Applied Data Analysis (CS401)

Lecture 12
Handling network data
6 Dec 2023

EPFL

Robert West

dlab

Announcements

- Homework H2 is being graded
  - Feedback to be released next week
- Final project milestone P3 due on Fri 22 Dec 2023
- Friday's lab session:
  - Project office hour (on Zoom)
    - Second-to-last project office hour before due date
    - To secure your team's personal time slot, follow protocol described in this Ed post
  - Exercises on handling networks (in BCH 2201)
    - In parallel to office hour

Give us feedback on this lecture here:
https://go.epfl.ch/ada2023-lec12-feedback

- What did you (not) like about this lecture?
- What was (not) well explained?
- On what would you like more (fewer) details?
- …

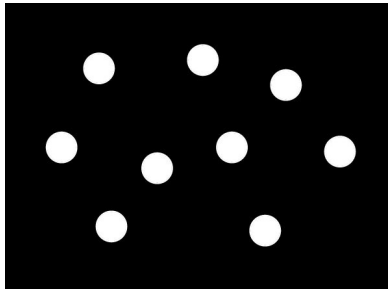
Beyond flat tables

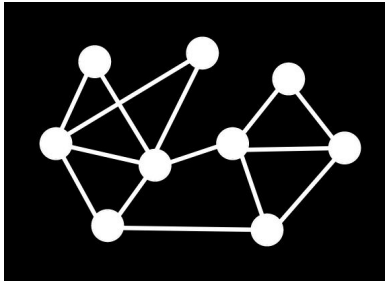
People

id	name	age
1	Bob	36
2	Willy	32
…	…	…

Marriages

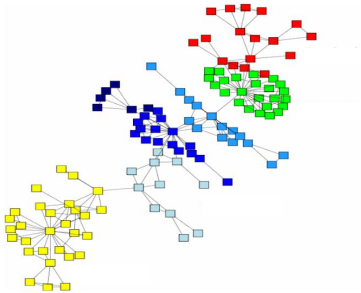
husband_id	wife_id
1	34
2	5
2	87
…	…



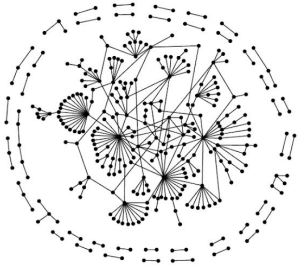


Examples

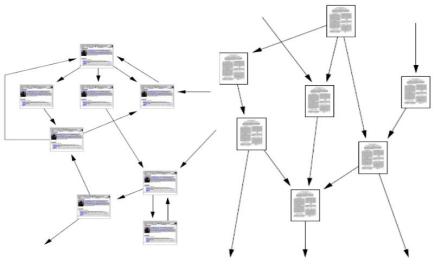
Social networks



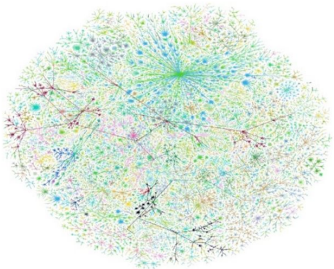
Economic networks



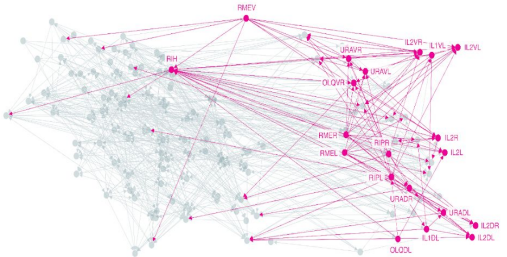
Communication graphs



Information networks:
Web & citations

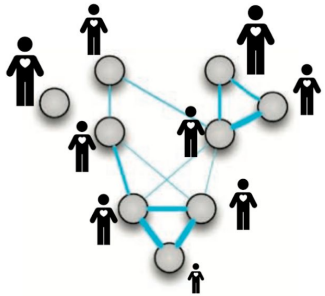


Internet

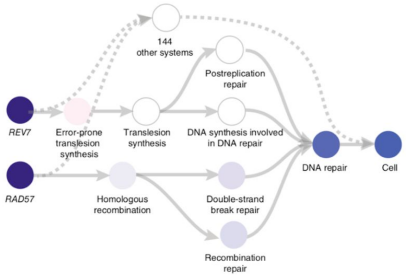


Networks of neurons

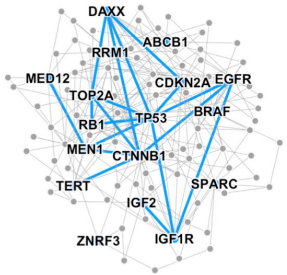
Examples



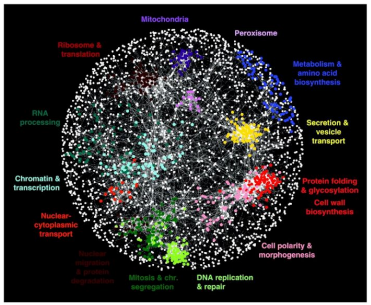
Patient networks



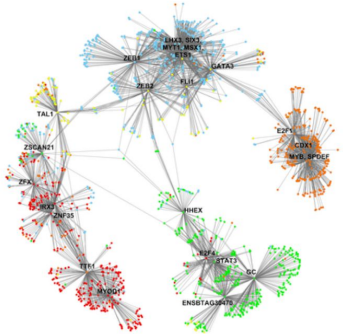
Hierarchies of cell systems



Disease pathways



Genetic interaction networks



Gene co-expression networks

Cell-cell similarity networks

Networks as graphs

- **Network:** a real-world system of dependent variables, e.g.,
  - WWW is a network of hyperlinked documents
  - Society is a network of individuals linked by family, friendship, professional ties
  - Metabolic network is sum of all chemical reactions in a cell
- **Graph:** mathematical abstraction for describing networks
- In practice, “network” and “graph” are used interchangeably
- You can make a graph out of almost anything (e.g., connect all people whose name starts with the same letter), so must ask: Does this graph correspond to a meaningful network?

