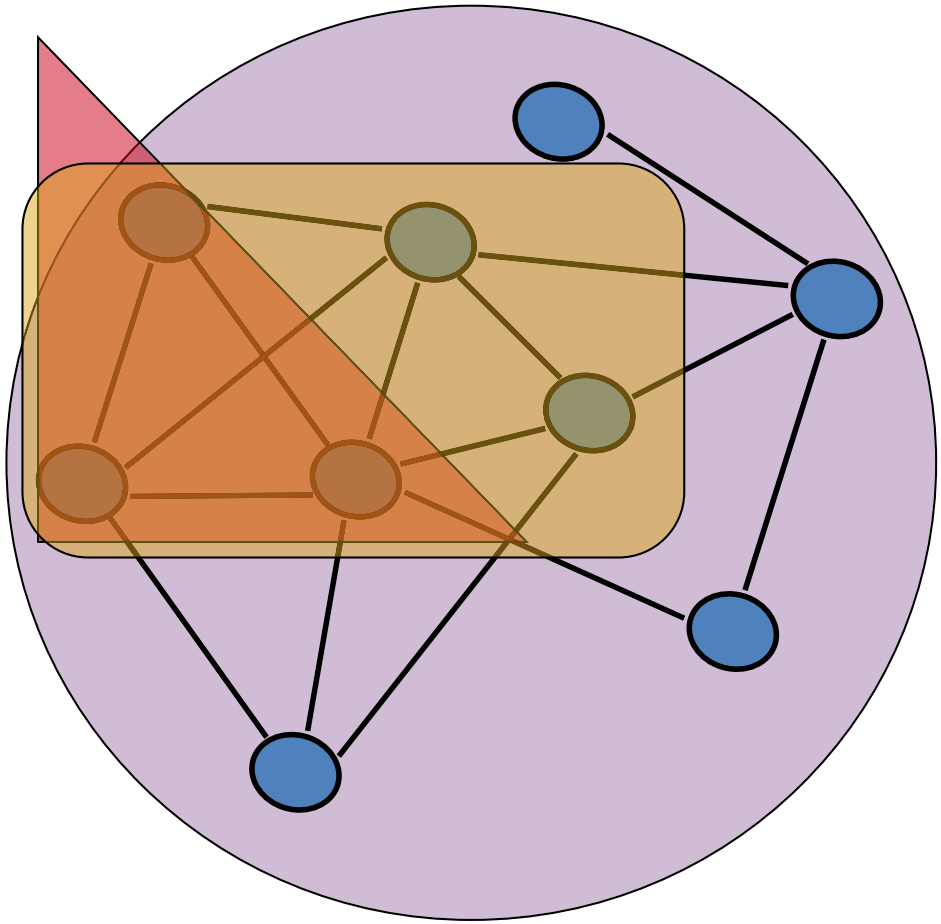
$$f_k = ck^{-\alpha}$$

- "heavy-tail" distribution, implies existence of hubs
- hubs are nodes with very high degree

How to Analyze Networks

Slides from Johannes Putzke, Social Network Analysis: Basic Concepts, Methods & Theory

Different Levels of Analysis

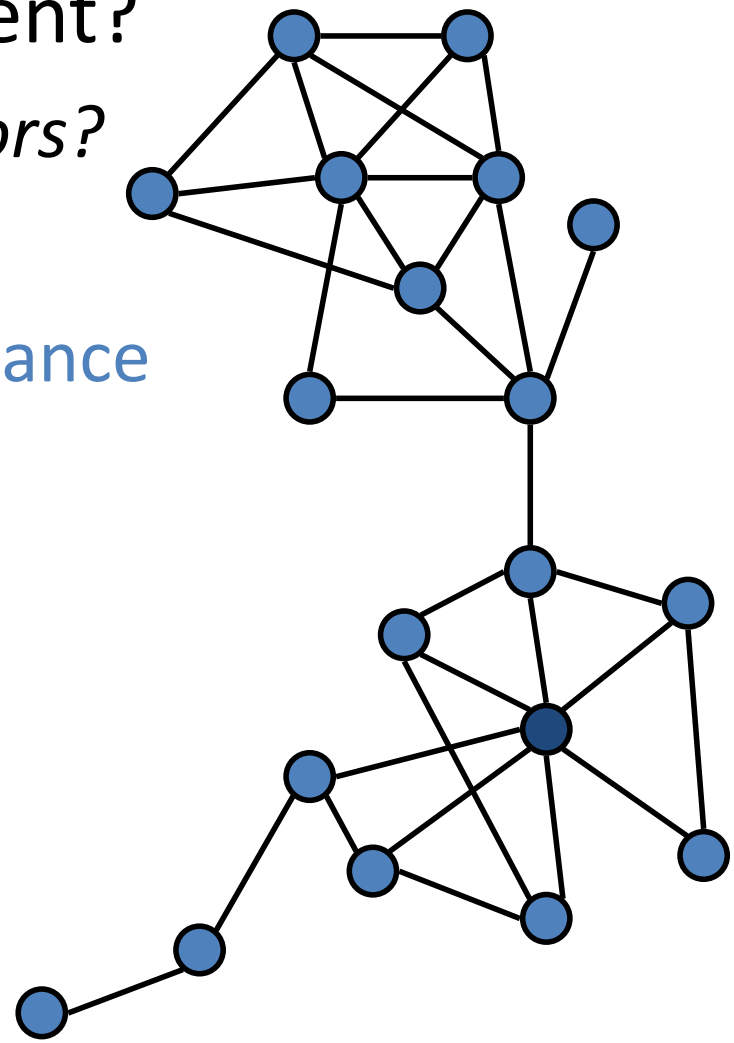


- Actor-Level
- Dyad-Level
- Triad-Level
- Subset-level (cliques / subgraphs)
- Group (i.e. global) level

