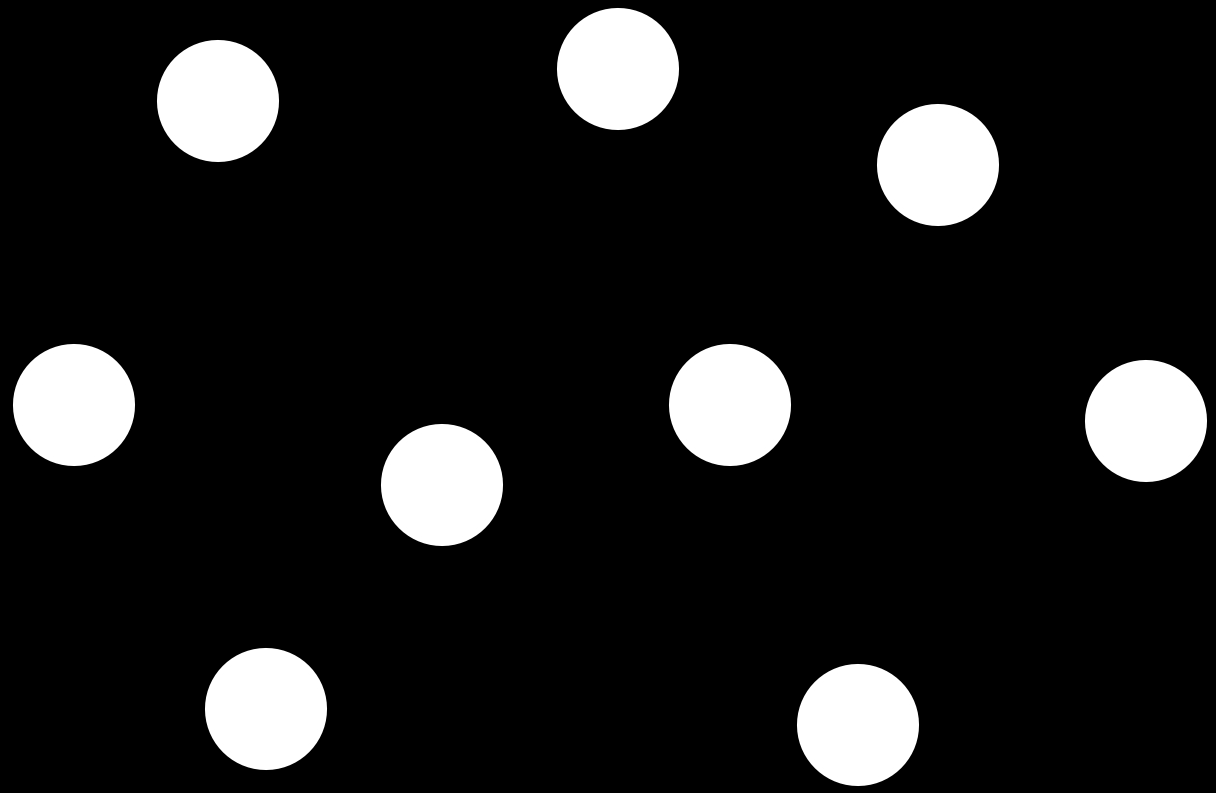
CS224W: Analysis of Networks

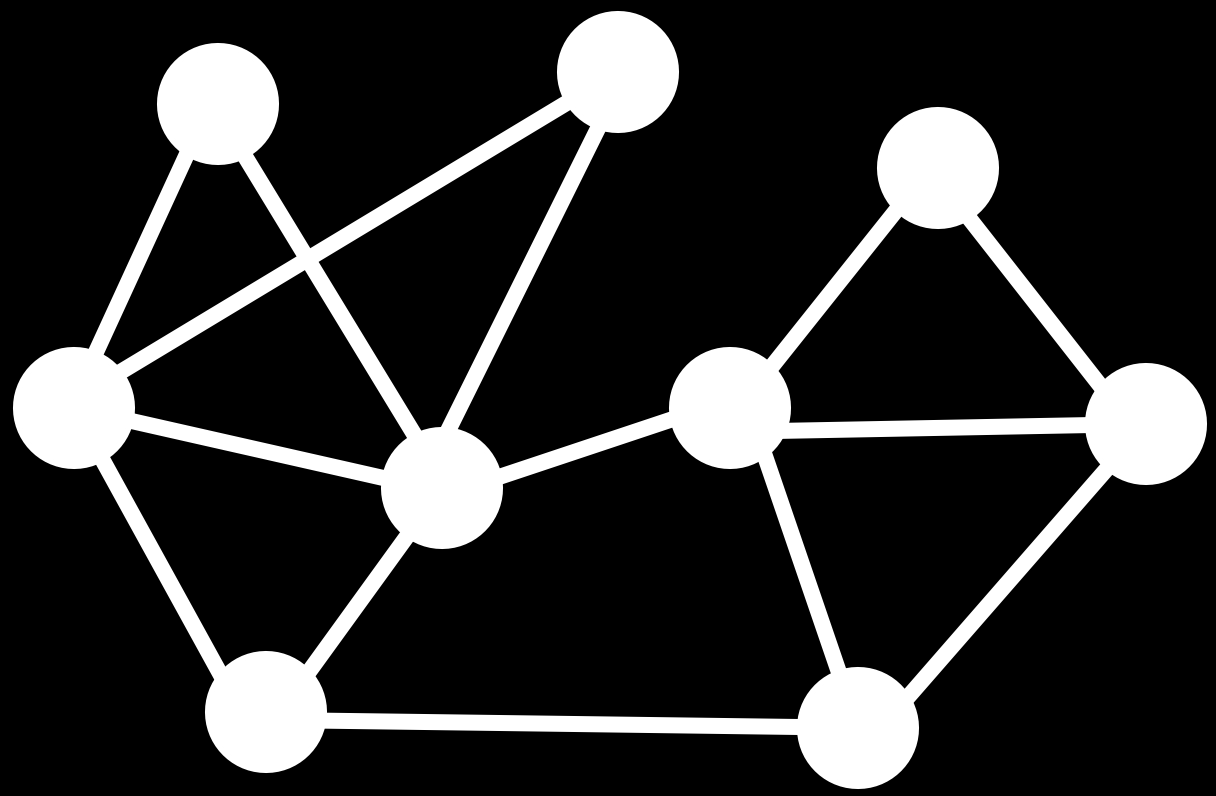
Mining and Learning with Graphs

CS224W: Analysis of Networks
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Why Networks?

Networks are a general language for describing complex systems



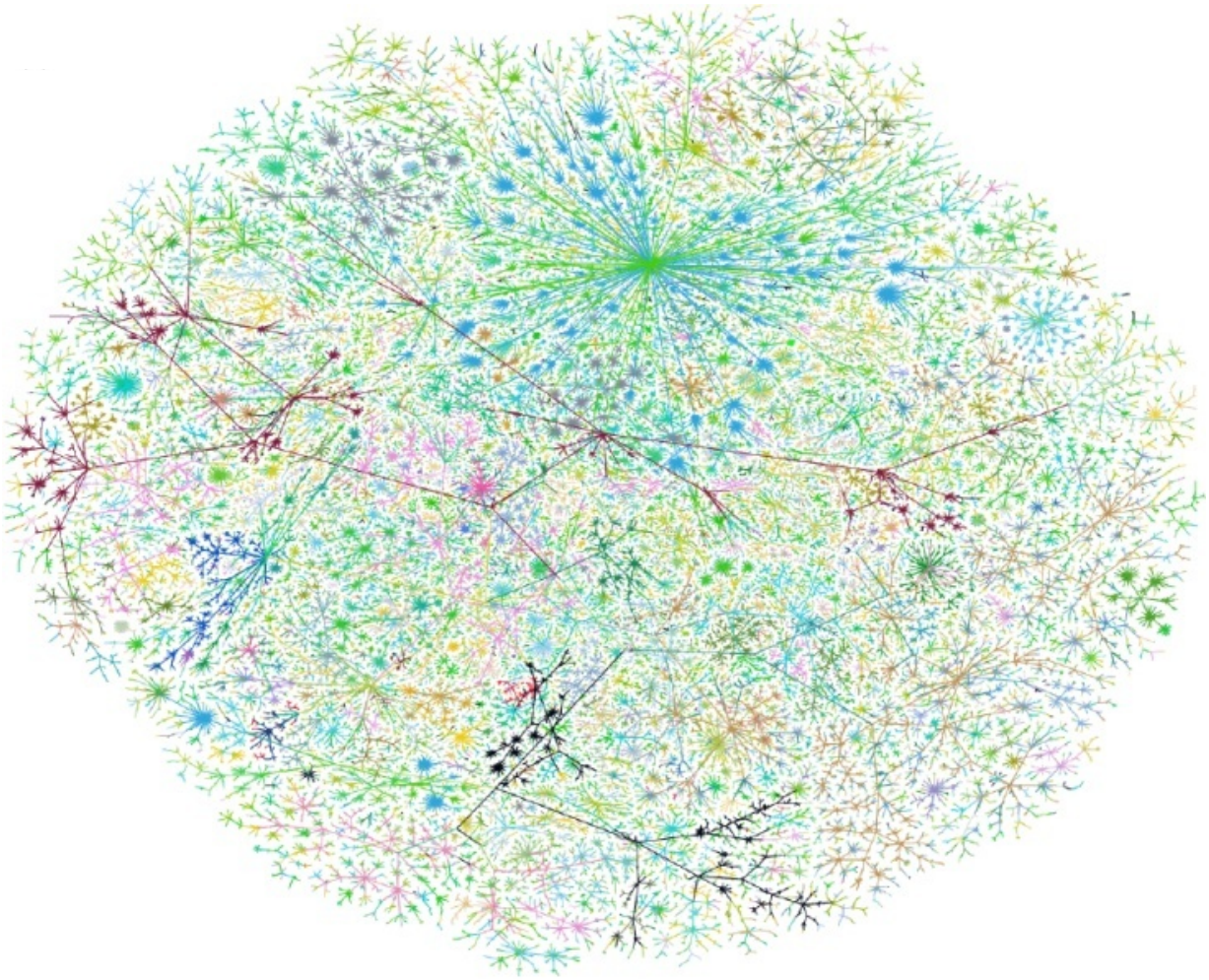


Network

Networks & Complex Systems

- **There are complex systems all around us:**
  - **Society** is a collection of 7+ billion individuals
  - **Communication systems** link electronic devices
  - **Information** and **knowledge** are organized and linked
  - Interactions between thousands of **genes/proteins** regulate life
  - Our **thoughts** are hidden in the connections between billions of neurons in our brain

What do these systems have in common?
How can we represent them?



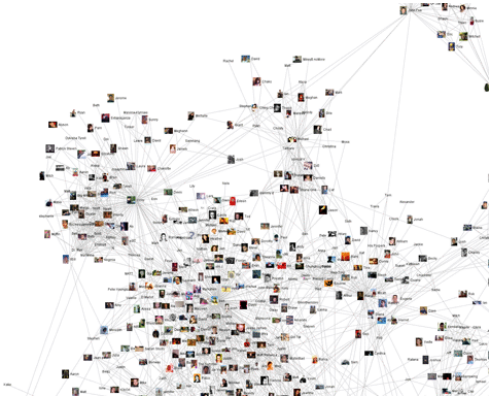
The Network!

Networks!!

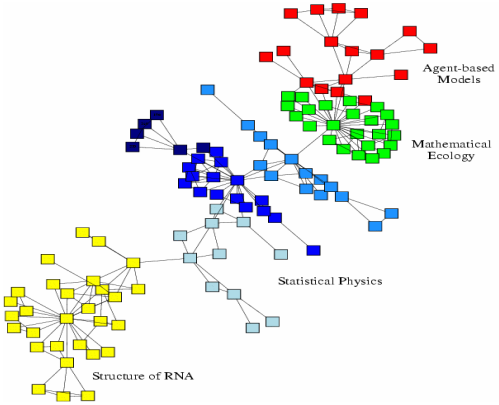
Behind many systems there is an intricate wiring diagram, **a network**, that defines the **interactions** between the components

We will never be able to model and predict these systems unless we understand the networks behind them!