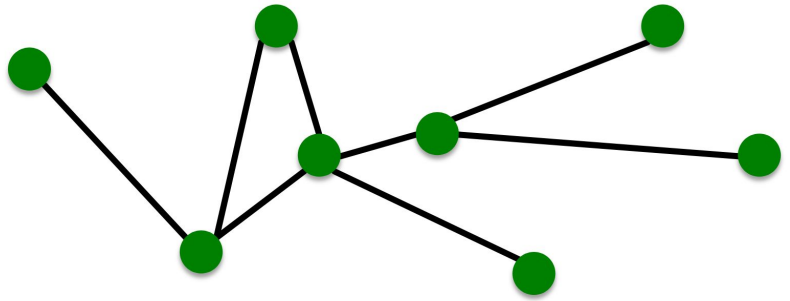
Today's lecture

- **Part 1:** Types of graphs
- **Part 2:** Representing graphs on computers
- **Part 3:** Properties of real-world networks
- **Part 4:** Measuring node importance

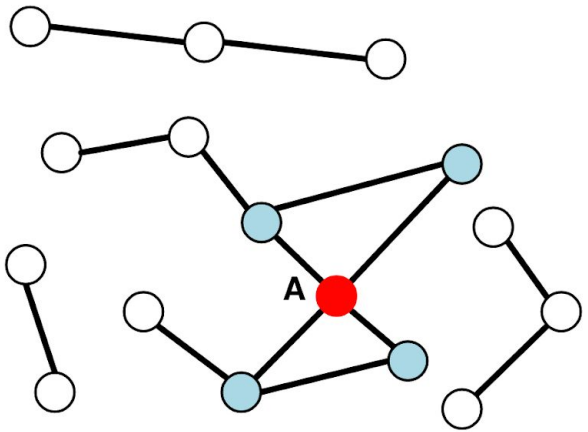
Part 1:
Types of graphs

Most basic type: undirected graphs

- Entities:
  nodes/vertices V
- Relationships/interactions:
  edges/links E
- Entire system:
  graph $G = (V, E)$



Node degree



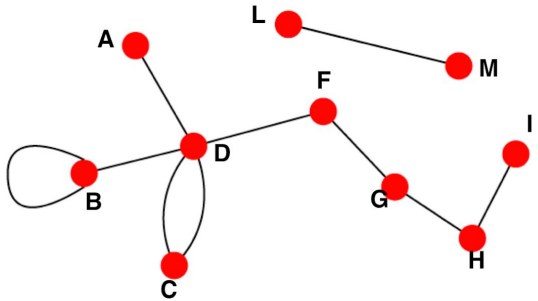
Node degree, k_i: the number of edges adjacent to node ***i***

$k_A = 4$

Types of graphs: undirected vs. directed

Undirected

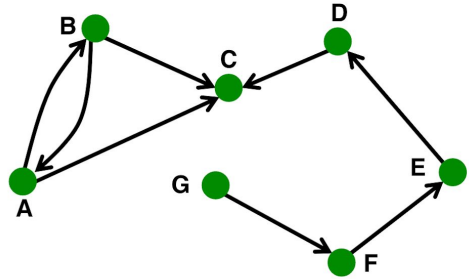
- **Links:** undirected (symmetrical, reciprocal)



- **Examples:**
  - Collaborations
  - Friendship on Facebook

Directed

- **Links:** directed (arcs)



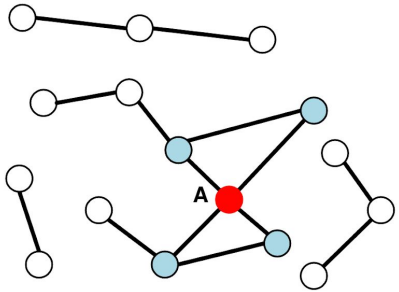
- **Examples:**
  - Phone calls
  - Following on Twitter

Average node degree

POLLING TIME

Scan QR code or go to https://web.speakup.info/room/join/66626

Undirected



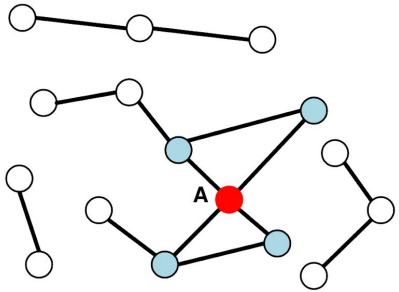
Node degree, k_i: the number of edges adjacent to node ***i***

$k_A = 4$

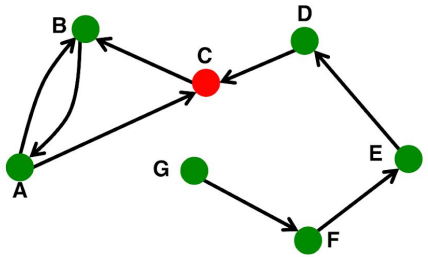
Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i =$

Average node degree

Undirected



Directed



Source: Node with $k^{in} = 0$
Sink: Node with $k^{out} = 0$

Node degree, k_i: the number of edges adjacent to node i

$k_A = 4$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}$

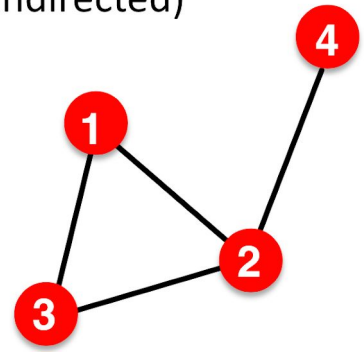
In directed networks we define an **in-degree** and **out-degree.**
The (total) degree of a node is the sum of in- and out-degrees.