Example: Centrality Measures

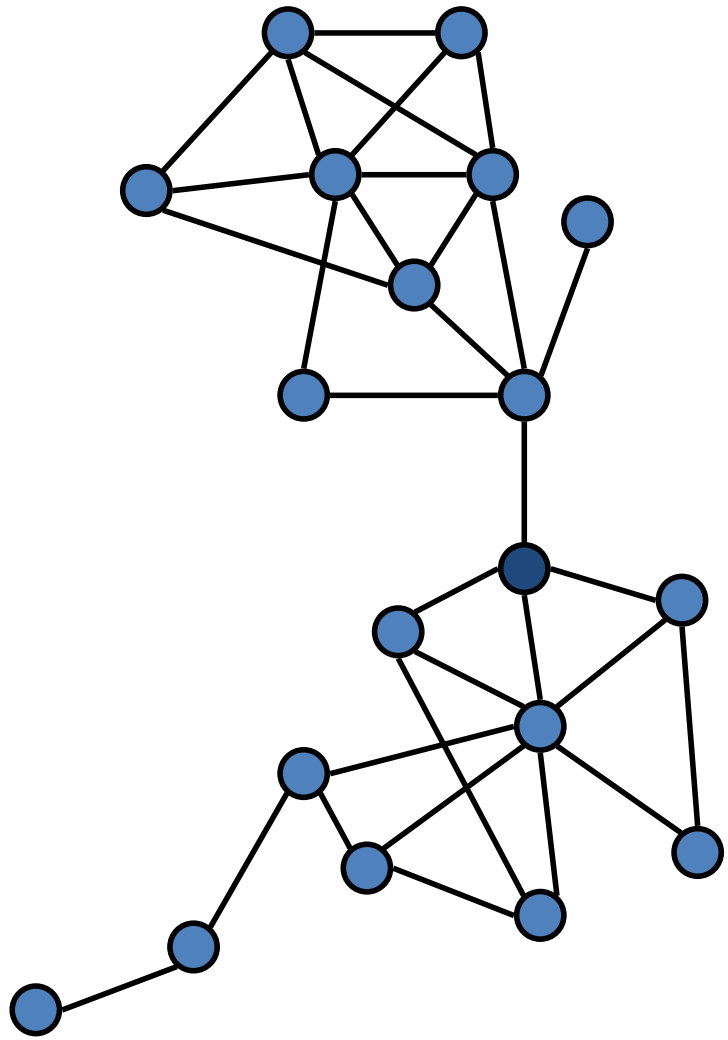
- Who is the most prominent?
  - *Who knows the most actors?*

    (Degree Centrality)
  - Who has the shortest distance to the other actors?
  - Who controls knowledge flows?
  - ...



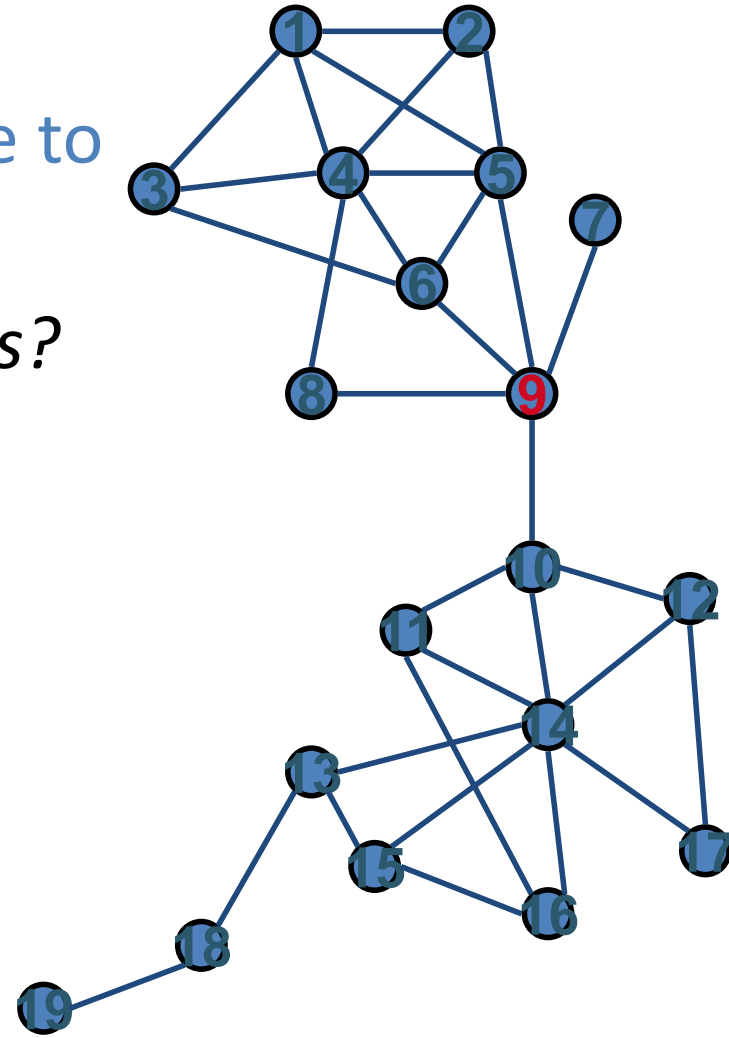
Closeness Centrality

- Who knows the most actors?
- *Who has the shortest distance to the other actors? (Closesness Centrality)*
- Who controls knowledge flows?
- ...



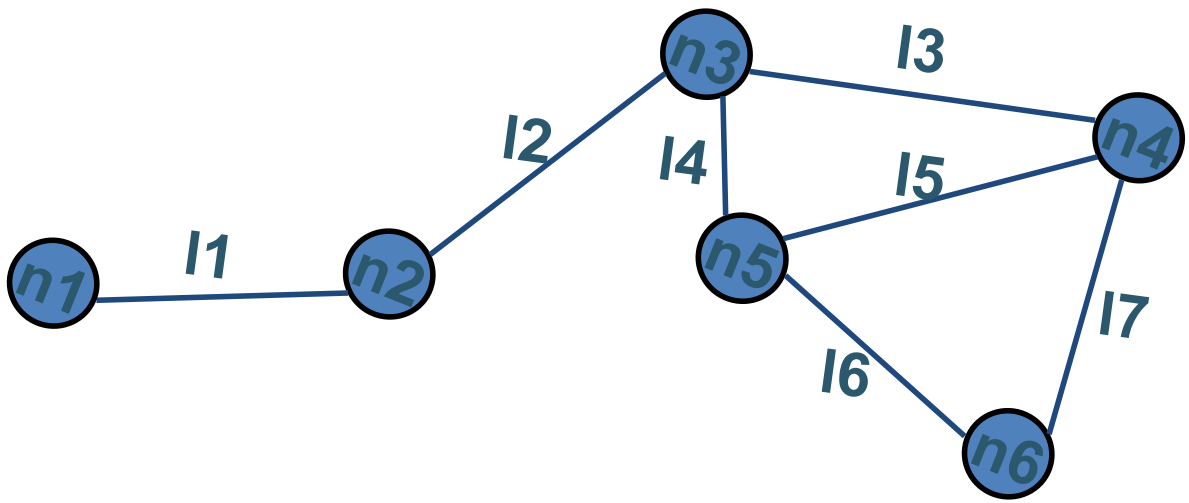
Betweenness Centrality

- Who knows the most actors?
- Who has the shortest distance to the other actors?
- *Who controls knowledge flows? (Betweenness Centrality)*
- ...