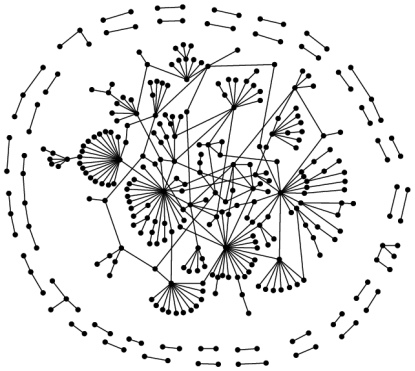
Many Types of Data are Networks



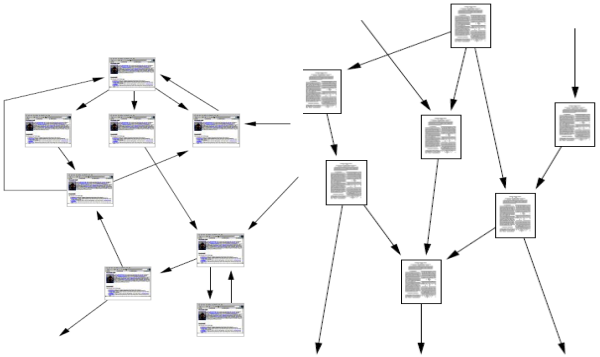
Social networks



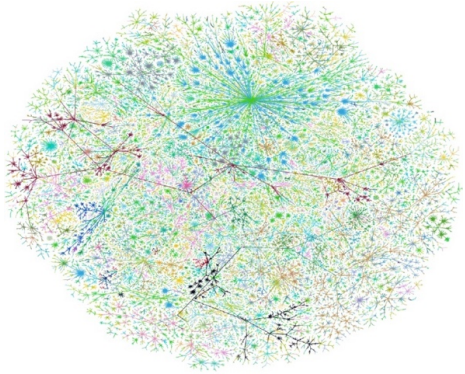
Economic networks



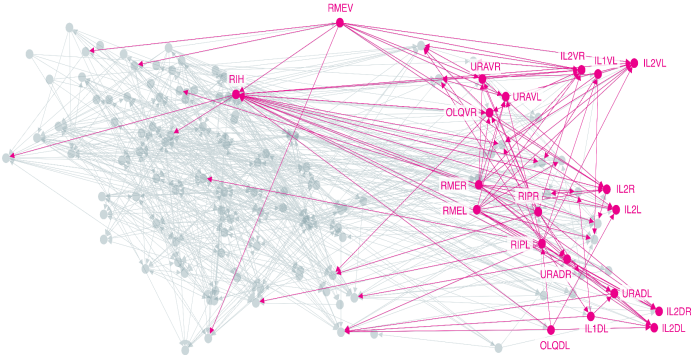
Communication graphs



Information networks:
Web & citations

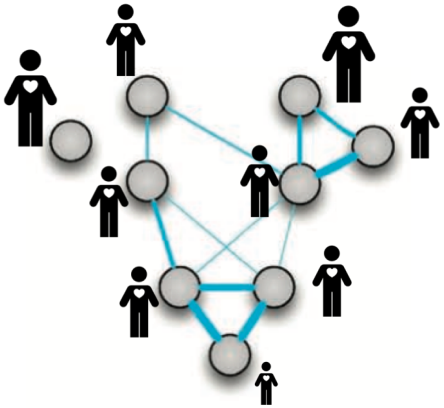


Internet

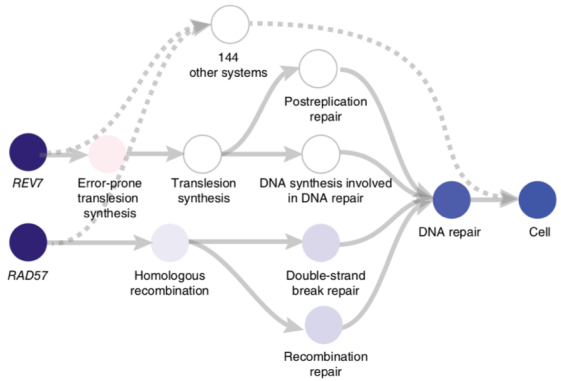


Networks of neurons

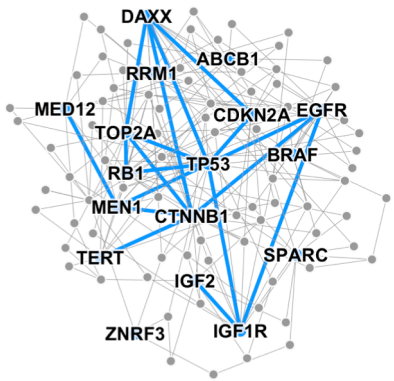
Many Types of Data are Networks



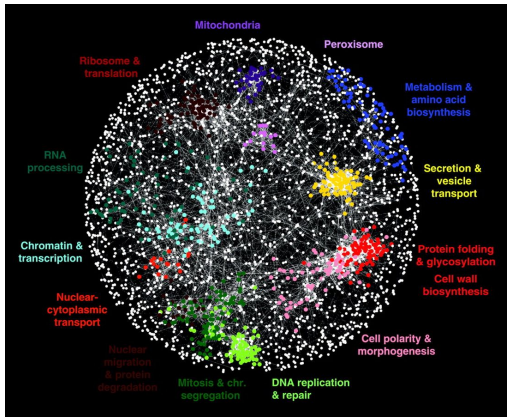
Patient networks



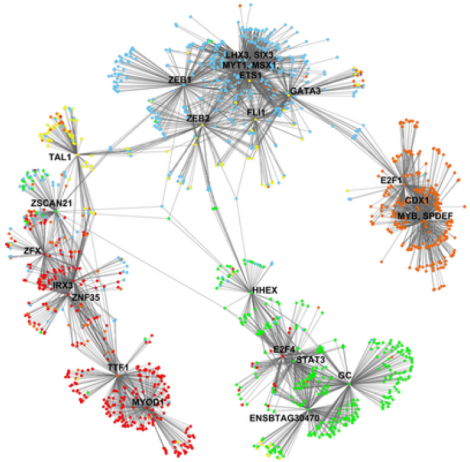
Hierarchies of cell systems



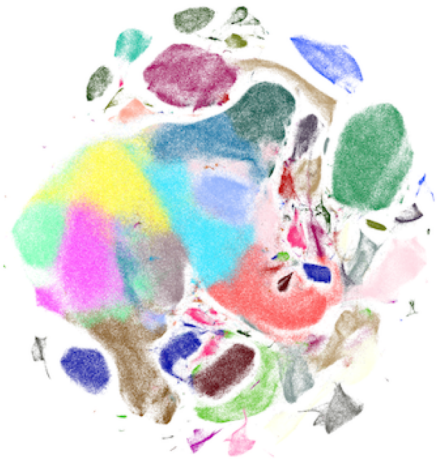
Disease pathways



Genetic interaction networks



Gene co-expression networks



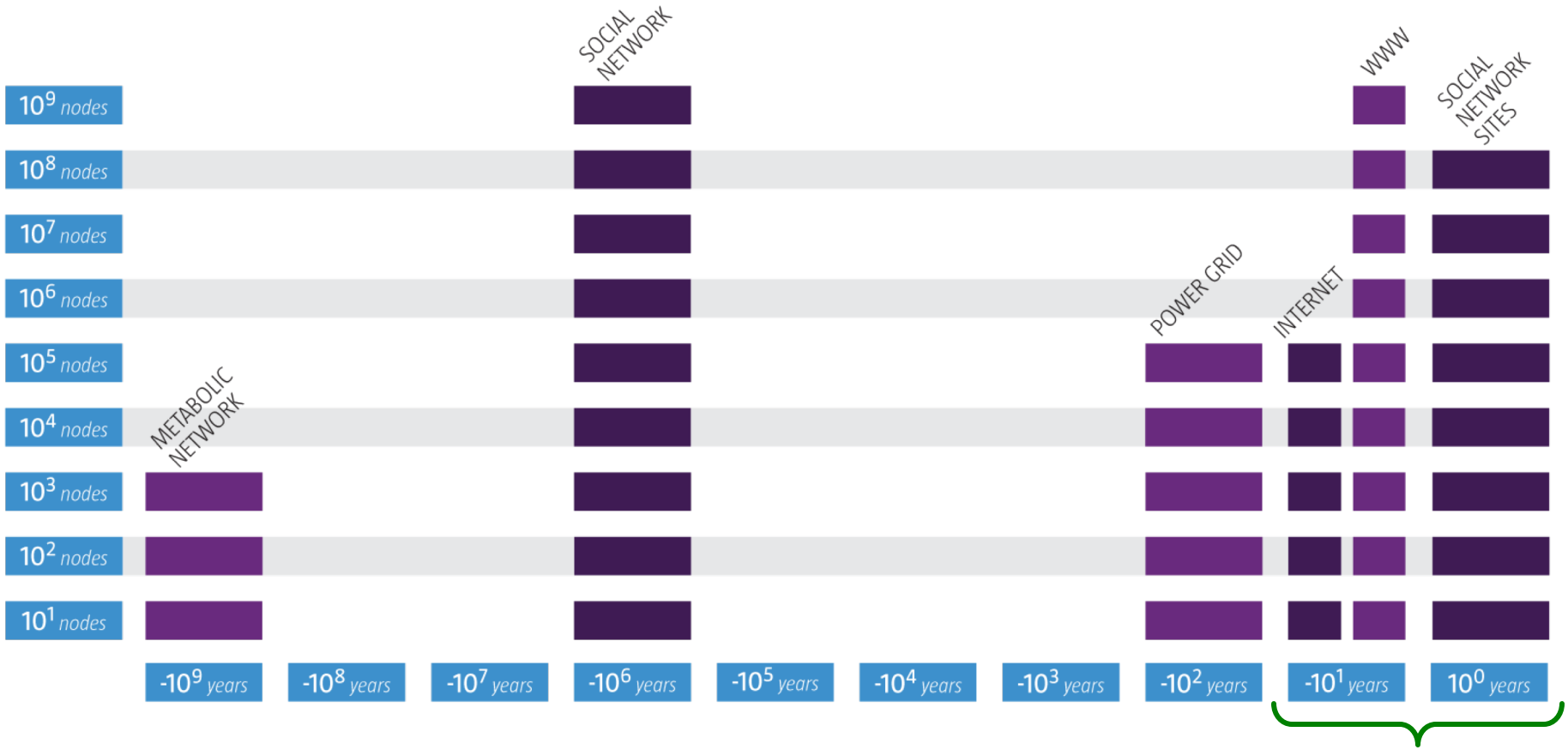
Cell-cell similarity networks

But Jure, why should I care about networks?

Why Networks? Why Now?

- **Universal language for describing complex data**
  - Networks from science, nature, and technology are more similar than one would expect
- **Shared vocabulary between fields**
  - Computer Science, Social Science, Physics, Economics, Statistics, Biology
- **Data availability & computational challenges**
  - Web/mobile, bio, health, and medical
- **Impact!**
  - Social networking, Social media, Drug design

Networks: Why Now?



Age and size of networks

CS!!

Networks: Impact

- Google
- Cisco
- Facebook
- Amazon

Networks and Applications

Ways to Analyze Networks

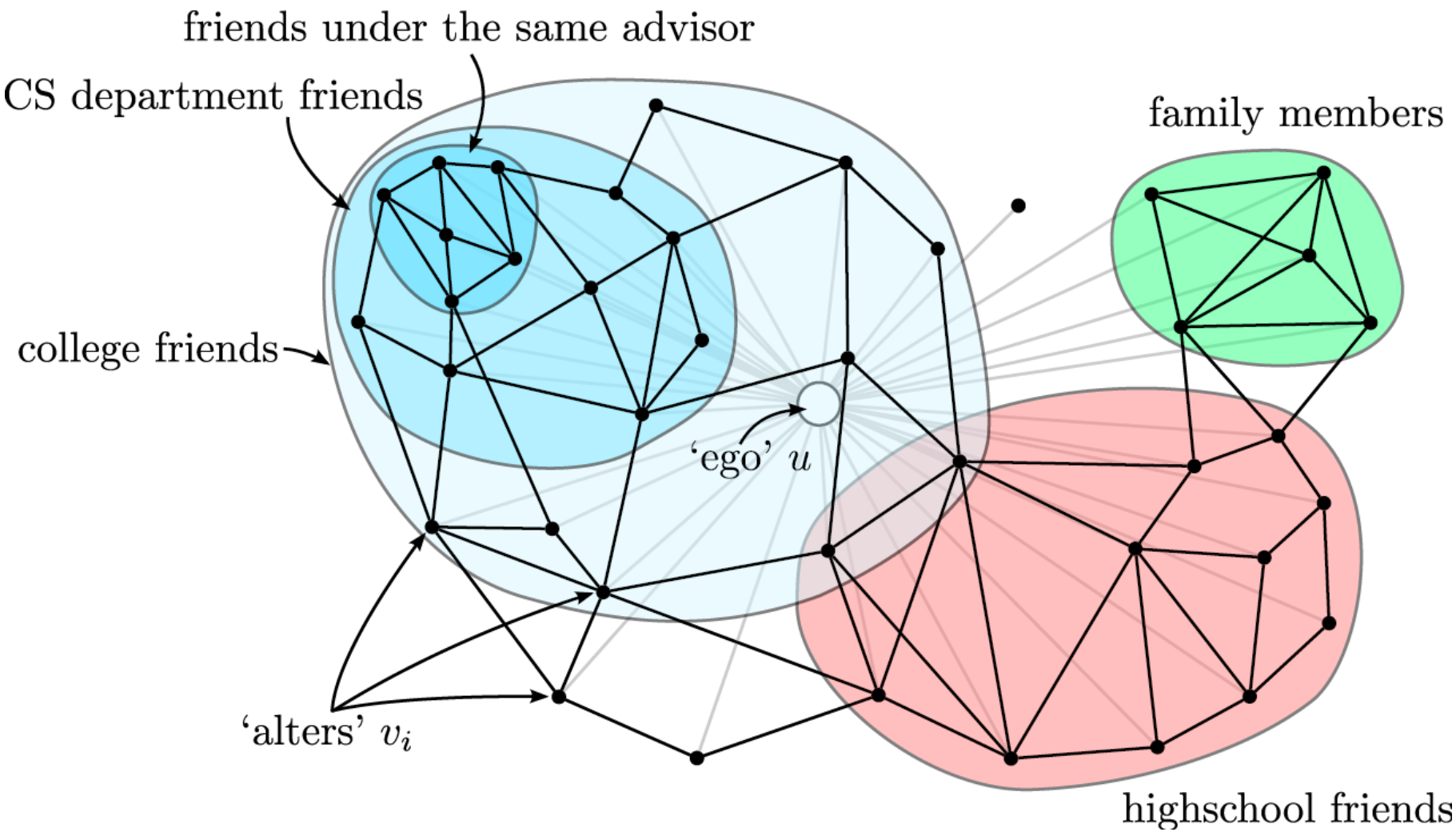
- **Predict the type/color of a given node**
  - Node classification
- **Predict whether two nodes are linked**
  - Link prediction
- **Identify densely linked clusters of nodes**
  - Community detection
- **Measure similarity of two nodes/networks**
  - Network similarity