$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$

$\bar{k} =$

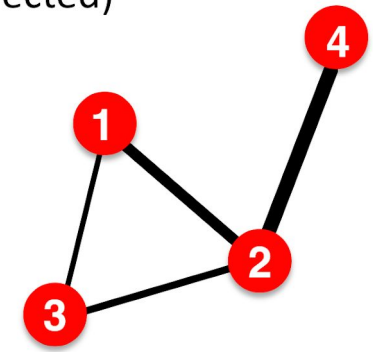
Types of graphs: weighted

- **Unweighted**
  (undirected)



Examples: Friendship, Hyperlink

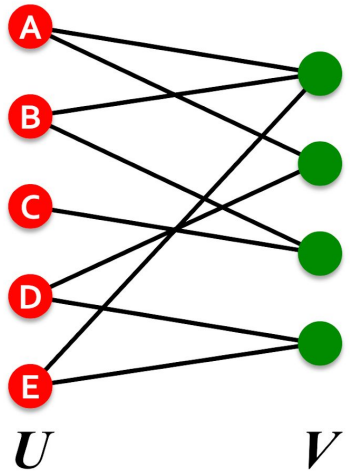
- **Weighted**
  (undirected)



Examples: Collaboration, Internet, Roads

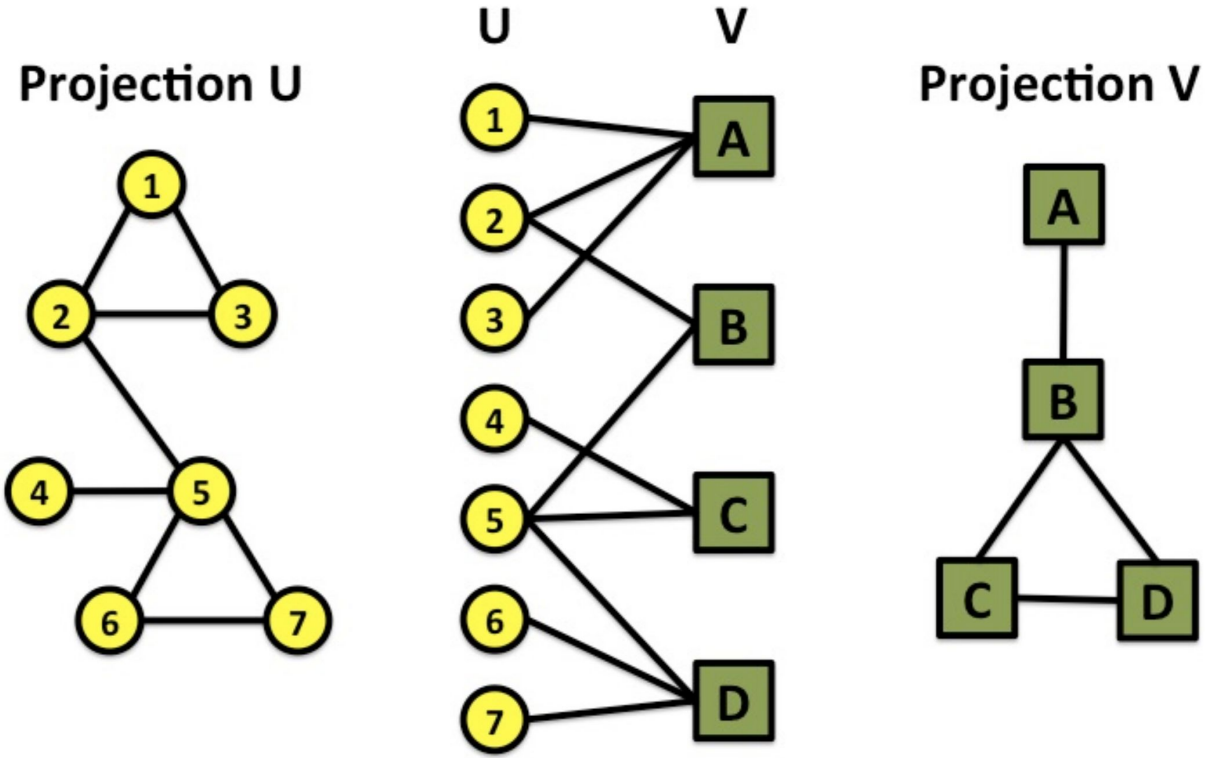
Types of graphs: bipartite

- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V

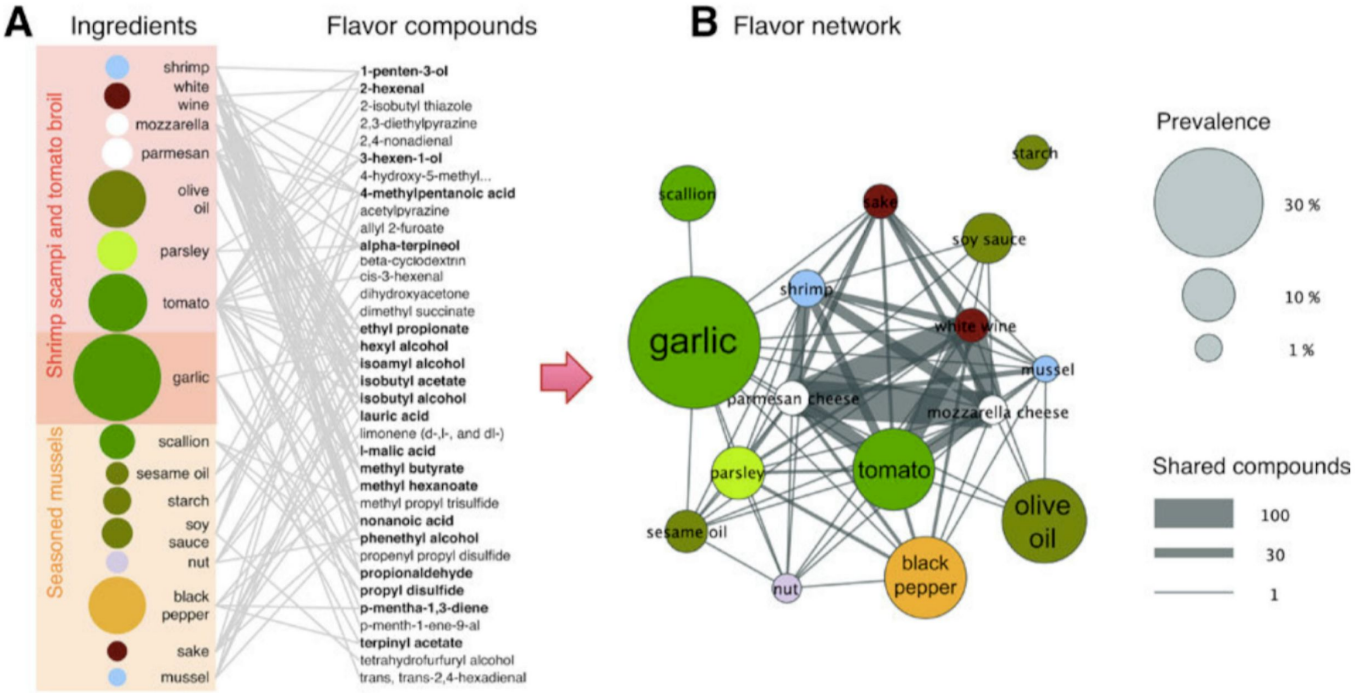


- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks (a.k.a. "projections"):**
  - Author collaboration networks
  - Movie co-rating networks

Projections of bipartite graph



Example: flavor networks

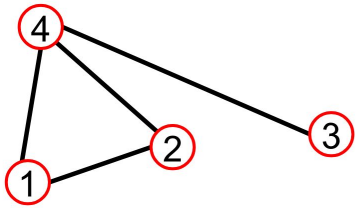


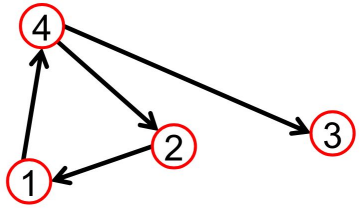
Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási
Flavor network and the principles of food pairing , Scientific Reports 196, (2011).

[paper]

Part 2:
Representing graphs on computers

Representing graphs on computers: adjacency matrix





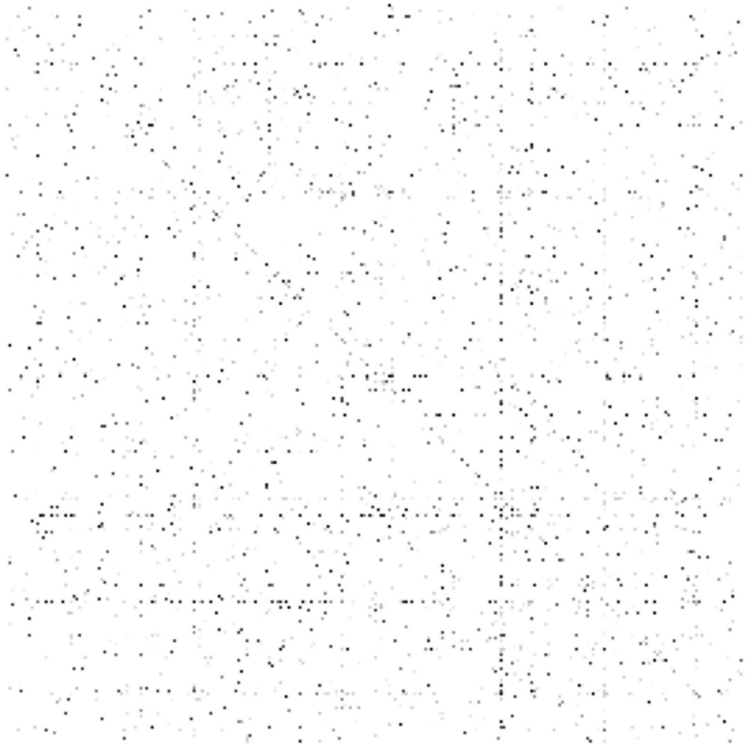