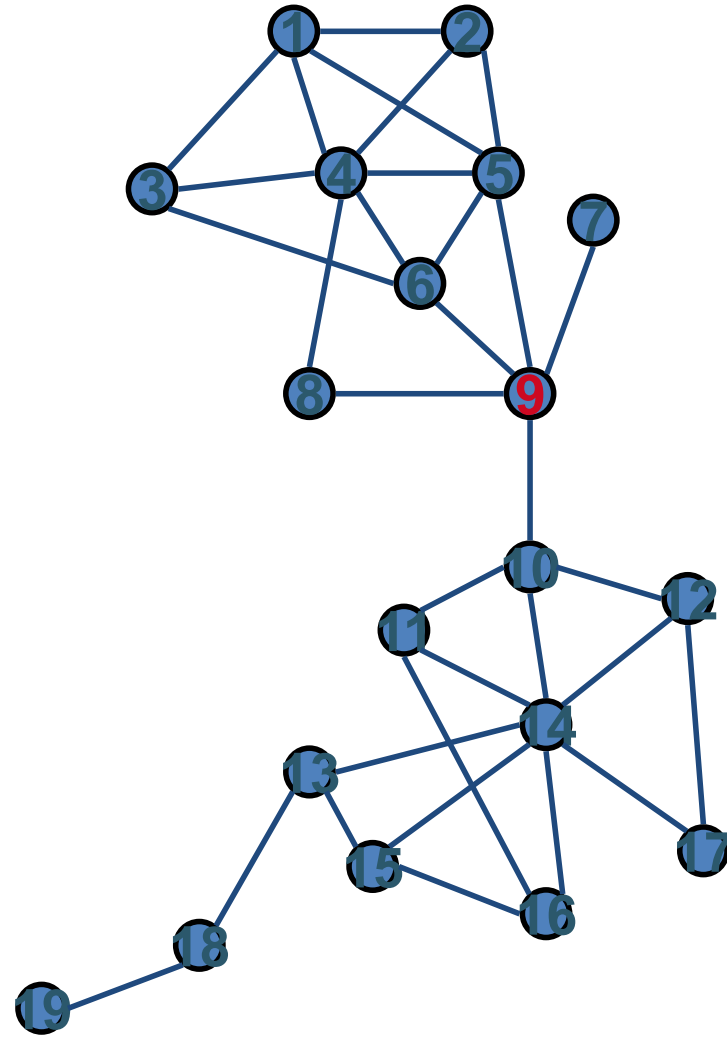
Reachability, Distances and Diameter

- *Reachability*
  - If there is a path between nodes n_i and n_j
- *Geodesic*
  - Shortest path between two nodes
- *(Geodesic) Distance* d(i,j)
  - Length of Geodesic (also called „degrees of separation“)



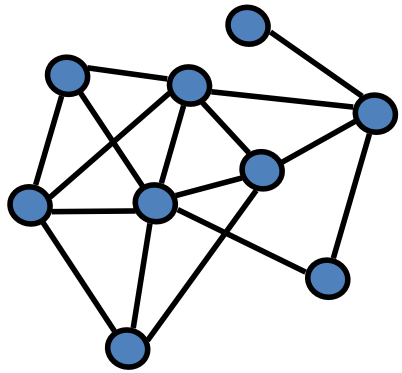
Diameter of a Graph and Average Geodesic Distance

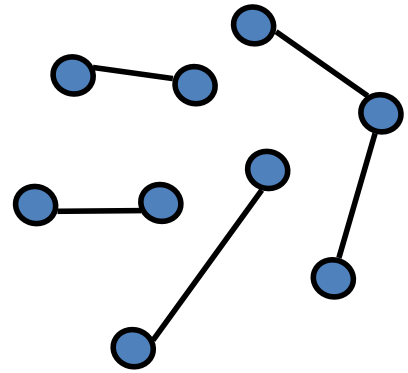
- Diameter
  - Largest geodesic distance between any pair of nodes
- Average Geodesic Distance
  - How fast can information get transmitted?



Density

- Proportion of ties in a graph





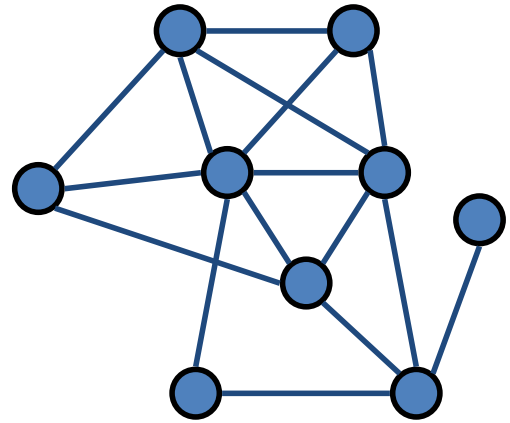
High density (44%)

Low density (14%)

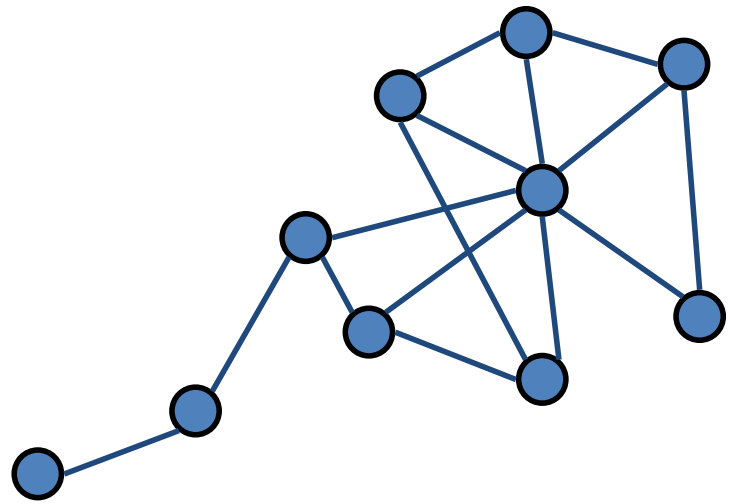
Connectivity of Graphs

Connected Graphs, Components, Cutpoints and Bridges

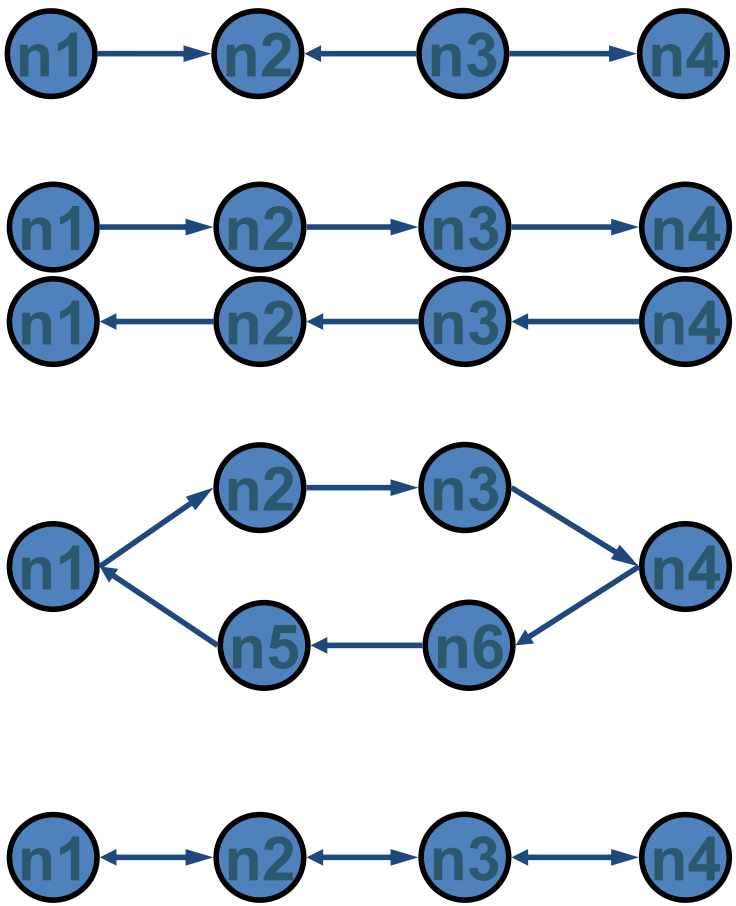
- *Connectedness*
  - A graph is connected if there is a path between every pair of nodes