(1) Networks: Social

Facebook social graph

4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

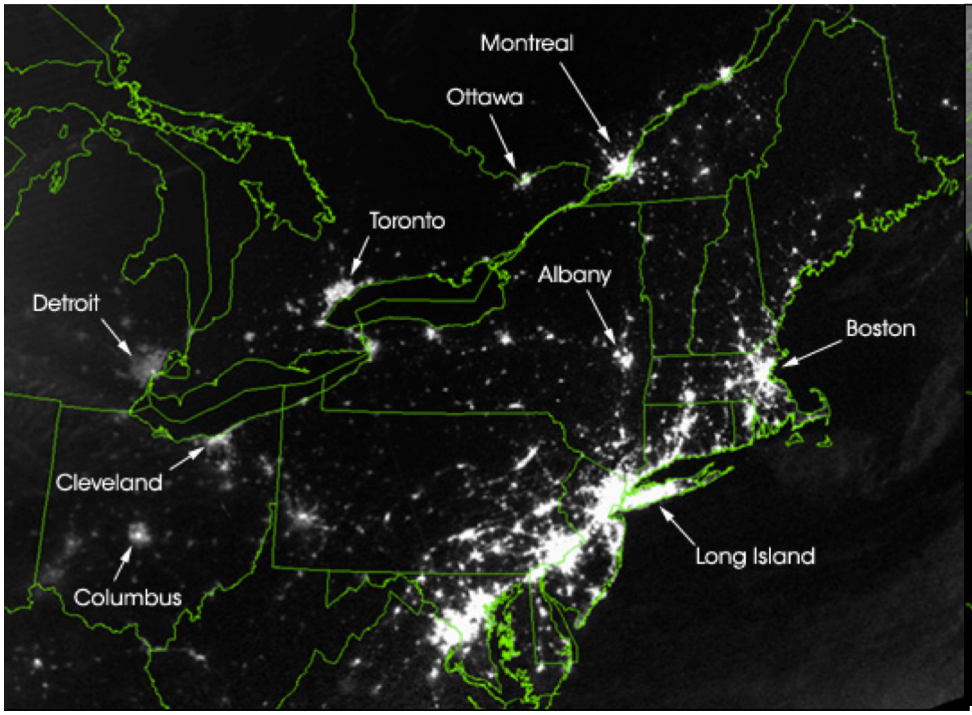
Application: Social Circle Detection



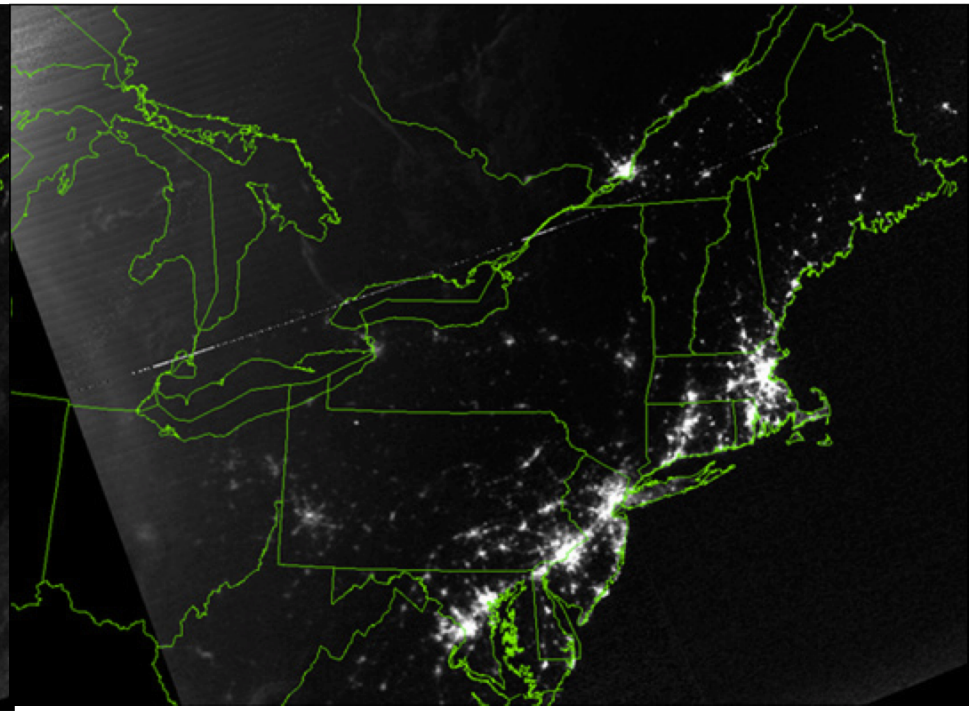
Discover circles and why they exist

(2) Networks: Infrastructure

- **August 15, 2003 blackout**



August 14, 2003: 9:29pm EDT
20 hours before



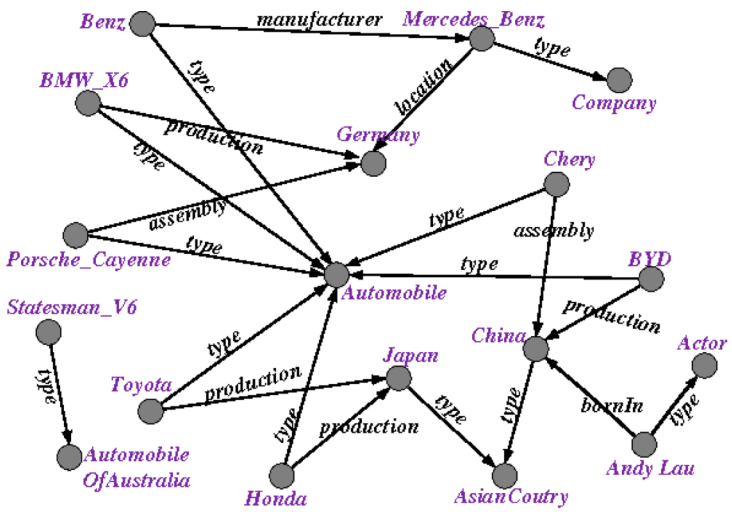
August 15, 2003: 9:14pm EDT
7 hours after

August 15, 2003 blackout

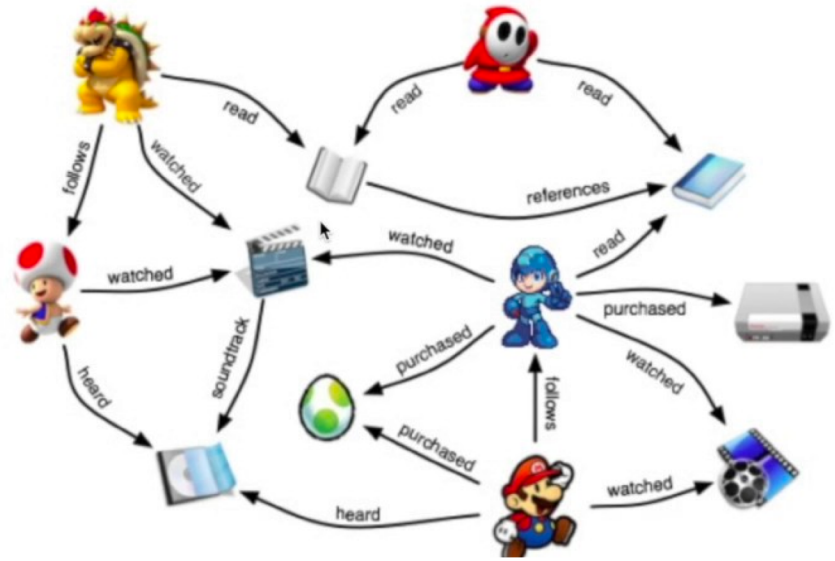
This reveals two important themes of this class:

- We must understand how network structure affects the robustness of a system
- Develop quantitative tools to assess the interplay between network structure and the dynamical processes on the networks, and their impact on failures
- We will learn that in reality failures follow reproducible laws, that can be quantified and even predicted using the tools of networks

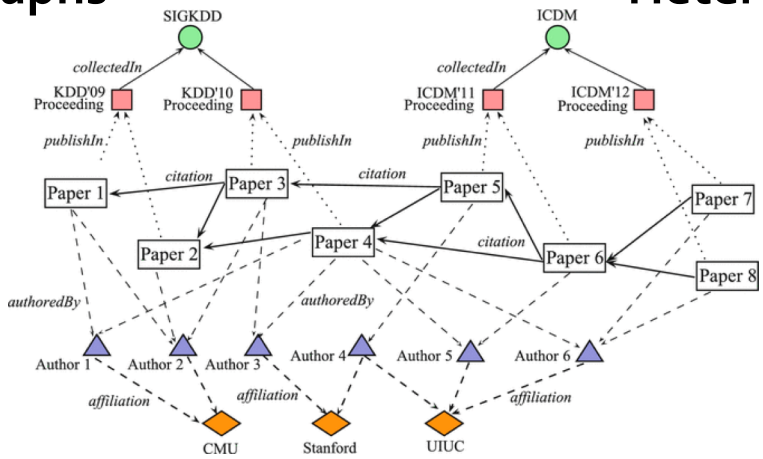
(3) Networks: Information, Knowledge



Knowledge Graphs



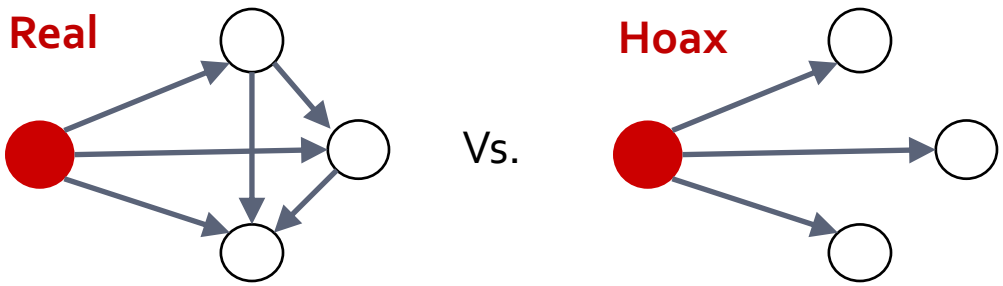
Heterogeneous Graphs

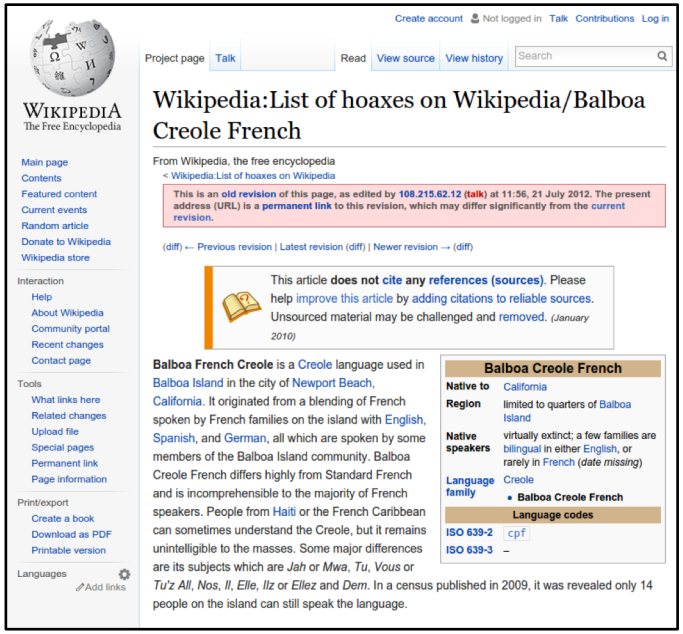


Multimodal Graphs

Application: Misinformation

- **Q: Is a given Wikipedia article a hoax?**
  - Real articles link more coherently:





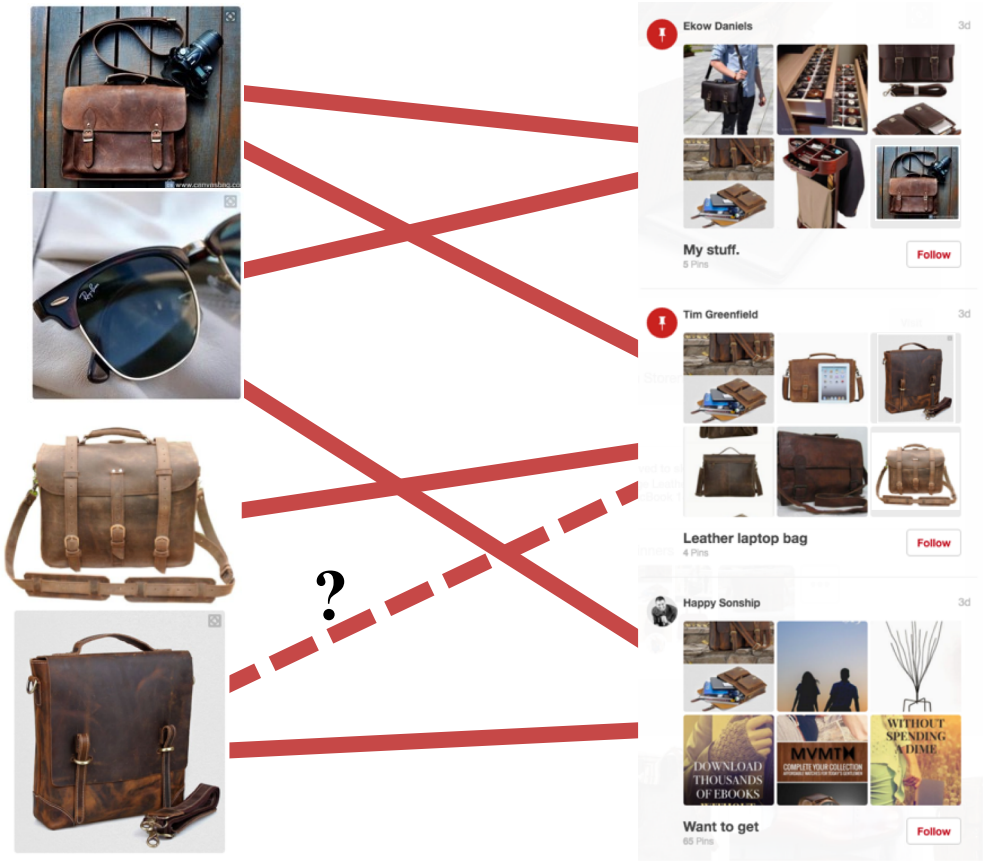
Hoax article detection performance:

50%	66%	86%
Random	Human	Network

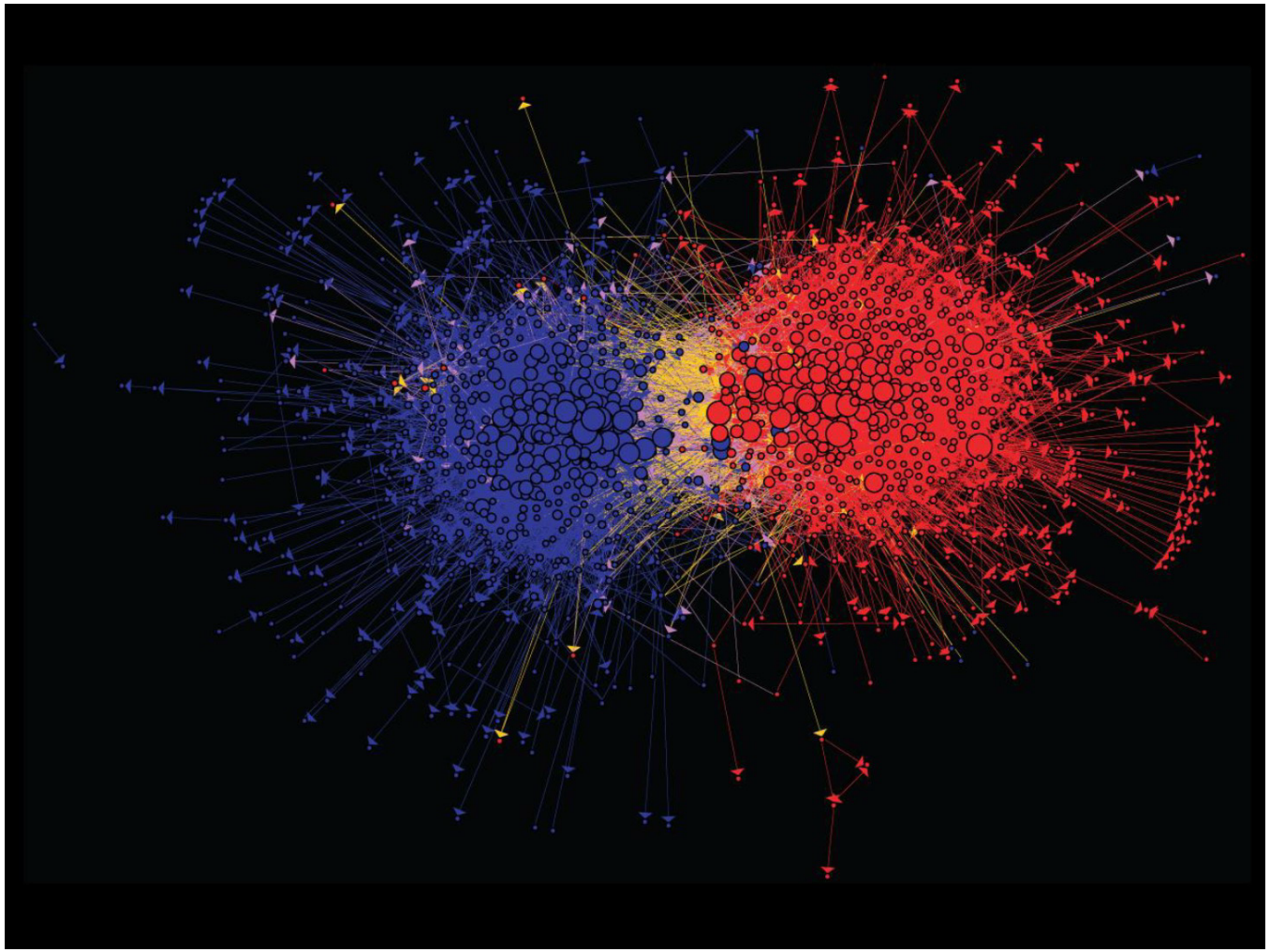
[Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes](). Kumar et al. WWW '16.