$A_{ij} = 1$ if there is a link from node *i* to node *j*

$A_{ij} = 0$ otherwise

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

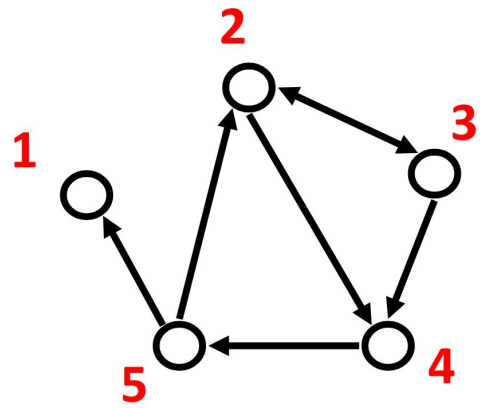
Note that for a directed graph (right) the matrix is not symmetric.

Adjacency matrix usually sparse



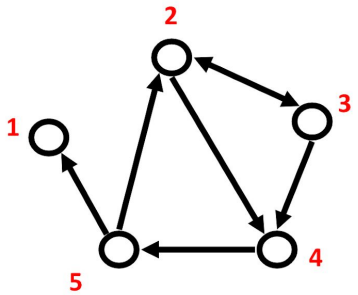
Representing graphs on computers: edge list

- **Represent graph as a set of edges:**
  - (2, 3)
  - (2, 4)
  - (3, 2)
  - (3, 4)
  - (4, 5)
  - (5, 2)
  - (5, 1)



Representing graphs on computers: adjacency list

- **Adjacency list:**
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node
    - 1:
    - 2: 3, 4
    - 3: 2, 4
    - 4: 5
    - 5: 1, 2



Graph processing libraries

- Good overview + benchmark available here
- NetworkX
  - Written in Python
  - Popular but slow
  - Ok when your graph is small
  - **This Friday's lab session: intro to NetworkX**
- NetworkKit, SNAP, iGraph, graph-tool
  - Written in C++
  - Much faster than NetworkX
  - Libraries also available in other languages (incl. Python)
  - Consider these if your graph is large