- *Components*
  - Connected subgraphs in a graph
  - Connected graph has 1 component
  - Two disconnected graphs are one social network!!!



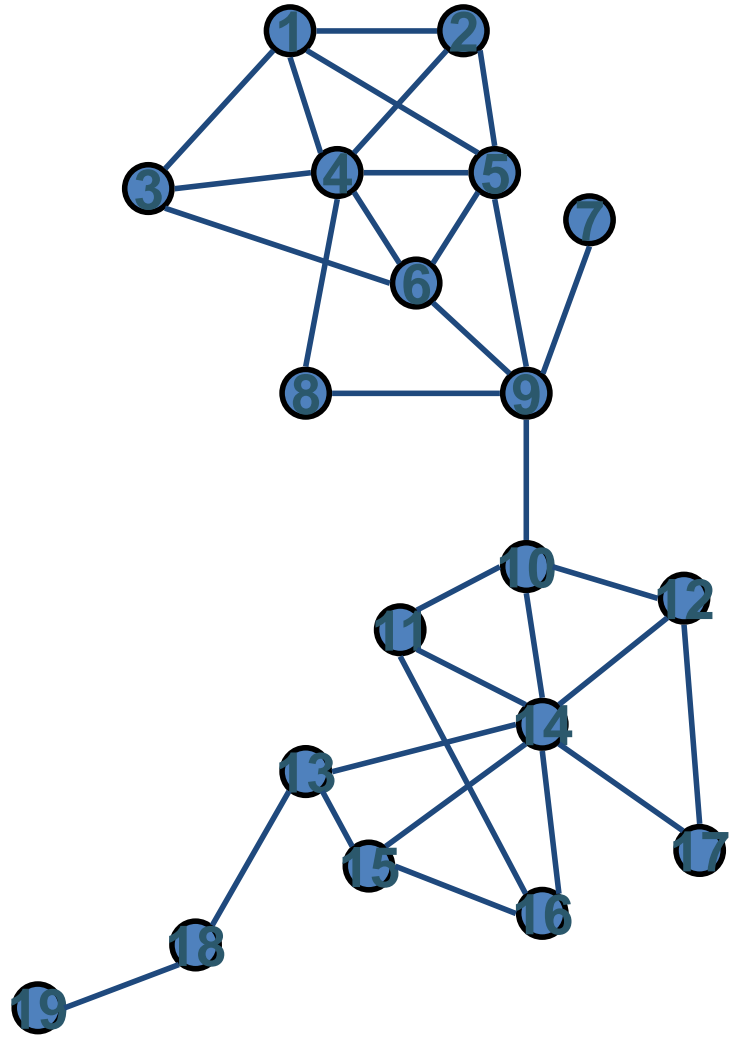
Connected Graphs, Components, Cutpoints and Bridges



- Connectivity of pairs of nodes and graphs
  - *Weakly connected*
    - Joined by semipath
  - *Unilaterally connected*
    - Path from n_j to n_j or from n_j to n_j
  - *Strongly connected*
    - Path from n_j to n_j *and* from n_j to n_j
    - Path may contain different nodes
  - *Recursively Connected*
    - Nodes are strongly connected and both paths use the same nodes and arcs in reverse order

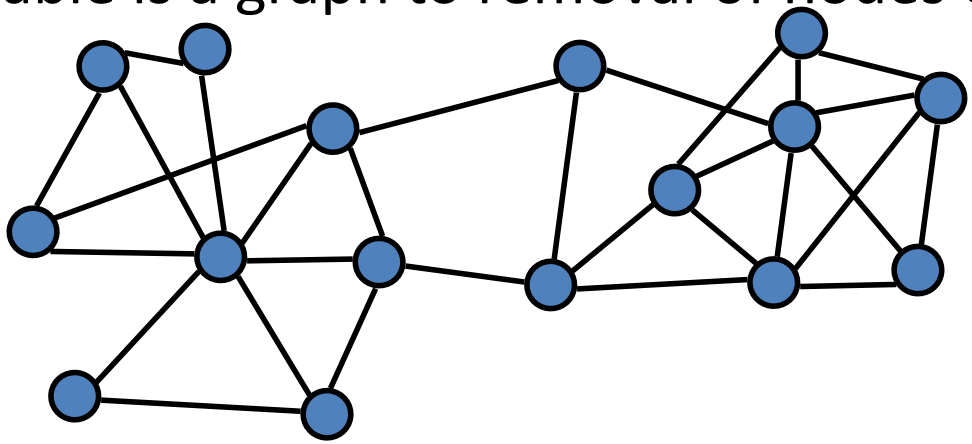
Connected Graphs, Components, Cutpoints and Bridges

- *Cutpoints*
  - number of components in the graph that contain node n_j is fewer than number of components in subgraphs that results from deleting n_j from the graph
- *Cutsets (of size k)*
  - *k*-node cut
- *Bridges / line cuts*
  - Number of components…that contain line l_k



Node- and Line Connectivity

- How vulnerable is a graph to removal of nodes or lines?