Application: Link Prediction

Content recommendation is link prediction



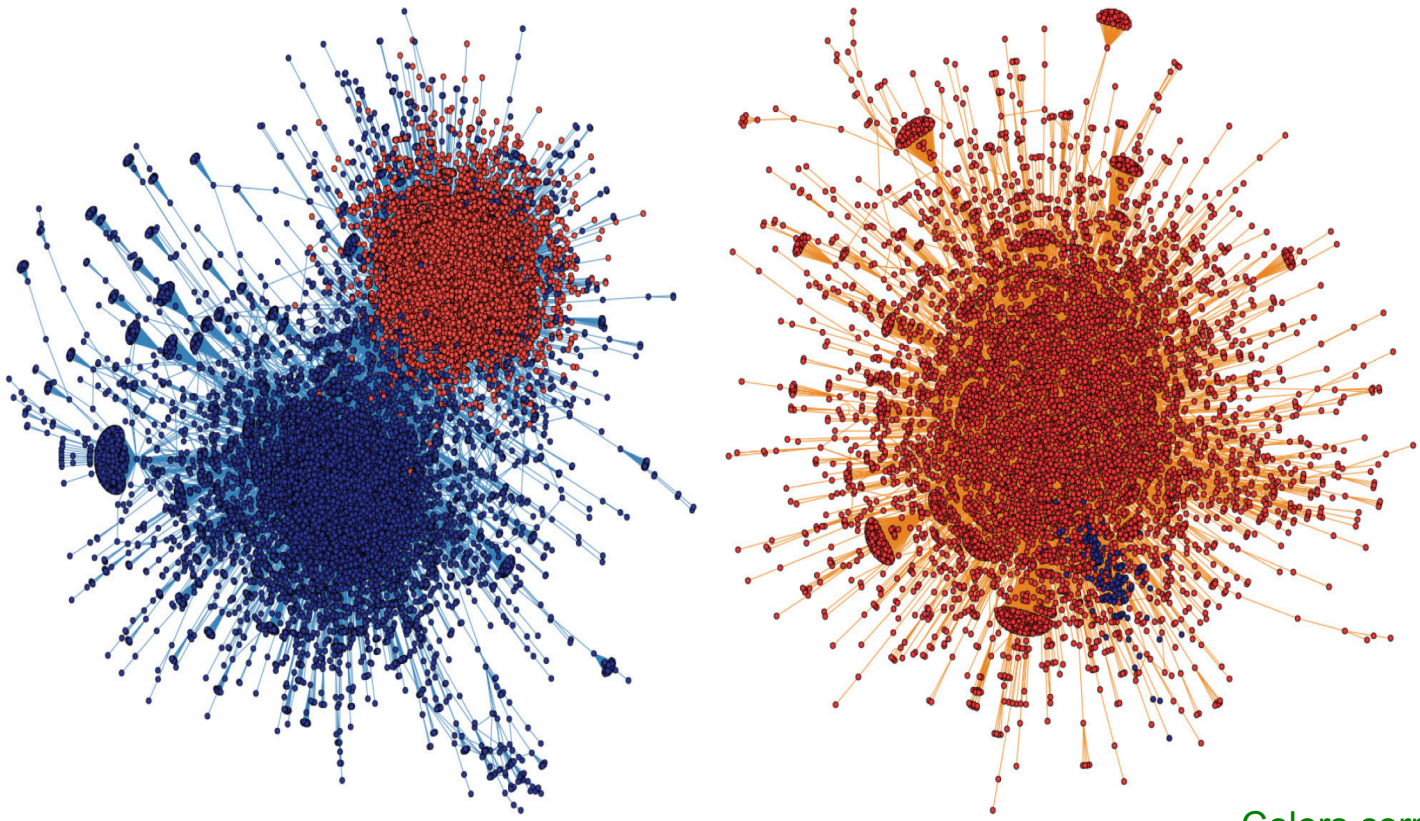
(4) Networks: Online Media



Connections between political blogs

Polarization of the network [Adamic-Glance, 2005]

Application: Polarization on Twitter

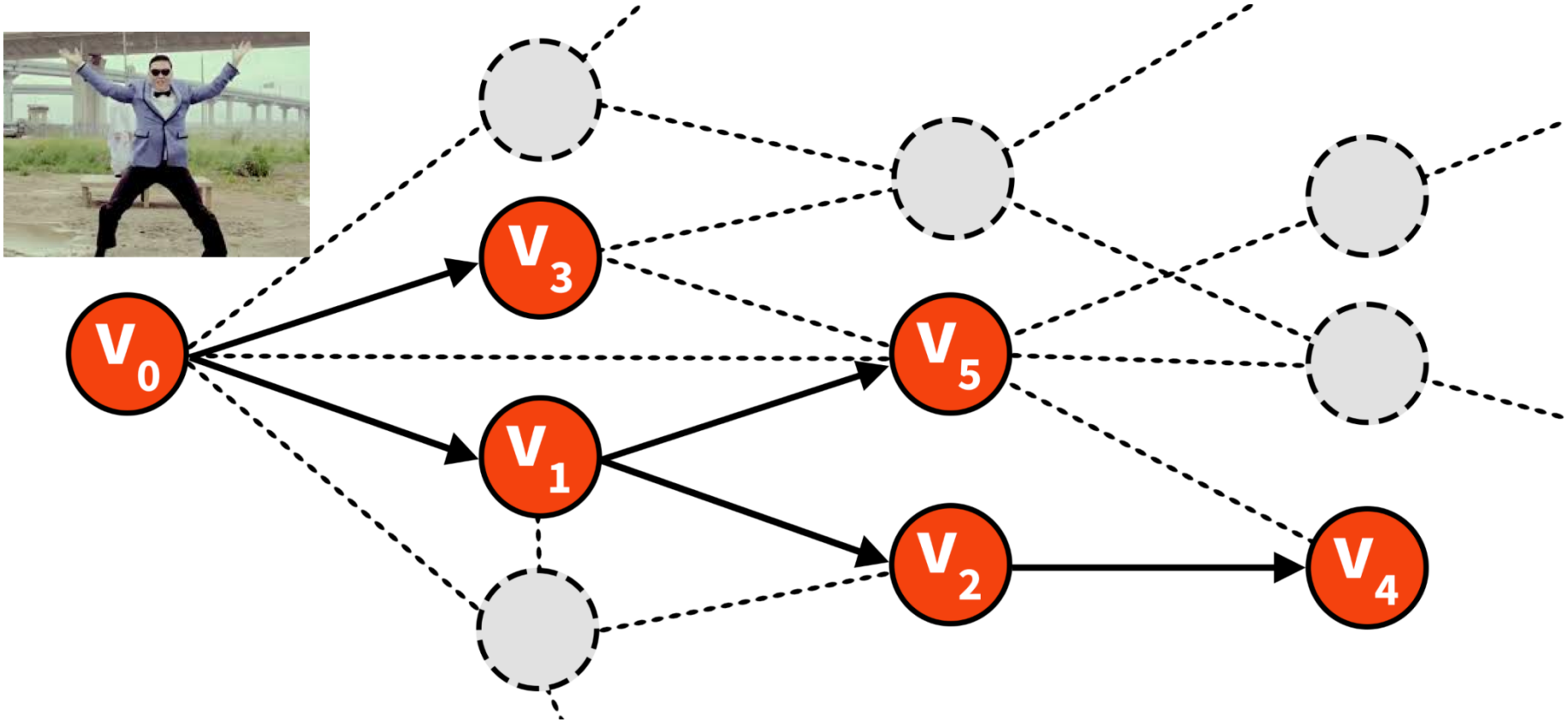


Colors correspond to clusters in the network

- **Retweet networks:**
  Polarized (left), Unpolarized (right)

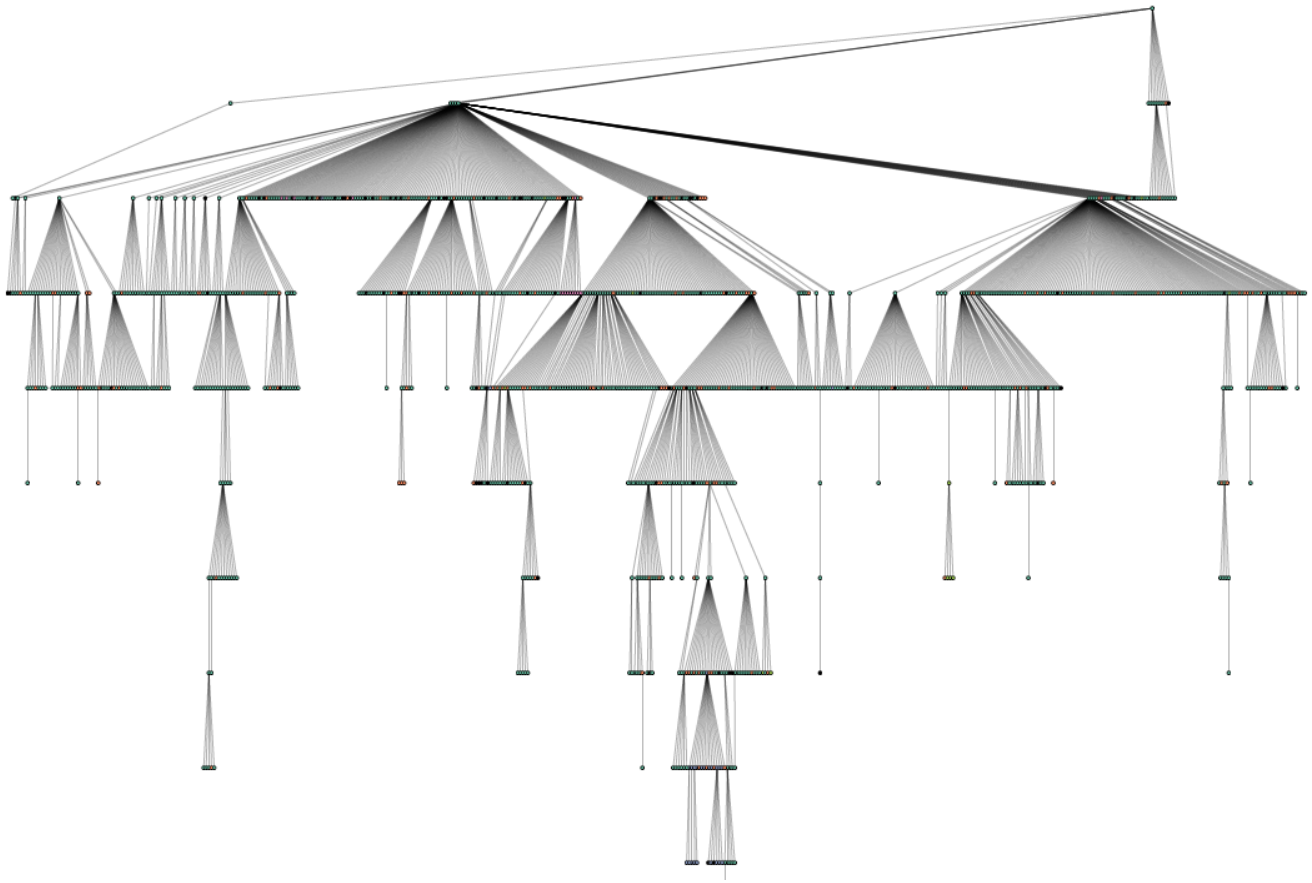
Conover, M., Ratkiewicz, J., Francisco, M. R., Gonçalves, B., Menczer, F., & Flammini, A. "Political Polarization on Twitter." (2011)

Application: Predicting Virality



Information cascade in social networks

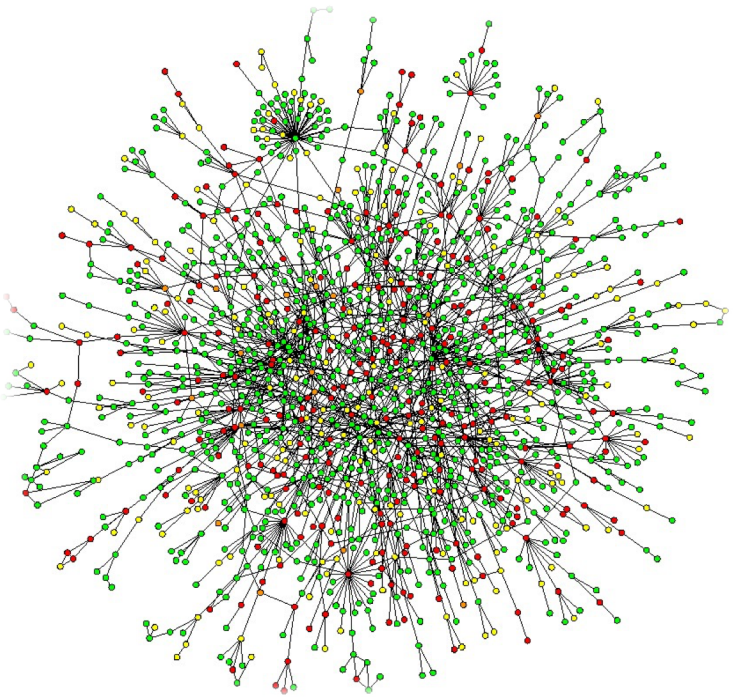
Application: Product Adoption

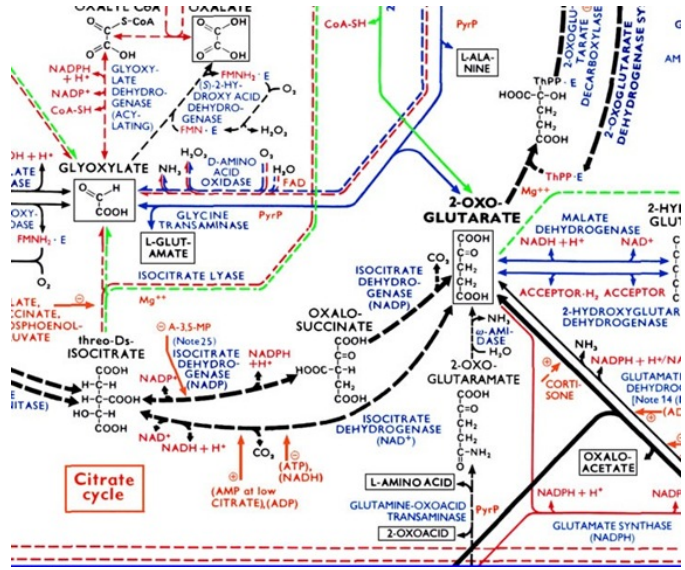


Invitation cascades: 60-90% of LinkedIn users signed up due to an invitation from another user.