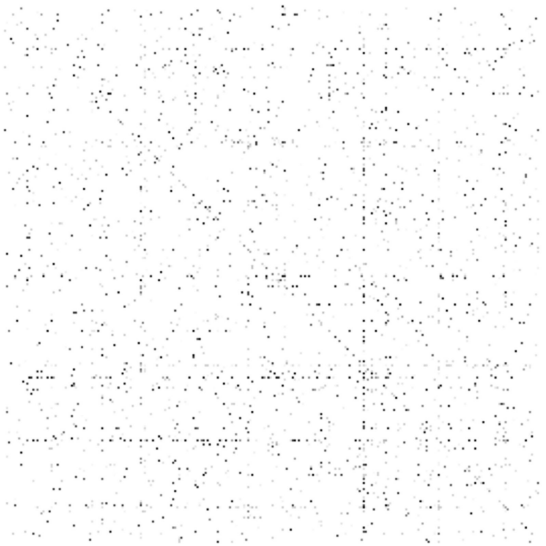
Part 3:
Properties of real-world networks

Properties of real-world networks

- Real networks are different from arbitrary graphs
- Real networks tend to share certain properties
- Remarkable, given the diversity of networks
  - Information networks (e.g., Web graph, knowledge graphs)
  - Social networks (e.g., Facebook, sexual networks)
  - Biological networks (e.g., protein–protein interaction)
  - …

Properties of real-world networks: sparsity

- Every node connected to only small fraction of all other nodes
- i.e., $k_i \ll N$
- Often bounded by a constant
  - e.g., social networks: Dunbar's number (cognitive limit to the number of people with whom one can maintain stable social relationships; allegedly 150)



Infomercial break

Today!
And happy ADAvent to all of you!

Random wiki fact:

Gemiler Island

Article Talk

From Wikipedia, the free encyclopedia

Gemiler Island (Turkish: *Gemile Adası or Gemiler Adası*, Greek: Γκεμιλέρ) is an island located off the coast of Turkey near the city of Fethiye. On the island are the remains of several churches built between the fourth and sixth centuries AD, along with a variety of associated buildings. Archaeologists believe it was the location of the original tomb of Saint Nicholas. The original Turkish name is Gemile from the Greek word καμήλα (kamila) meaning camel, so called because of its geographical shape (see Fethiye Gemile Island Archaeological Site).

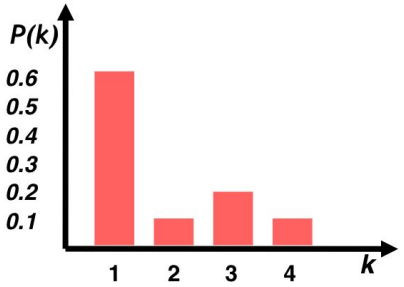
Properties of real-world networks: degree distribution

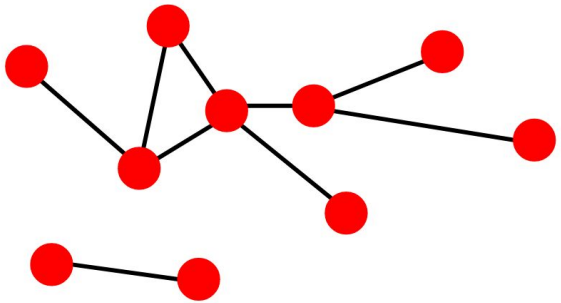
- **Degree distribution $P(k)$:** Probability that a randomly chosen node has degree k

  N_k = # nodes with degree k

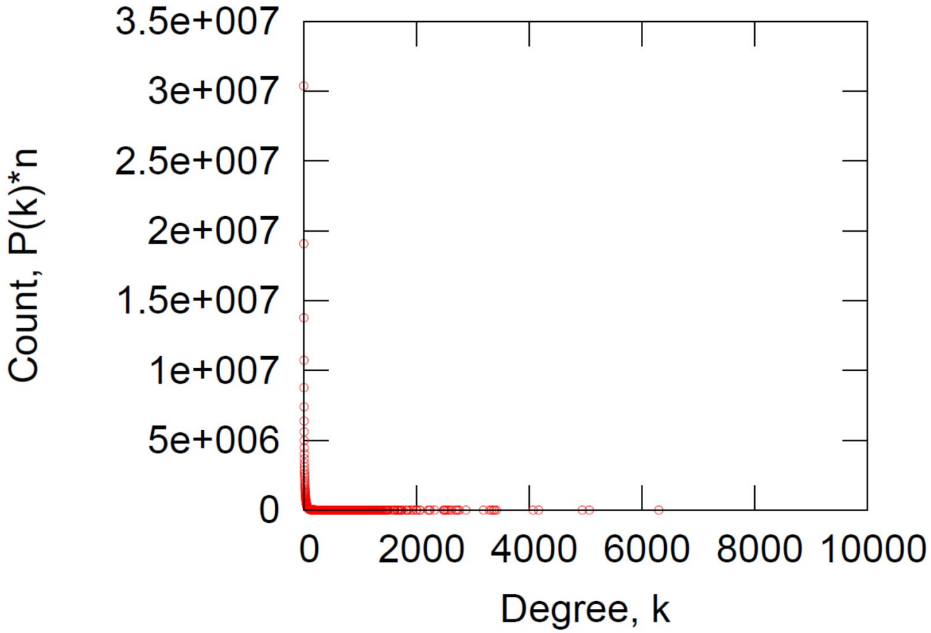
- Normalized histogram:

  $P(k) = N_k / N$ ➔ **plot**



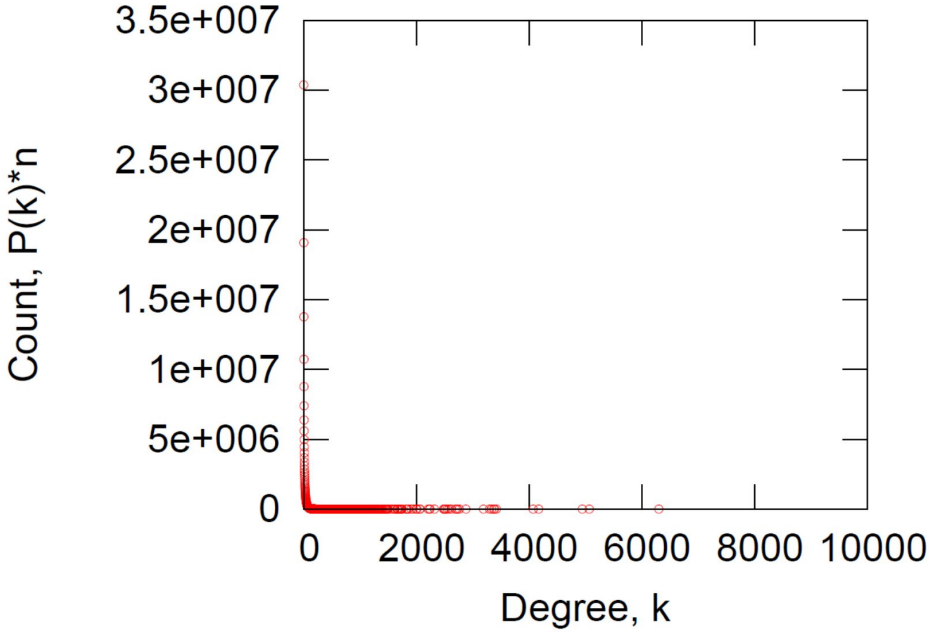


Properties of real-world networks: degree distribution

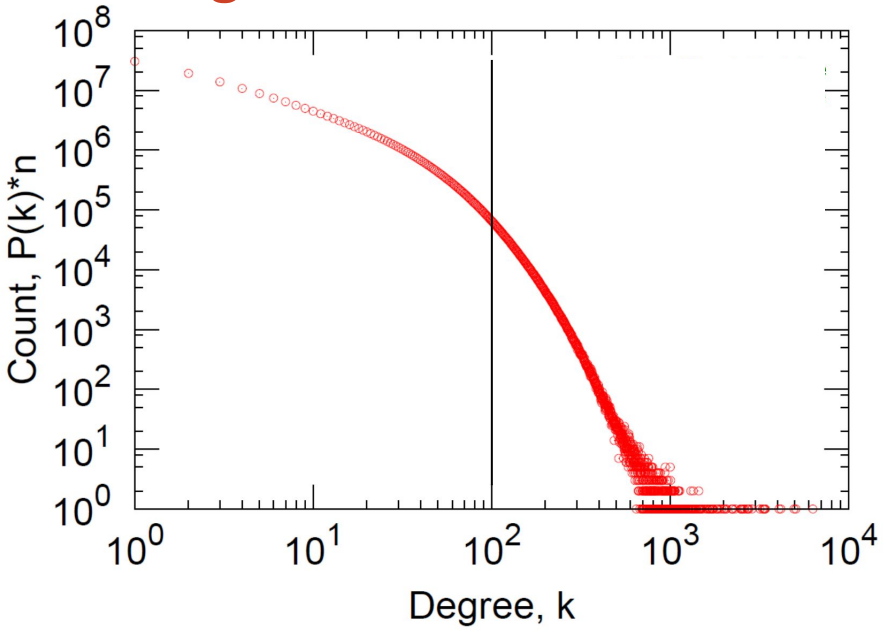


Properties of real-world networks: degree distribution

Linear axes

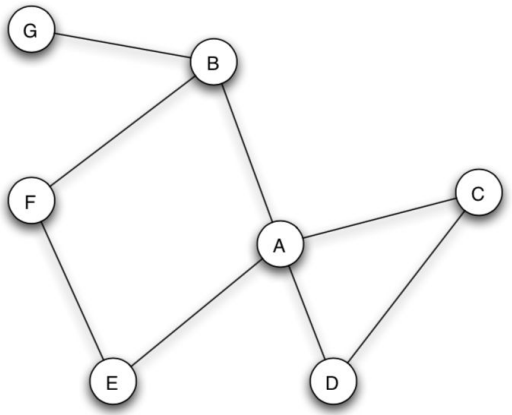


Logarithmic axes



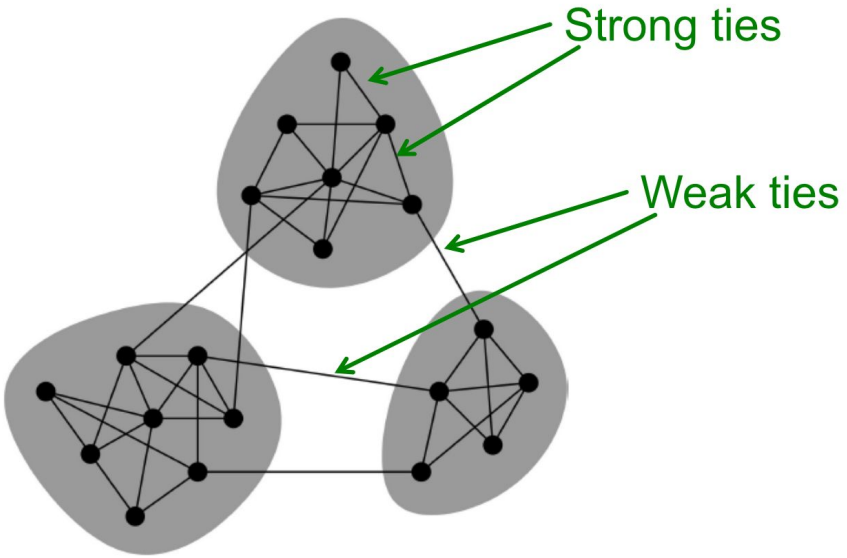
Properties of real-world networks: triadic closure

- "A friend of my friend is my friend"
- Measured via clustering coefficient C_i of node i:
  C_i = (#edges among neighbors of i) / (#potential edges among neighbors of i)
- #potential edges among neighbors of i in undirected graph: $k_i (k_i - 1) / 2$
- C_A = 1 / (4*3/2) = 1/6
- In real networks, nodes have large clustering coefficients