Point connectivity / Node connectivity

- Minimum number of *k* for which the graph has a *k*-node cut
- For any value <k the graph is *k-node-connected*

Line connectivity / Edge connectivity

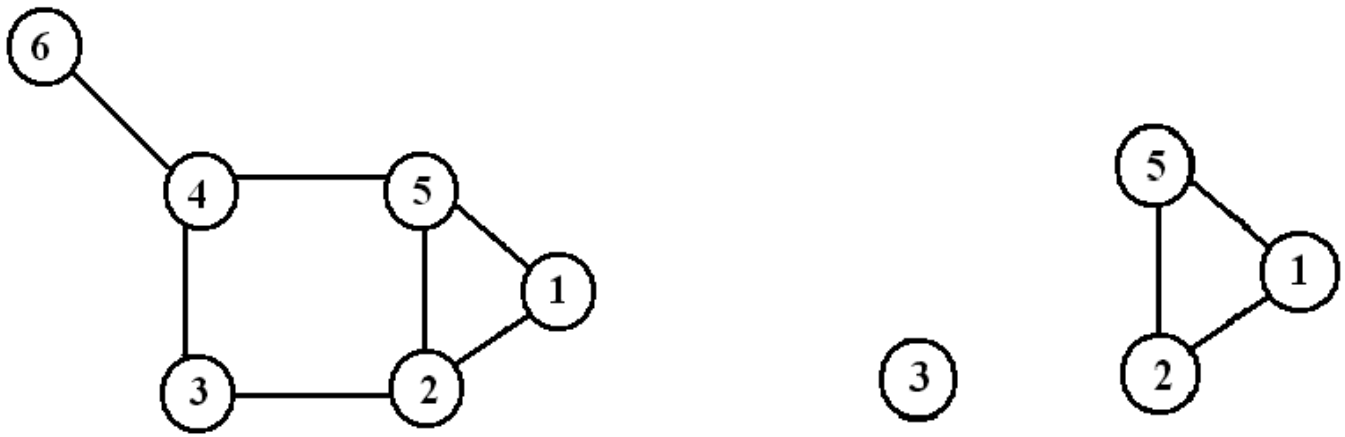
- Minimum number λ for which for which graph has a λ-line cut

How to Analyze Networks (cont.)

Slides from Jon Crowcroft, Introduction to Network Theory

Subgraph

- Vertex and edge sets are subsets of those of G
  - a *supergraph* of a graph G is a graph that contains G as a subgraph.



Isomorphism

- Bijection, i.e., a one-to-one mapping:

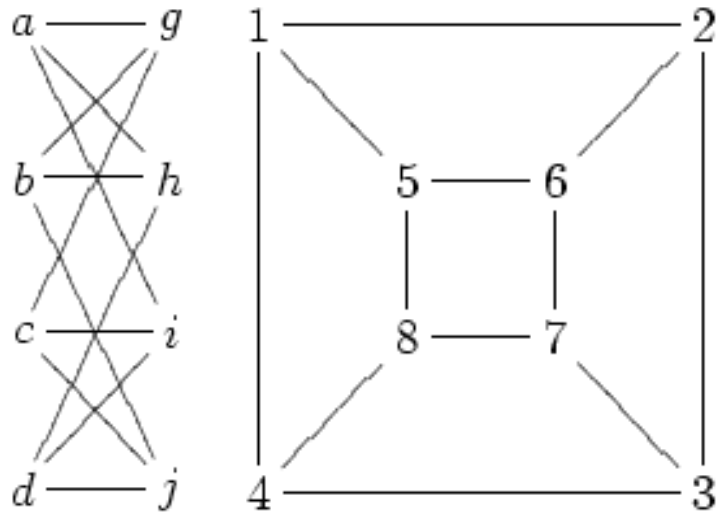
  f : V(G) -> V(H)

  u and v from G are adjacent if and only if f(u) and f(v) are adjacent in H.
- If an isomorphism can be constructed between two graphs, then we say those graphs are ***isomorphic***.

Isomorphism Problem

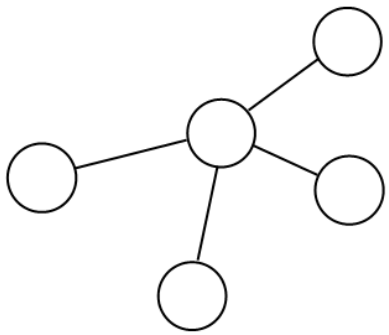
- Determining whether two graphs are isomorphic
- Although these graphs look very different, they are isomorphic; one isomorphism between them is

  f(a)=1 f(b)=6 f(c)=8 f(d)=3
  f(g)=5 f(h)=2 f(i)=4 f(j)=7



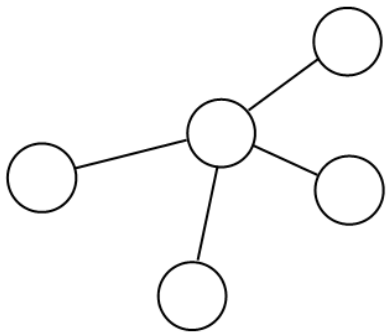
Analyzing using subgraph counting (more cont.)