Global Diffusion via Cascading Invitations: Structure, Growth, and Homophily. Anderson et al., WWW '15.

(5) Networks: Biology





Protein-protein interaction (PPI) networks:

Nodes: Proteins

Edges: 'Physical' interactions

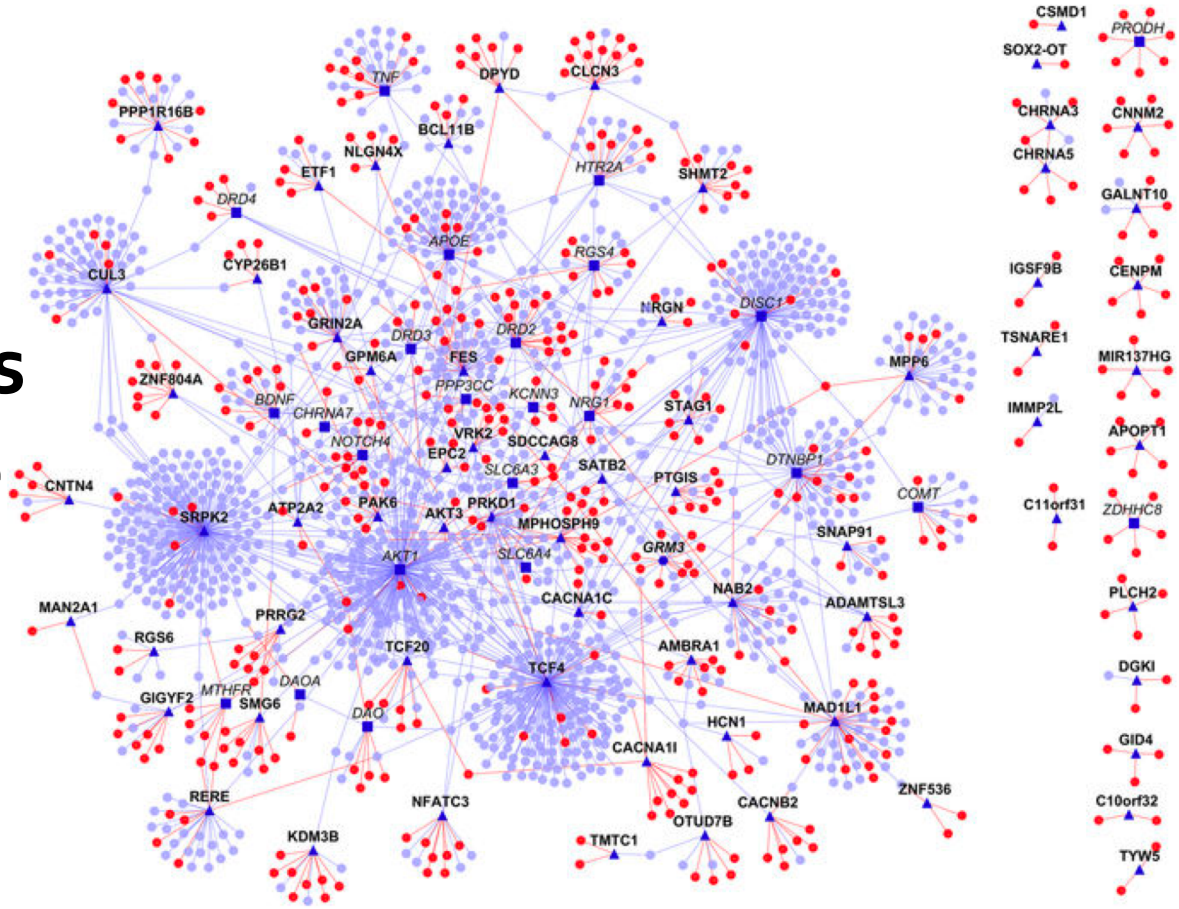
Metabolic networks:

Nodes: Metabolites and enzymes

Edges: Chemical reactions

Application: Function Prediction

Classifying the function of proteins in the interactome



Ganapathiraju et al. 2016. Schizophrenia interactome with 504 novel protein–protein interactions. *Nature*.

About CS224W

Reasoning about Networks

- **What do we hope to achieve from studying networks?**
  - Patterns and statistical **properties** of network data
  - **Design principles** and **models**
  - **Algorithms** and **predictive models** to answer questions and make predictions
    - Predict behavior of networked systems

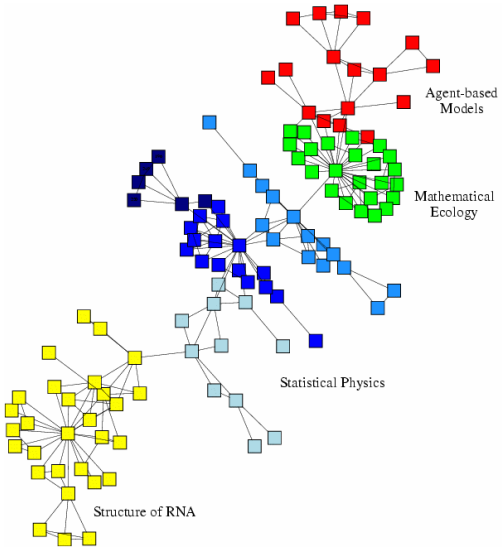
Mining and Learning with Graphs

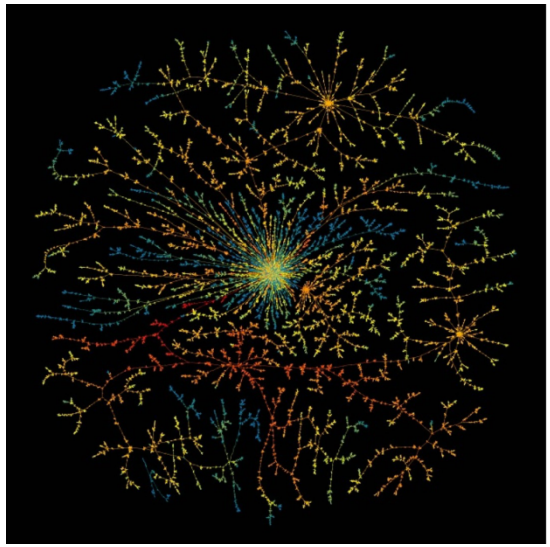
- **How do we mine networks?**
  - **Empirically:** Study network data to find organizational principles
    - How do we measure and quantify networks?
  - **w/ Mathematical models:** Graph theory and statistical models
    - Models allow us to understand behaviors and distinguish surprising from expected phenomena
  - **w/ Algorithms** for analyzing graphs
    - Hard computational challenges

Networks: Structure & Process

What do we study in networks?

- **Structure and evolution:**
  - What is the structure of a network?
  - Why and how did it come to have such structure?
- **Processes and dynamics:**
  - Networks provide a "skeleton" for spreading of information, behavior, diseases
  - How do information and diseases spread?





(Tentative) Course Outline

	lectures
Graph concepts, models and measurements	(1-3)
Network Construction and Inference	(4)
Link Analysis: PageRank and SimRank	(5)
Network Motifs: Structural roles in Networks	(6)
Community Detection	(7, 8)
Link Prediction	(9)
Node Representation Learning: Node2Vec	(10)
Information Cascades	(11, 12)
Influence & Outbreak Detection	(13, 14)
Network Robustness	(15)
Network Evolution	(16)
Node Centrality and Anomaly Detection	(17)
Knowledge Graphs and Metapaths	(18)
Message passing and Node classification	(19)
Graph Convolutional Neural Networks	(20)

Logistics: Course Assistants

Michele Catasta
Co-instructor and head TA

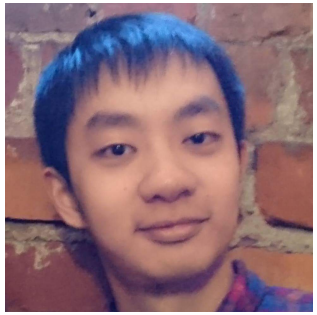
Alex
Haigh

Alexander
Wang

Javier
Sagastuy
Brena

Jayadev
Bhaskaran

Megha
Jhunjhunwala



Shuyang
Shi

Logistics: Website

- http://cs224w.stanford.edu
  - Slides posted the night before the class
- **Readings:**
  - Mostly research papers
- **Optional readings:**
  - Papers and pointers to additional literature
  - **This will be very useful for project proposals**

Logistics: Communication

- **Piazza Q&A website:**
  - http://piazza.com/stanford/fall2018/cs224w
    - Use access code "snap"
  - **Please participate and help each other!**
    - Don't post code, annotate your questions, search for answers before you ask
- **For e-mailing course staff, always use:**
  - cs224w-aut1819-staff@lists.stanford.edu
- We will post course announcements to Piazza (make sure you check it regularly)

Work for the Course & Grading

- **Final grade will be composed of:**
  - **Homework: 30%**
    - Homeworks 1, 2, 3, each worth 9.6%, HW0 worth 1%
  - **Exam: 30%**
  - **Course project: 40%**
    - Proposal: 20%
    - Project milestone: 20%
    - Final report: 50%
    - Poster presentation: 10%
  - **Extra credit: Piazza participation, Snap code contribution**
    - Used if you are on the boundary between grades

Homework, Write-ups

- **Assignments are long and take time (10-20h) Start early!**
  - A combination of data analysis, algorithm design, and math
- **How to submit?**
  - **Upload via Gradescope (http://gradescope.com)**
    - **To register use the code 9ZZ2XY**
      - **Use your Stanford email (if non-SCPD) and include your Stanford ID # (everyone)**
    - **Each answer must start on a new page**
  - **Code and project write-ups** (proposal, milestone, final report) have to **also** be uploaded at http://snap.stanford.edu/submit/
- **Total of 2 late periods for the quarter:**
  - Late period expires on Monday at 23:59 Pacific Time
  - Max 1 late period per assignment (but not for final report)

Exam

- **November 29th, 2018 (in the evening)**
- **Duration: 2 hours**
- **Covers the content up to and including November 27th**
- **Open book/notes**
  - **Exercises where you will have to explain the solution process**
- **We will strictly enforce the Stanford Honor Code**
  - No cheating!

Course Projects

- **Course project:**
  - **Empirical analysis** of network data to develop a model of behavior
  - **Algorithms and models** to make predictions on a network dataset
  - **Scalable algorithms** for massive graphs
  - **Theoretical project** that considers a model/algorithm and derives a rigorous result about it
- **Performed in groups of up to 3 students**
  - Fine to have groups of 1 or 2. The team size will be taken under consideration when evaluating the scope of the project in breadth and depth. But 3 person teams can be more efficient.
  - Project is the **important work** for the class
  - We will help with ideas, data and mentoring
  - Start thinking about this now!
  - Ok to combine projects. Clearly indicate which part of the project is done for CS224W and which part is done for the other class.
- Poster session: Dec 11 3:30-6:30pm
- **Read: http://web.stanford.edu/class/cs224w/info.html**

Course Schedule

Week	Assignment	Due on (23:59 PST)
2	**Homework 0**	Thu, October 4
3	**Homework 1**	Thu, October 11
4	**Project proposal**	Thu, October 18
5	**Homework 2**	Thu, October 25
7	**Project Milestone**	Thu, November 8
8	**Homework 3**	Thu, November 15
9	**Thanksgiving break**	
10	**Exam**	Thu, November 29
11	**Project report**	**Sun December 9** (no late periods!)
	Poster session	**Tue, December 11 3:30pm-6:30pm**

Prerequisites

- **No single topic in the course is too hard by itself**
- **But we will cover and touch upon many topics and this is what makes the course hard**
  - **Good background in:**
    - Algorithms and graph theory
    - Probability and statistics
    - Linear algebra
  - **Programming:**
    - You should be able to write non-trivial programs (in Python)
  - **2 recitation sessions (will be recorded):**
    - SNAP.PY review and installation party:
      Gates B01, Friday 9/28, 2:30-3:20 PM
    - Review of Probability, Linear Algebra, and Proof Techniques:
      Gates B01, Friday 10/5, 2:30-3:20 PM

Network Analysis Tools

- **We highly recommend SNAP:**
  - **SNAP.PY:** Python ease of use, most of C++ scalability
    - HW0 asks you to do some very basic network analysis with snap.py
      - If you find HW0 difficult, this class is probably not for you
  - **SNAP C++:** more challenging but more scalable
  - Other tools include NetworkX, iGraph

SNAP.PY review and installation party:
Gates B01, Fri 9/28, 2:30-3:20 PM

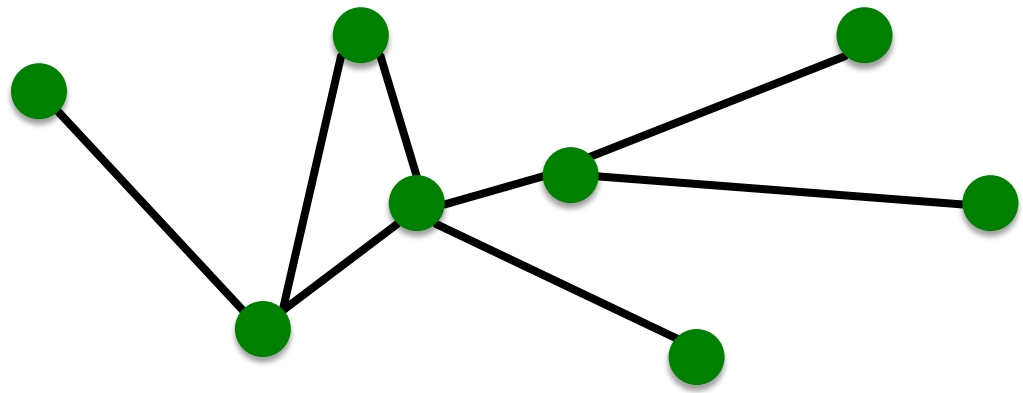
Starter Topic: Structure of Graphs

Structure of Networks?