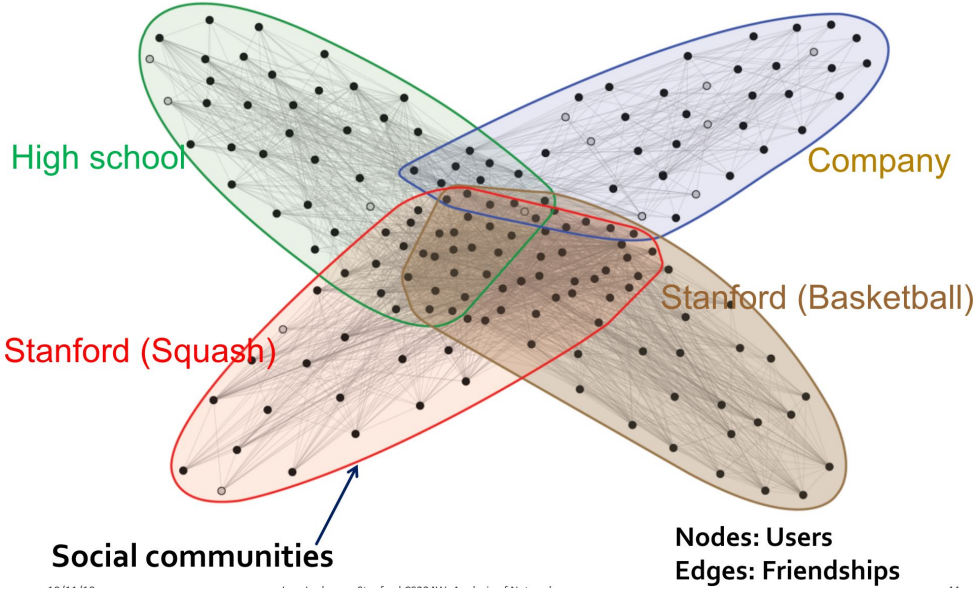
Properties of real-world networks: community structure

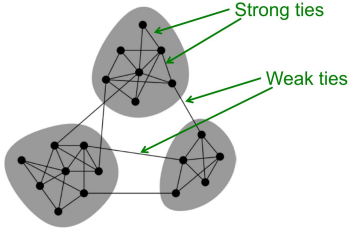
- Triadic closure makes real networks cluster into locally dense "communities"
- Communities connected via "weak ties"
- "The strength of weak ties" (Granovetter 1973)
- Weak ties fill "structural holes"



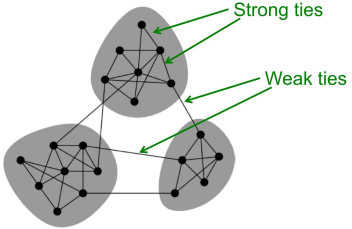
Properties of real-world networks: community structure

- In real life, communities are often not disjoint, but overlapping:





[George Costanza's ideal world; unfortunately not realistic]



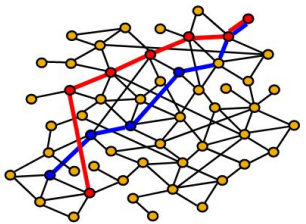
Extra homework: Start watching *Seinfeld,*

so you understand the "Worlds Collide" theory

Properties of real-world networks: average shortest-path length

- **What is the typical shortest path length between any two people?**
  - **Experiment on the global friendship network**
    - Can't measure, need to probe explicitly
- **Small-world experiment** [Milgram '67]
  - Picked 300 people in Omaha, Nebraska and Wichita, Kansas
  - Ask them to get a letter to a stock-broker in Boston by passing it through friends
- **How many steps did it take?**

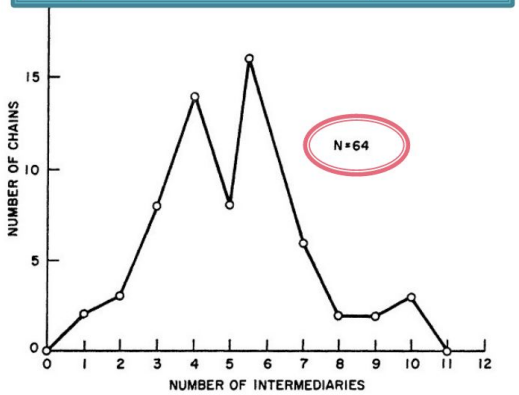
[Movie]



Properties of real-world networks: average shortest-path length

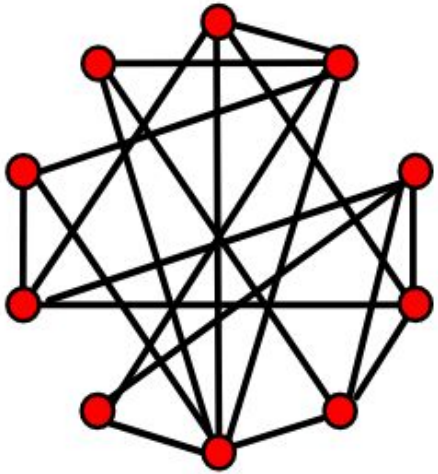
- **64 chains completed:**
  (i.e., 64 letters reached the target)
  - It took 6.2 steps on the average, thus **"6 degrees of separation"**
- **Further observations:**
  - People who owned stock had shorter paths to the stockbroker than random people: 5.4 vs. 6.7
  - People from the Boston area have even closer paths: 4.4

Milgram's small world experiment



Properties of real-world networks: navigability

- For decades, people focused on the fact that short paths exist in social networks
- But this is true even in random graphs (e.g., Erdős-Rényi model)
- The truly amazing fact is not that short paths exist, but that they are discoverable via greedy decentralized routing (as in Milgram's experiment)
- Intrigued? Read Jon Kleinberg: "The Small-World Problem: An Algorithmic Perspective"

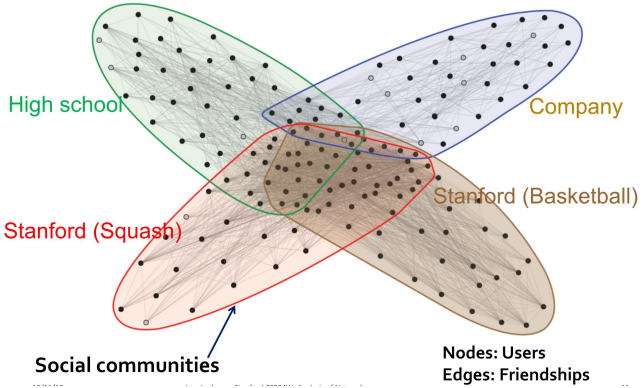


Don’t believe me?