Subgraph Counting

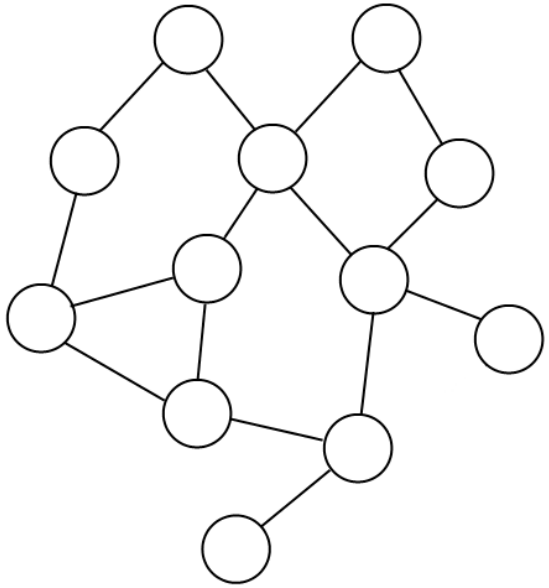


Template

Subgraph Counting

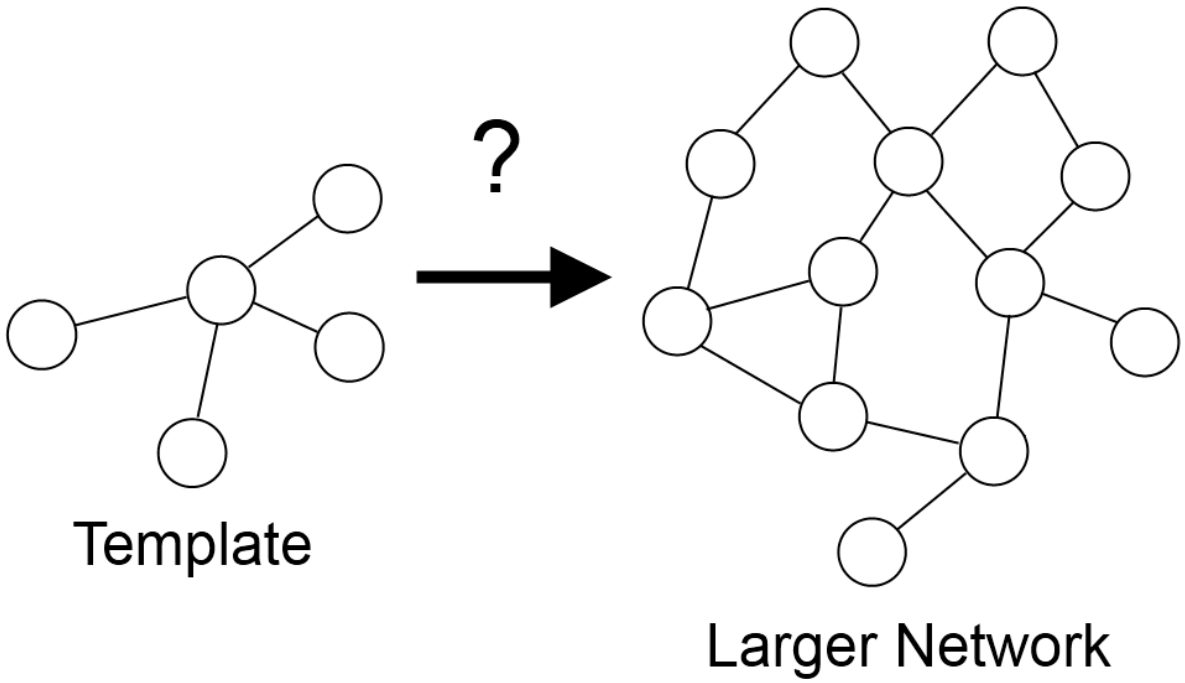


Template

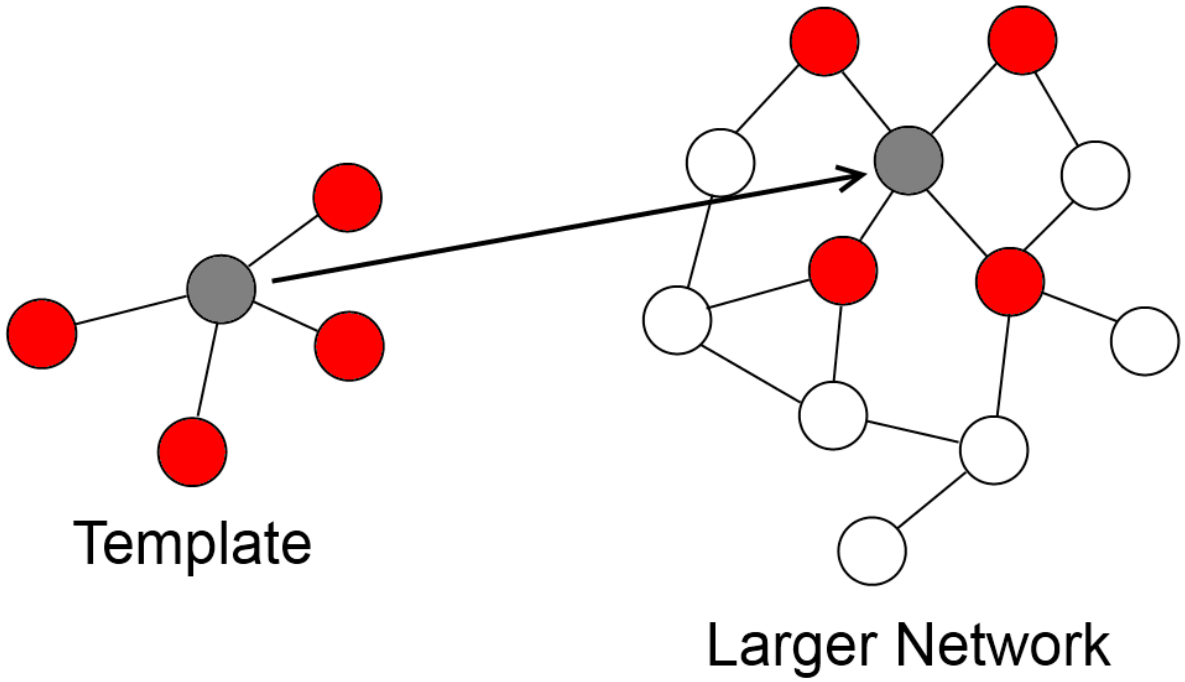


Larger Network

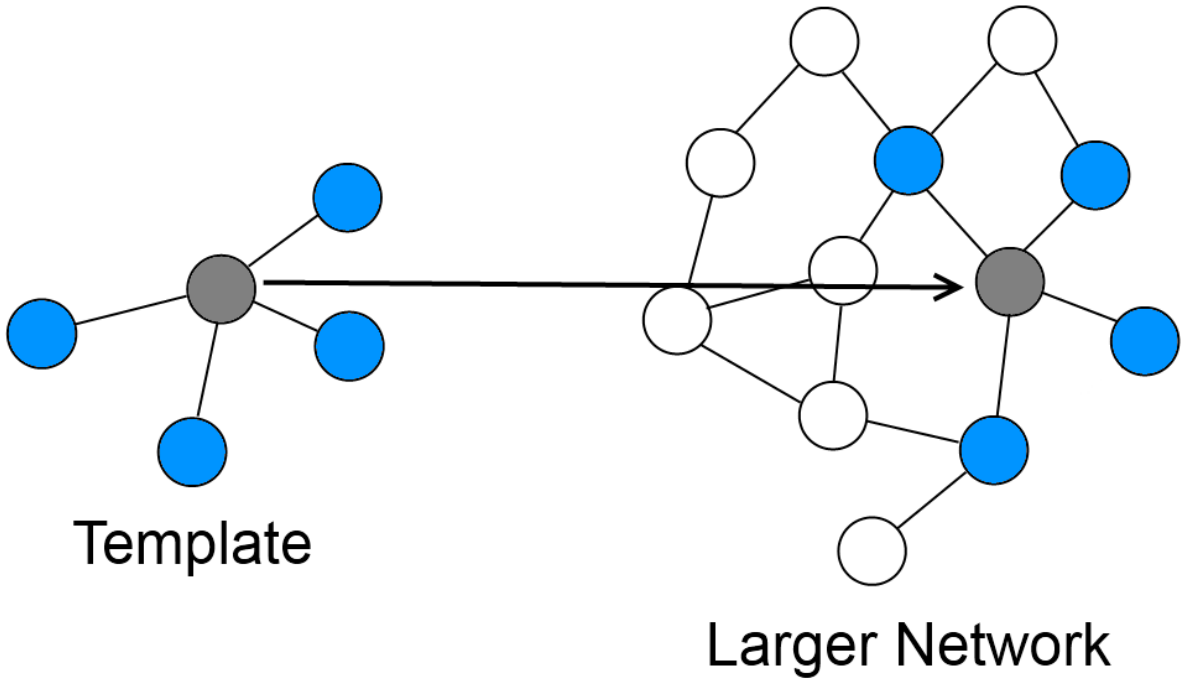
Subgraph Counting



Subgraph Counting



Subgraph Counting



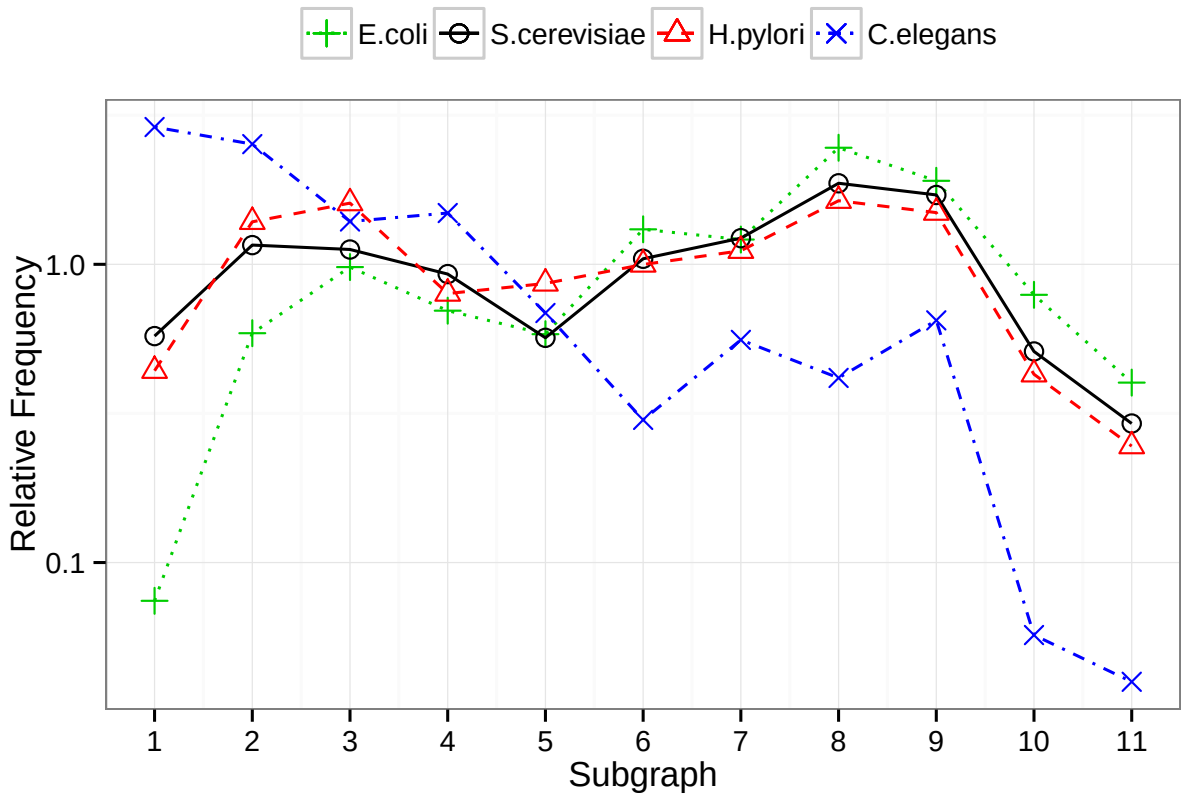
Motivations for Subgraph Counting, Path Finding

Why do we want fast algorithms for subgraph counting and weighted path finding?

- Important to social network analysis, communication network analysis,bioinformatics, chemoinformatics, etc.
- Forms basis of more complex analysis
  - Motif finding, anomaly detection
  - Graphlet frequency distance (GFD)
  - Graphlet degree distributions (GDD)
  - Graphlet degree signatures (GDS)
- Counting and enumeration on large networks is very tough, $O(n^k)$ complexity for naïve algorithm