A network is a collection of objects where some pairs of objects are connected by links

What is the structure of the network?

Components of a Network



- **Objects:** nodes, vertices N
- **Interactions:** links, edges E
- **System:** network, graph $G(N,E)$

Networks or Graphs?

- **Network** often refers to real systems
  - Web, Social network, Metabolic network

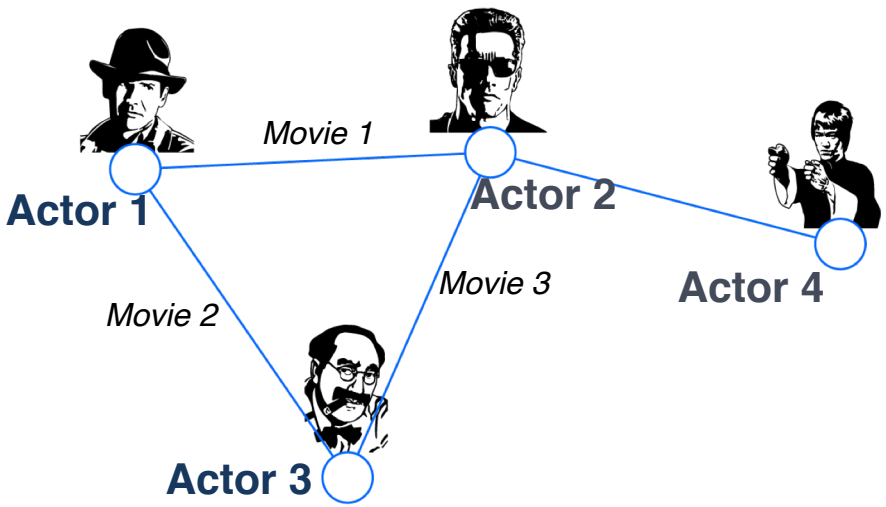
  Language: Network, node, link

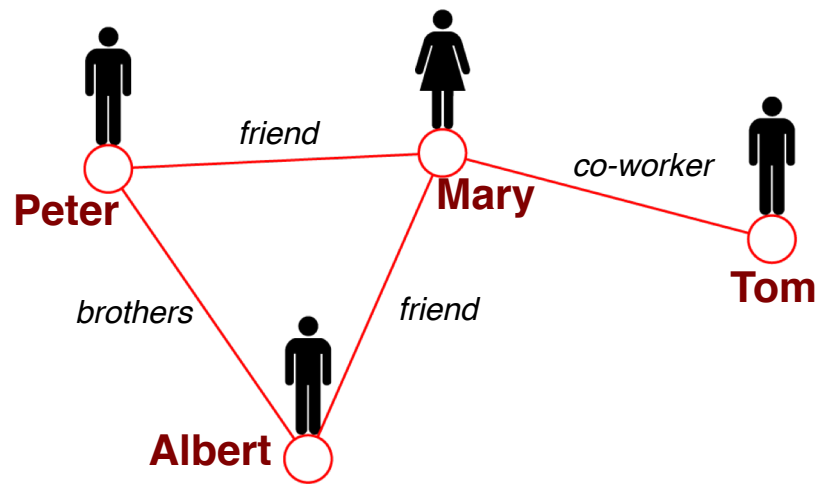
- **Graph** is a mathematical representation of a network
  - Web graph, Social graph (a Facebook term)

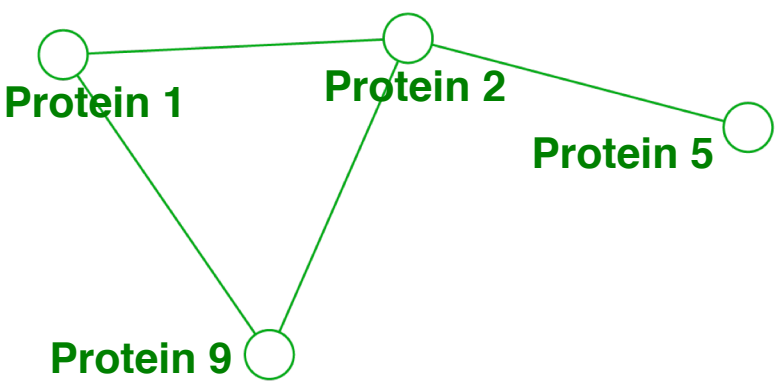
  Language: Graph, vertex, edge

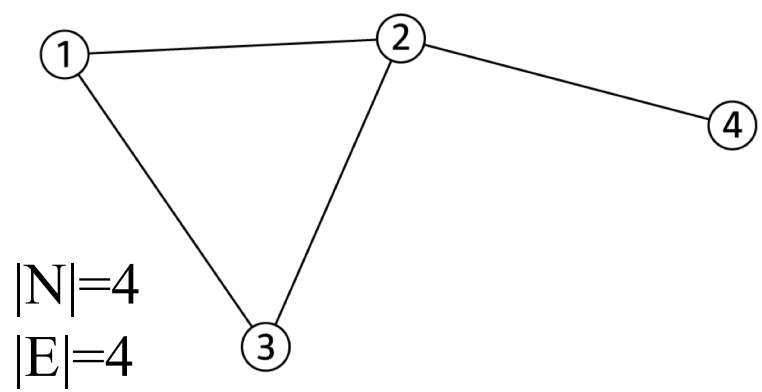
We will try to make this distinction whenever it is appropriate, but in most cases we will use the two terms interchangeably

Networks: Common Language



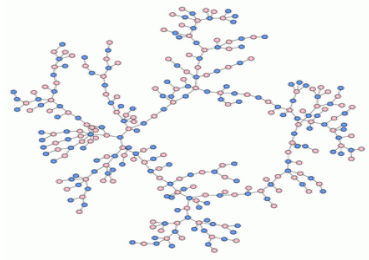


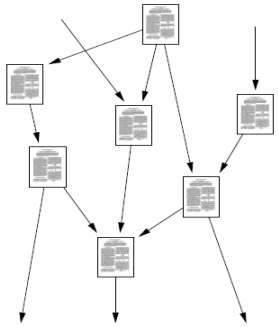




Choosing Proper Representations

- If you connect individuals that work with each other, you will explore a **professional network**
- If you connect those that have a sexual relationship, you will be exploring **sexual networks**
- If you connect scientific papers that cite each other, you will be studying the **citation network**





- **If you connect all papers with the same word in the title, what will you be exploring?** It is a network, nevertheless

How do you define a network?

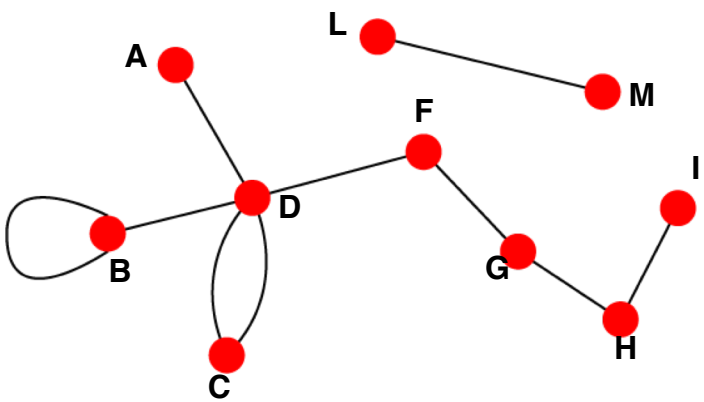
- **How to build a graph:**
  - **What are nodes?**
  - **What are edges?**
- **Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:**
  - In some cases there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature of the question you can study

Choice of Network Representation

Directed vs. Undirected Graphs

Undirected

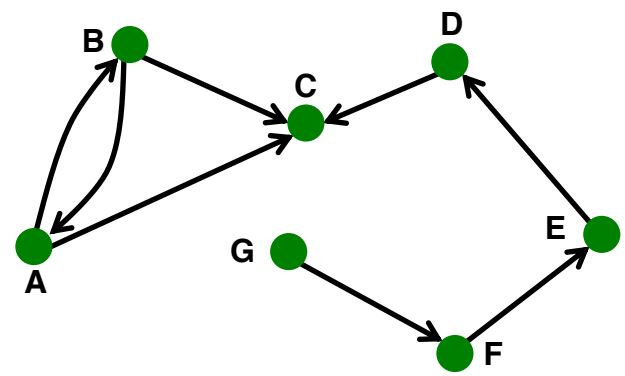
- **Links:** undirected (symmetrical, reciprocal)



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

Directed

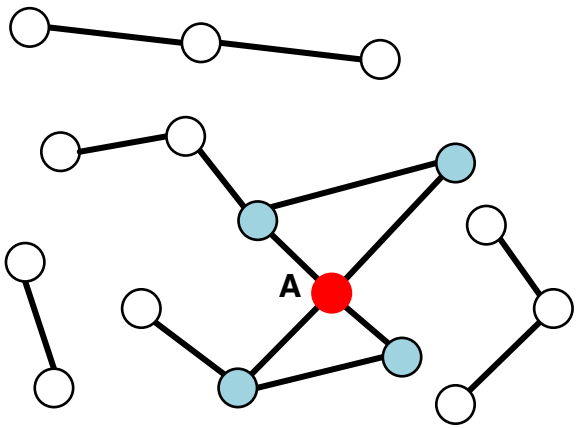
- **Links:** directed (arcs)



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

Node Degrees

Undirected

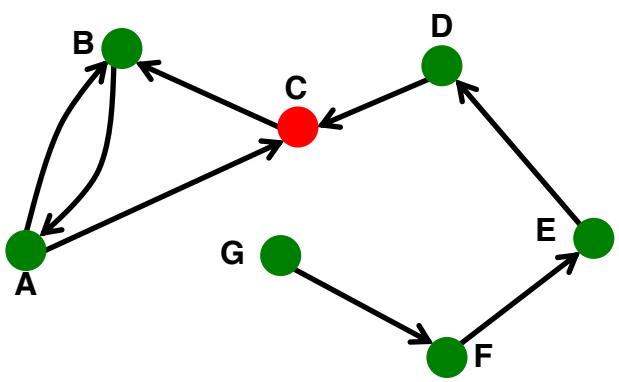


Node degree, k_i: the number of edges adjacent to node i

$$k_A = 4$$

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

Directed



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.

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

$$\bar{k} = \frac{E}{N} \qquad \overline{k^{in}} = \overline{k^{out}}$$

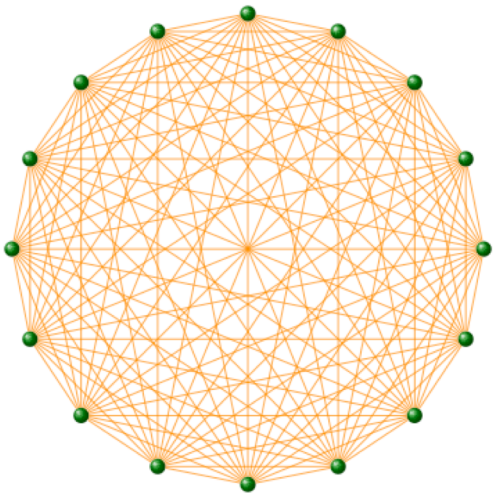
Source: Node with $k^{in} = 0$

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

Complete Graph

The **maximum number of edges** in an undirected graph on N nodes is

$$E_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$



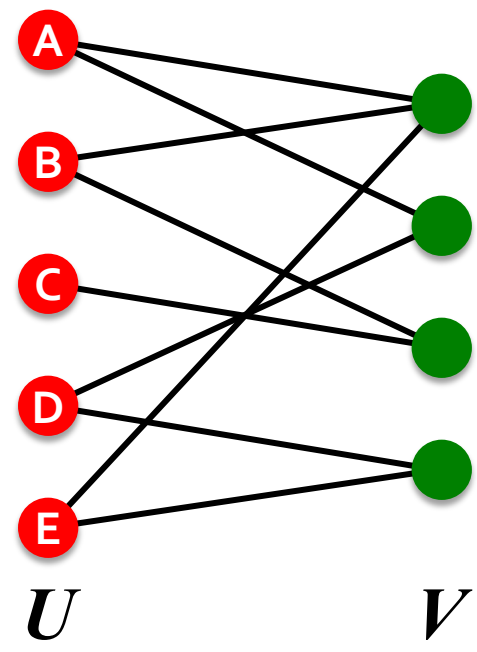
An undirected graph with the number of edges $E = E_{max}$ is called a **complete graph**, and its average degree is N-1

Directedness & Average Degrees

NETWORK	NODES	LINKS	DIRECTED UNDIRECTED	N	L	⟨k⟩
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.33
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Mobile Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Paper	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