- Play a game and see for yourself how well you can navigate an a-priori unknown network:
  - Wikispeedia.net

Properties of real-world networks: homophily/heterophily

- "Birds of a feather flock together"
- Especially in social networks
- Big confound and cause of debate: Influence vs. homophily
- E.g., obese people's friends are more likely to also be obese
  - Influence: I copy eating behavior of those around me [argument for influence]
  - Homophily: people with similar eating behavior prone to become friends [argument for homophily]



Properties of real-world networks: summary

Real-world networks (across many types)

- are sparsely connected,
- but some nodes are much more connected than most others (i.e., skewed degree distribution);
- form locally dense clusters via triadic closure,
- which leads to community structure;
- have short paths between random node pairs (partly due to “hubs” [skewed degree distribution!]),
- and the short paths are easily discoverable.

Part 4:
Measuring node importance

How to measure “importance” of a node?

- Formalized via *centrality measures*
- Map each node *i* to a scalar value *C*(*i*) capturing its importance in the overall network

Degree centrality

- Simplest centrality measure
- Many neighbors → important node
  - $C(i)$ = number of neighbors of i
- Very brittle, easy to “rig”
  - E.g., Twitter: scam account, followed by 100,000 other scam accounts

Closeness centrality

- Farness(x) = total distance to x from other nodes
- $C(x)$ = 1 / Farness(x)
  - Reciprocal to turn farness into closeness
  - Only defined for connected graphs (otherwise d infinite)

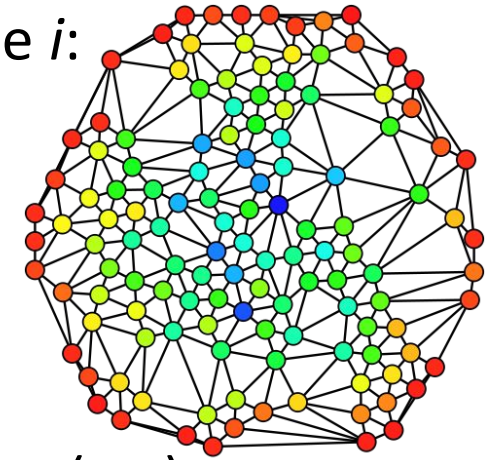
$$C(x) = \frac{1}{\sum_y d(y,x)}$$

- Under closeness centrality, nodes that are easy to reach from anywhere in the network are considered important
- Variant: harmonic centrality: switch sum and reciprocal
  - $C(x)$ = total reciprocal distance of x to other nodes
  - Defined even for disconnected graphs (define $1/d$ of disconnected nodes as 0)

$$C(x) = \sum_{y \neq x} \frac{1}{d(y,x)}$$

Betweenness centrality

- $C(i)$ = average fraction of all shortest paths in the network that pass through node i
- Computation of betweenness centrality of node i:
  - For each pair of vertices (s, t):
    - Find all shortest paths from s to t
    - Compute the fraction of these shortest paths that pass through i
  - Average this fraction over all pairs of vertices (s, t)
- Expensive to compute



Katz centrality

- Generalization of degree centrality
- Degree centrality counts only number of direct neighbors (i.e., neighbors at distance 1)
- Katz centrality also counts neighbors at distances 2, 3, …
- More precisely, number of paths from other nodes to *i*, giving less weight to larger distances:

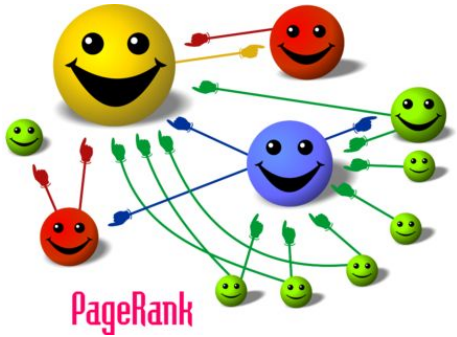
$$C(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{N} \alpha^k (A^k)_{ji}$$

k-th power of adjacency matrix *A* contains number of length-*k* paths for each node pair

- More robust than degree centrality

PageRank centrality

- Recursive definition: my centrality $C(i)$ =: x_i is high if I receive inlinks from many other central nodes:



$$x_i = \sum_j a_{ji} \frac{x_j}{L(j)}$$

$$L(j) = \sum_i a_{ji}$$

a_{ji}: entry (j, i) of adjacency matrix A (1 if j links to i, else 0)
$L(j)$: out-degree of j

- Some extra tweaks to make it work with any graph (e.g., disconnected)

PageRank centrality

$$x_i = \sum_j a_{ji} \frac{x_j}{L(j)}$$

- Matrix notation: $x = M x$
  (where *M* is computed from adjacency matrix *A*)
- Do you recognize this?
  - *x* is eigenvector of *M* with eigenvalue 1 → we're in linear-algebra land (home sweet home)
  - *x* is the steady state the Markov chain induced by the network: x_i is fraction of time a random walker will have spent in node *i*, after a very long (→∞) random walk

PageRank centrality

- The technology that made Google huge
  - "Page" in PageRank for Larry Page
  - MapReduce (next lecture!) was invented to compute PageRank on full Google Web crawl
  - "The $25,000,000,000 eigenvector" [link]
- Bottom line:
  Pay attention in your linear algebra class

Give us feedback on this lecture here:
https://go.epfl.ch/ada2023-lec12-feedback

- What did you (not) like about this lecture?
- What was (not) well explained?
- On what would you like more (fewer) details?
- …

Credits

- Some slides borrowed from CS224W class (Jure Leskovec, Stanford)