- Finding minimum-weight paths – NP-hard problem

Motif Finding

- Motif finding: Look for all subgraphs of a certain size (and structure)
- Highly occuring subgraphs can have structural significance



Graphlet Frequency Distance Analysis

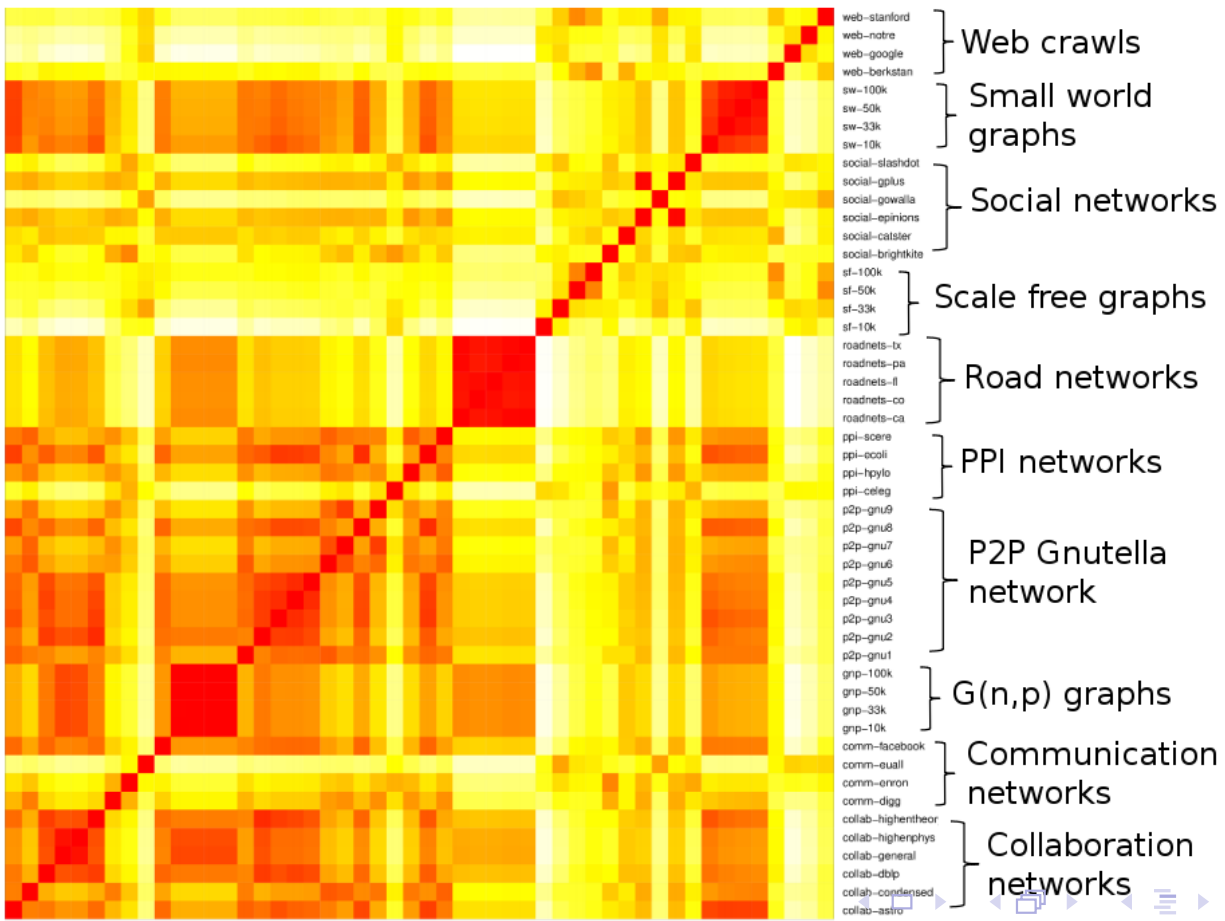
- GFD: Numerically compare occurrence frequency to other networks
- $S_i(G)$ – relative frequency for subgraph i in graph G
- C_i – counts of subgraph i
- $D(G,H)$ – frequency distance between two graphs G,H

$$S_i(G) = -\log\left(\frac{C_i(G)}{\sum_{i=1}^{\mathrm{n}} C_i(G)}\right)$$

$$D(G,H) = \sum_{i=1}^{\mathrm{n}} |S_i(G) - S_i(H)|$$

Graphlet Frequency Distance Analysis

- GFD: Numerically compare occurrence frequency to other networks
- Heatmap of distances between many networks (red = similar, white = dissimilar)
- Note occurrence of high intra-network type similarities