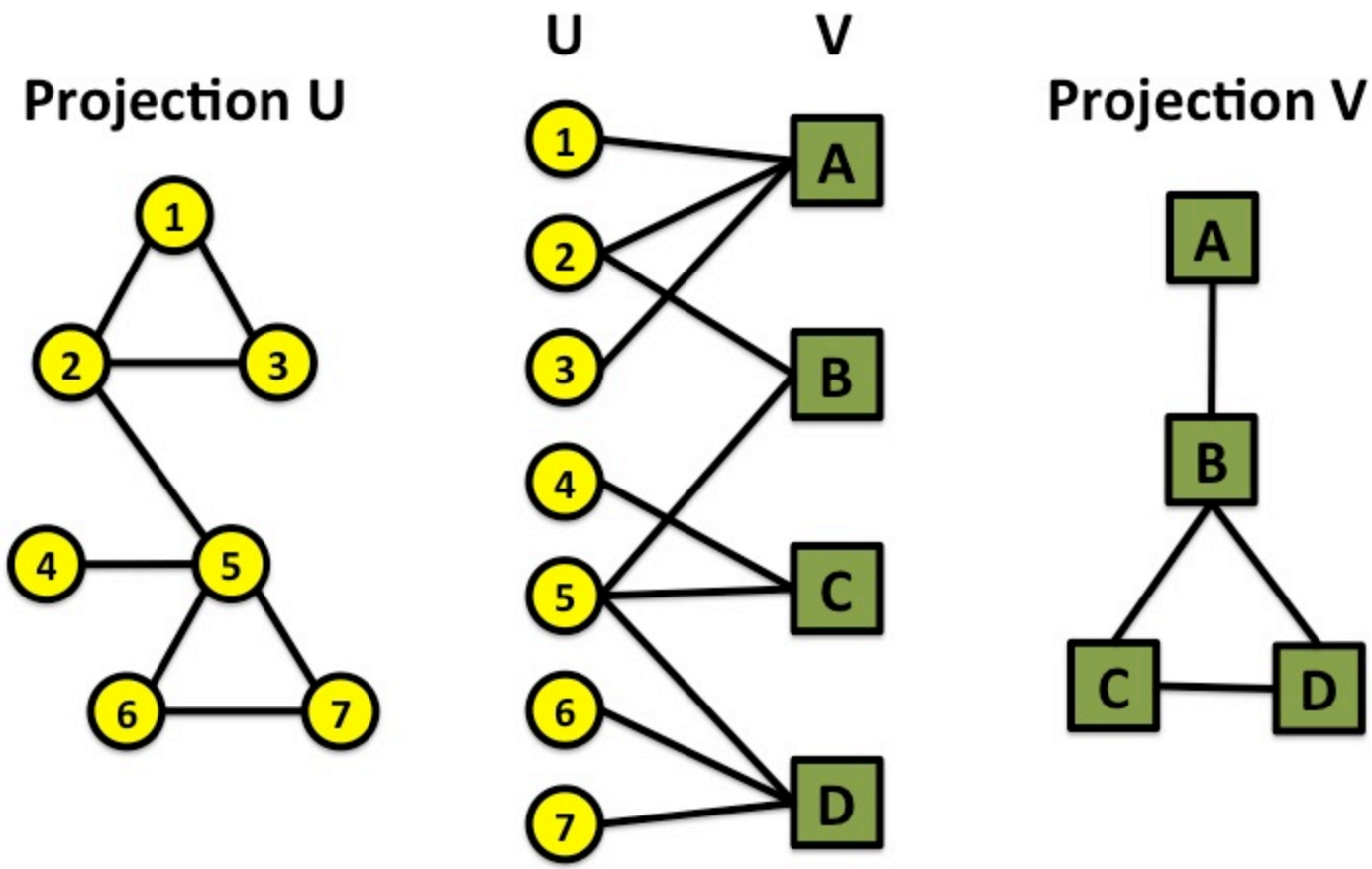
Bipartite Graph

- **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; that is, U and V are **independent sets**

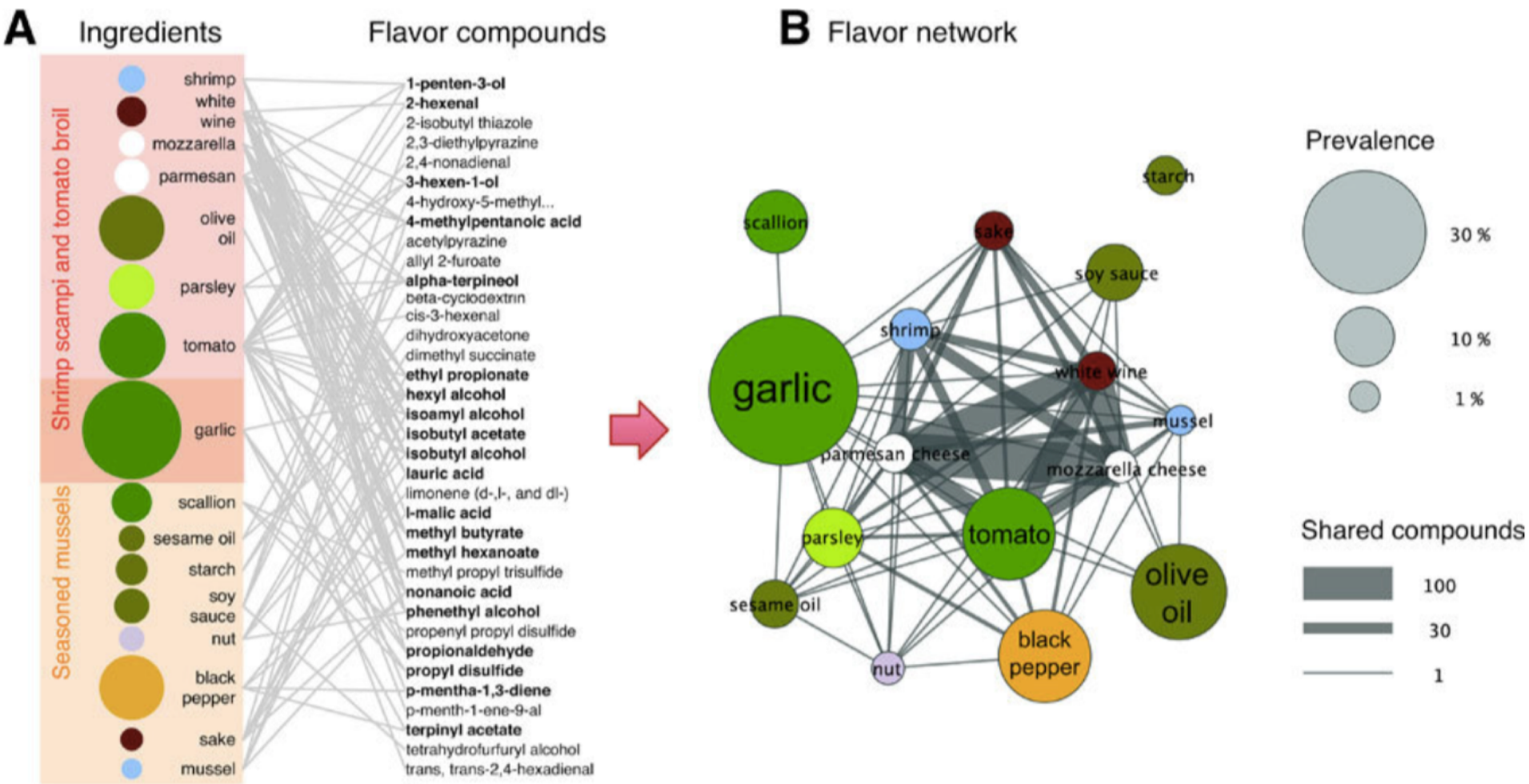


- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks

Folded/Projected Bipartite Graphs



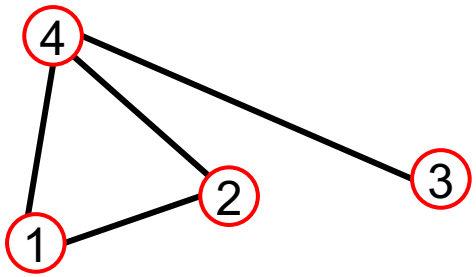
Example: Ingredients and Flavors

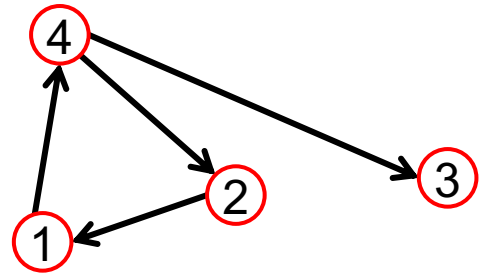


Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási
Flavor network and the principles of food pairing , Scientific Reports 196, (2011).

Network Science: Graph Theory

Representing Graphs: Adjacency Matrix





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

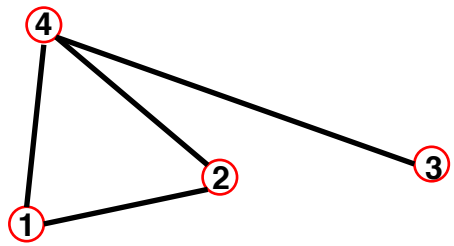
$A_{ij} = 0$ otherwise

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

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

Adjacency Matrix

Undirected



$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

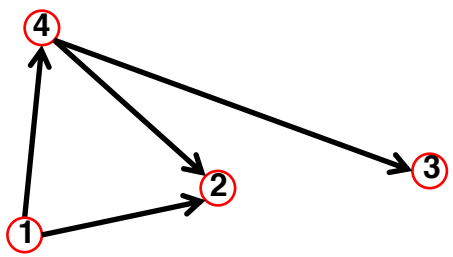
$$A_{ij} = A_{ji}$$
$$A_{ii} = 0$$

$$k_i = \sum_{j=1}^{N} A_{ij}$$

$$k_j = \sum_{i=1}^{N} A_{ij}$$

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i = \frac{1}{2}\sum_{ij}^{N} A_{ij}$$

Directed



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

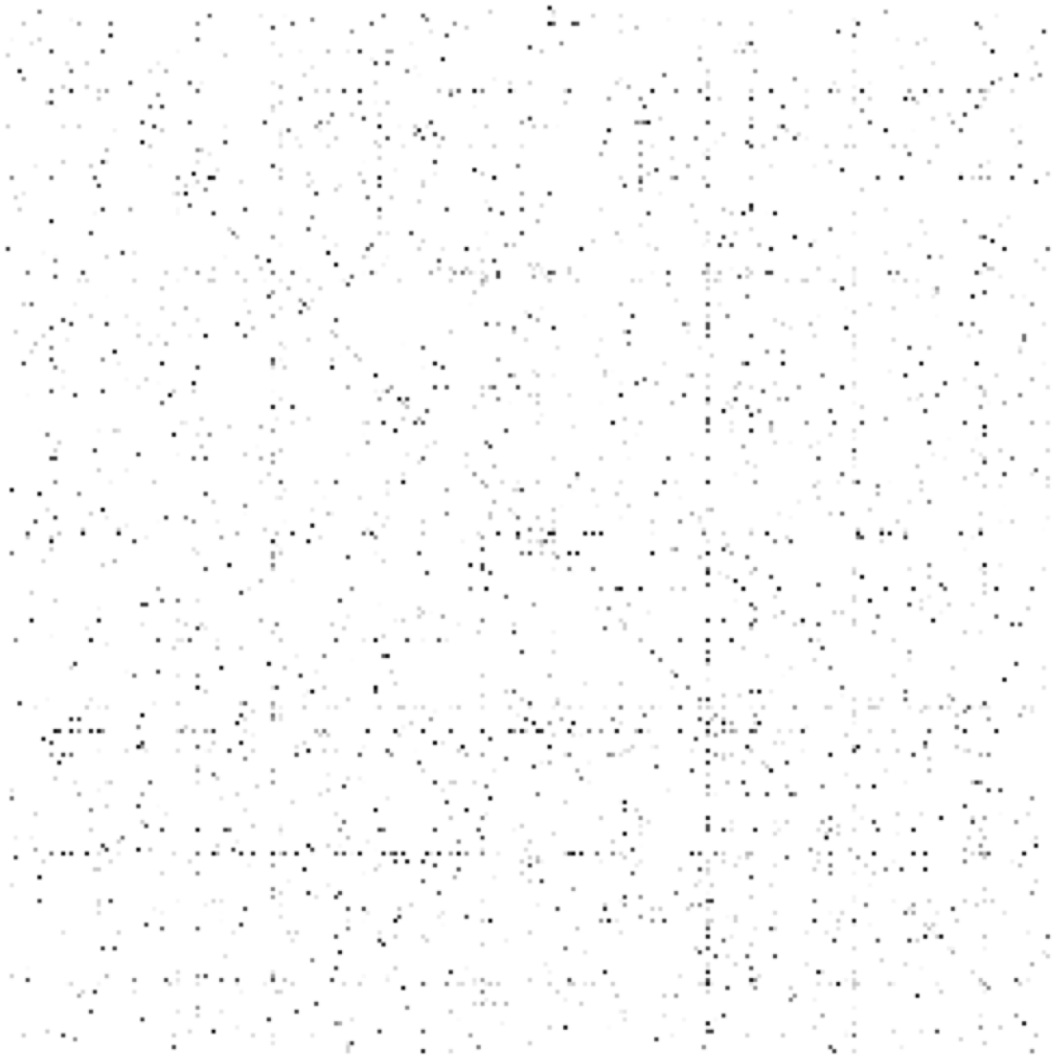
$$A_{ij} \neq A_{ji}$$
$$A_{ii} = 0$$

$$k_i^{out} = \sum_{j=1}^{N} A_{ij}$$

$$k_j^{in} = \sum_{i=1}^{N} A_{ij}$$

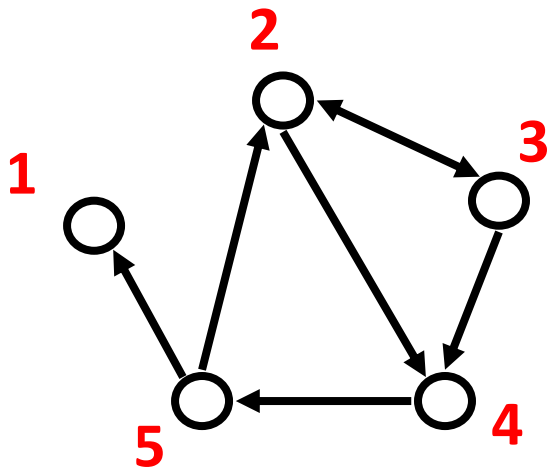
$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{j=1}^{N} k_j^{out} = \sum_{i,j}^{N} A_{ij}$$

Adjacency Matrices are Sparse



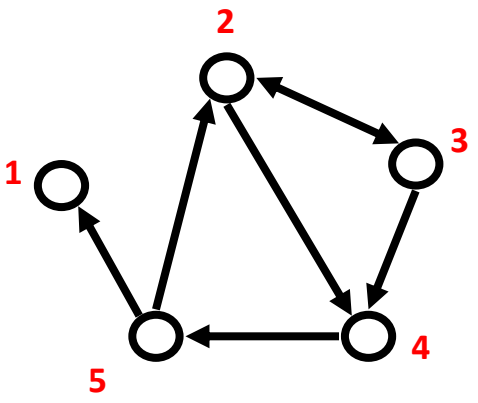
Representing Graphs: Edge list

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



Representing Graphs: 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



Networks are Sparse Graphs

Most real-world networks are sparse

$$E << E_{max} \quad (\text{or } \bar{k} << N-1)$$

WWW (Stanford-Berkeley):	N=319,717	$\langle k \rangle$=9.65
Social networks (LinkedIn):	N=6,946,668	$\langle k \rangle$=8.87
Communication (MSN IM):	N=242,720,596	$\langle k \rangle$=11.1
Coauthorships (DBLP):	N=317,080	$\langle k \rangle$=6.62
Internet (AS-Skitter):	N=1,719,037	$\langle k \rangle$=14.91
Roads (California):	N=1,957,027	$\langle k \rangle$=2.82
Proteins (S. Cerevisiae):	N=1,870	$\langle k \rangle$=2.39

(Source: *Leskovec et al., Internet Mathematics, 2009*)

Consequence: Adjacency matrix is filled with zeros!