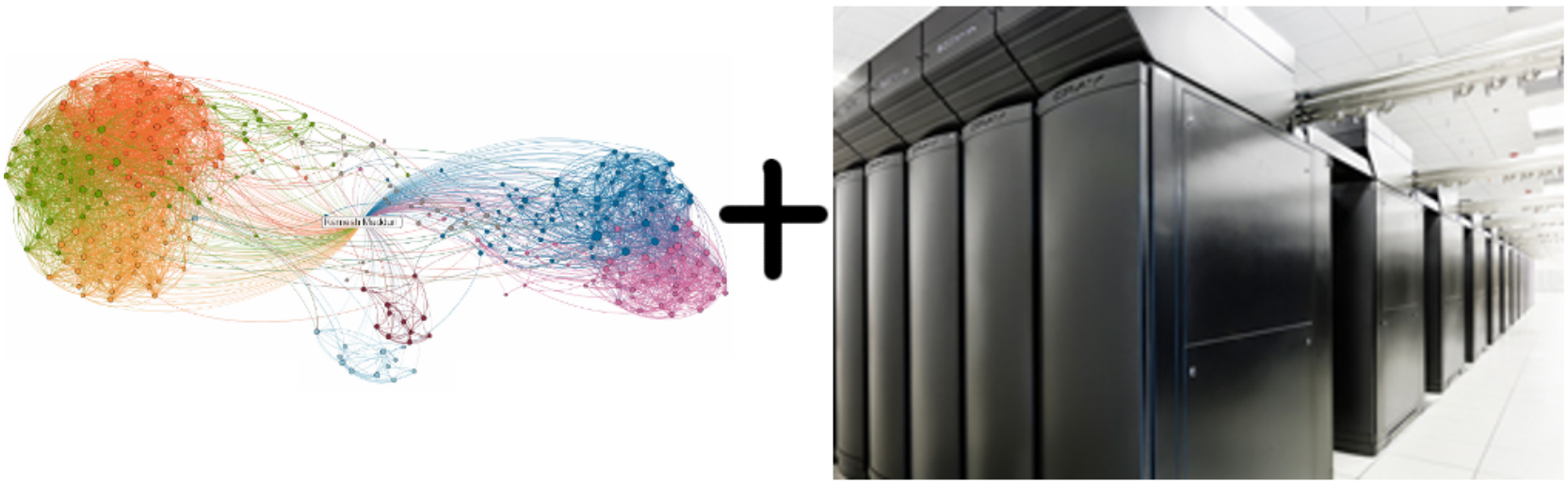


Computational Aspects for Massive Graphs

AKA why efficient parallelization is important - i.e. the point of this class

What?

Graph Analytics and HPC



Or, given modern extreme-scale graph-structured datasets (web crawls, brain graphs, human interaction networks) and modern high performance computing systems (Blue Waters), how can we develop a generalized approach to efficiently study such datasets on such systems?

Why?

Why do want to study these large graphs?

Human Interaction Graphs:

- Finding hidden communities, individuals, malicious actors
- Observe how information and knowledge propagates

Brain Graphs:

- Study the topological properties of neural connections
- Finding latent computational substructures, similarities to other information processing systems

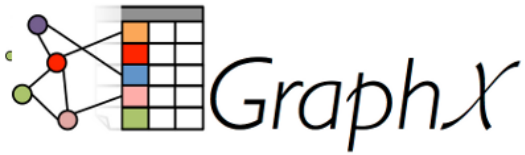
Web Crawls:

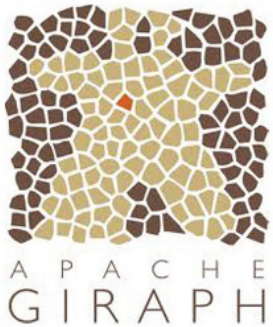
- Identifying trustworthy/important sites
- Spam networks, untrustworthy sites

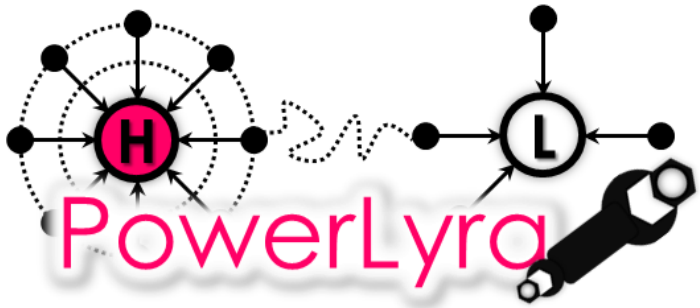
Prior Approaches

Can we use them to analyze large graphs on HPC?

- Some limited by shared-memory and/or specialized hardware
- Some run in distributed memory but graph scale is still limited
- Others, graph scale isn't limiting factor but performance can be







Graph analytics on HPC

So why do we want to run graph analytics on HPC?

- Scalability for analytic performance and graph size
  - Efficient implementations should be limited only by distributed memory capacity
  - Graph500.org - demonstration of performance achievable for irregular computations through breadth-first search (BFS)
- Relative availability of access in academic/research communities
  - Private clusters of various scales, shared supercomputers
  - Access for domain experts, those using analytics on real-world graphs

Can we create an approach that is as simple to use as the aforementioned frameworks but runs on common cluster hardware and gives state-of-the-art performance?

Challenges

Scale

- This work considers "extreme-scale" graphs – billion+ vertices and up to trillion+ edges
- Processing these graphs requires at least hundreds to thousands of compute nodes or tens of thousands of cores
- Graph analytic algorithms are generally memory-bound instead of compute-bound; in the distributed space, this results in a ratio of communication versus computation that increases with core/node count

Complexity

- Real-world extreme-scale graphs have similar characteristics: small-world nature with skewed degree distributions
- Small-world graphs are difficult to partition for distributed computation or to optimize in terms of cache due to "too much locality"
- Skewed degree distributions make efficient parallelization and load balance difficult to achieve
- Multiple levels of cache/memory and increasing reliance on wide parallelism for modern HPC systems compounds the above challenges