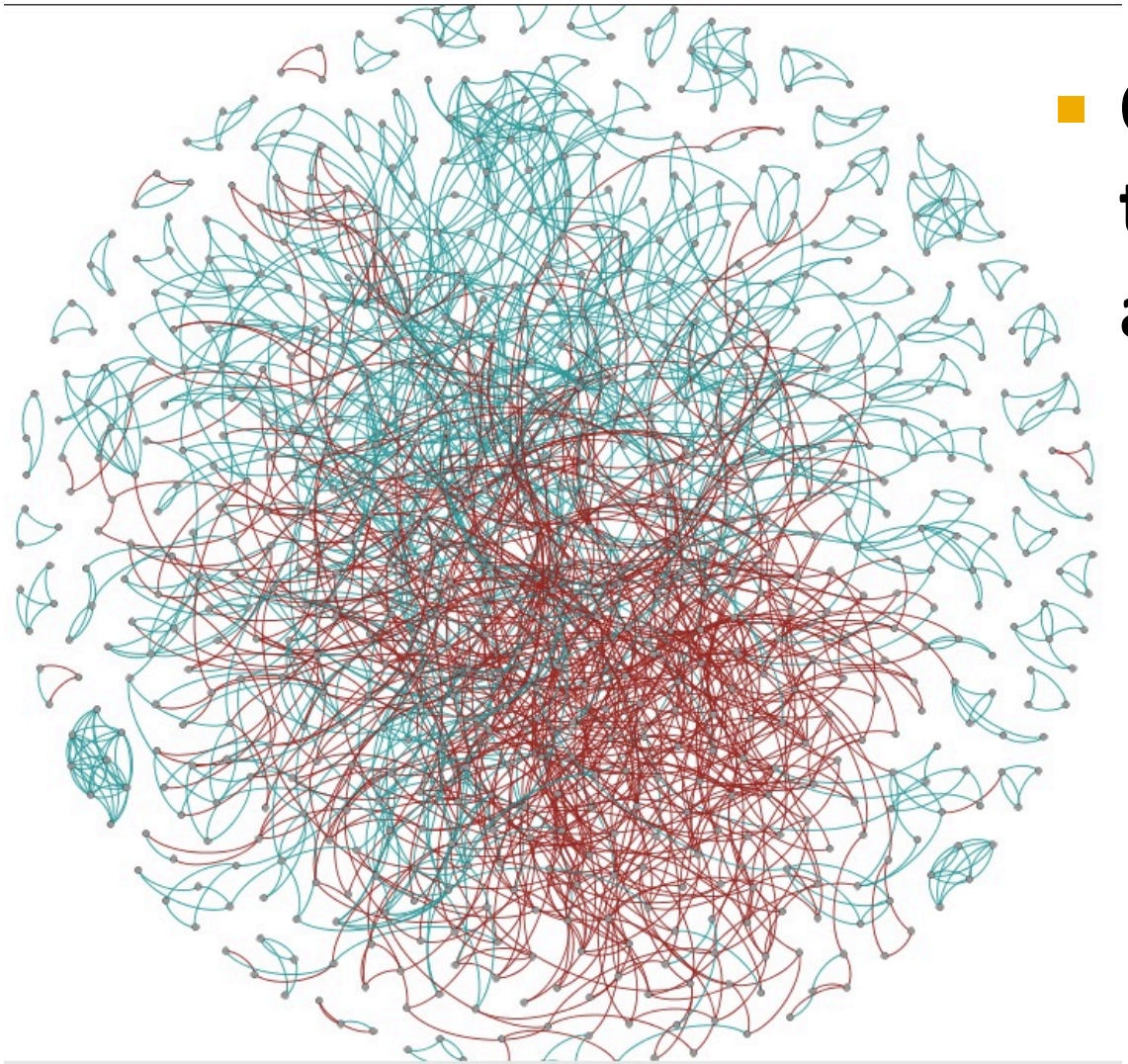
(**Density of the matrix (E/N^2):** WWW=1.51×10^{-5}, MSN IM = 2.27×10^{-8})

Edge Attributes

Possible options:

- Weight (e.g. frequency of communication)
- Ranking (best friend, second best friend…)
- Type (friend, relative, co-worker)
- Sign: Friend vs. Foe, Trust vs. Distrust
- Properties depending on the structure of the rest of the graph: number of common friends

Positive and Negative Weights

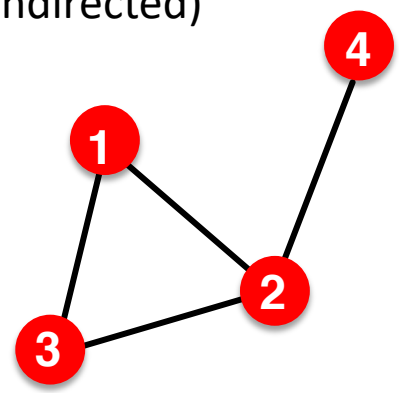


- **One person trusting/distrusting another**
  - Research challenge: How does one 'propagate' negative feelings in a social network? Is my enemy's enemy my friend?

sample of positive & negative ratings from Epinions network

More Types of Graphs

- **Unweighted**
  (undirected)



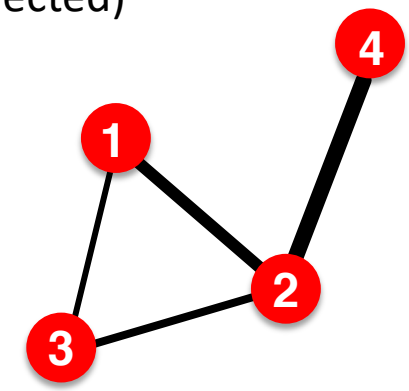
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad \bar{k} = \frac{2E}{N}$$

Examples: Friendship, Hyperlink

- **Weighted**
  (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

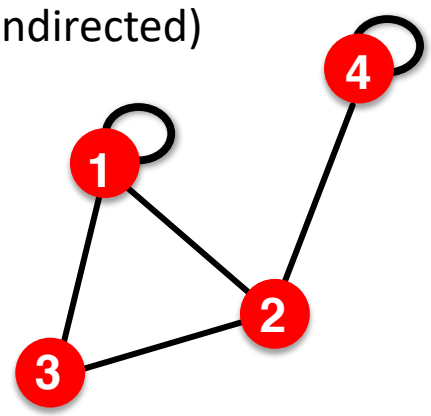
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Collaboration, Internet, Roads

More Types of Graphs

- **Self-edges (self-loops)**
  (undirected)



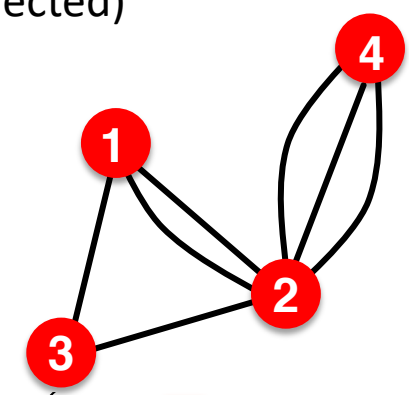
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1, i \neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$$

Examples: Proteins, Hyperlinks

- **Multigraph**
  (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

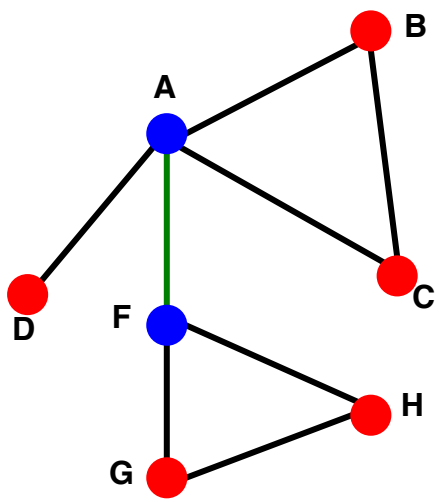
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

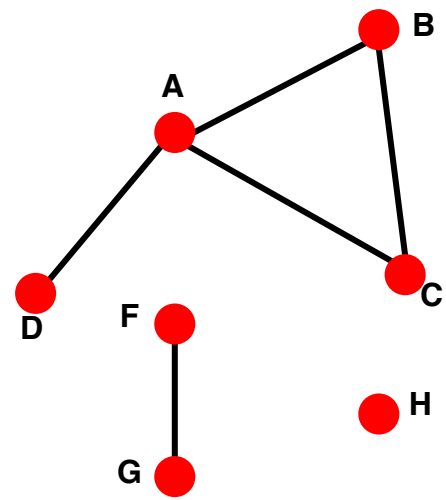
$$E = \frac{1}{2} \sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Communication, Collaboration

Connectivity of Undirected Graphs

- **Connected (undirected) graph:**
  - Any two vertices can be joined by a path
- A disconnected graph is made up by two or more connected components





Largest Component:
Giant Component

Isolated node (node H)

Bridge edge: If we erase the **edge**, the graph becomes disconnected

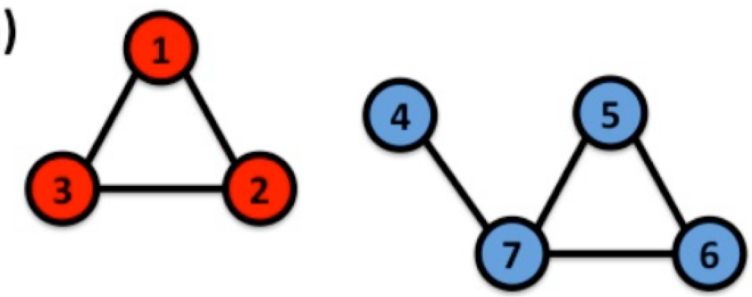
Articulation node: If we erase the **node**, the graph becomes disconnected

Connectivity: Example

- The adjacency matrix of a network with several components can be written in a block- diagonal form, so that nonzero elements are confined to squares, with all other elements being zero:

Disconnected

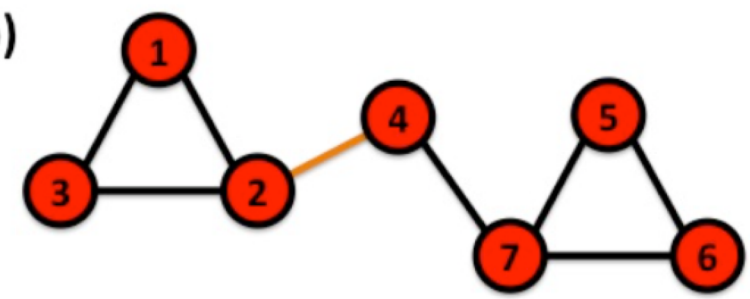
(a)



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Connected

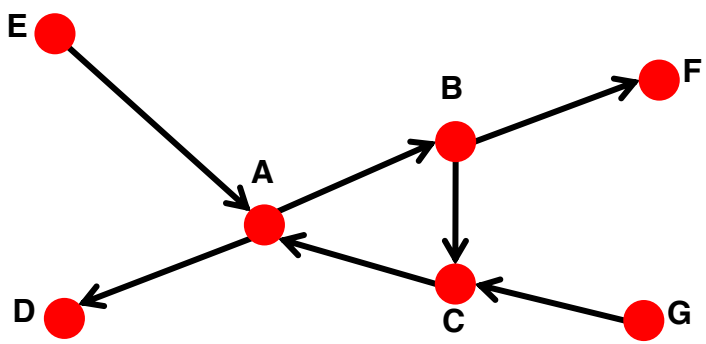
(b)



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Connectivity of Directed Graphs

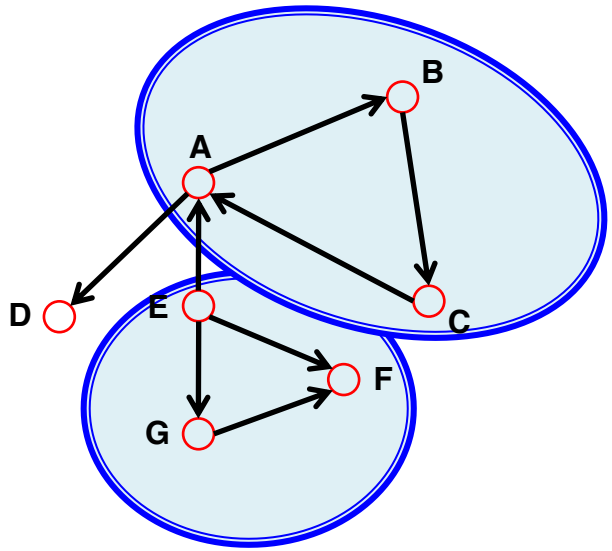
- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions

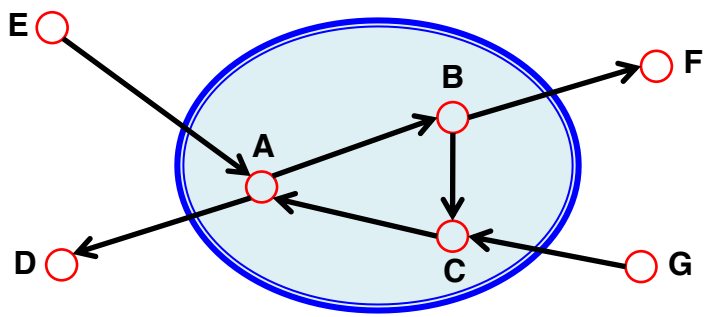


Graph on the left is connected but not strongly connected (e.g., there is no way to get from F to G by following the edge directions).

Connectivity of Directed Graphs

- **Strongly connected components (SCCs)** can be identified, but not every node is part of a nontrivial strongly connected component.





In-component: nodes that can reach the SCC,
Out-component: nodes that can be reached from the SCC.

Network Representations

Email network >> directed multigraph with self-edges

Facebook friendships >> undirected, unweighted

Citation networks >> unweighted, directed, acyclic

Collaboration networks >> undirected multigraph or weighted graph

Mobile phone calls >> directed, (weighted?) multigraph

Protein Interactions >> undirected, unweighted with self-interactions