CentraleSupélec

ESSEC BUSINESS SCHOOL

Network Science Analytics

Option Applied Math and M.Sc. in DSBA

Lecture 1A

Introduction to network science and graph mining

Fragkiskos Malliaros

Friday, January 19, 2018

About Me

- Undergrad at the University of Patras, Greece
- Ph.D. in CS at Ecole Polytechnique, Paris
- Postdoc researcher at UC San Diego
- Assistant Professor at CentraleSupélec (since October 2017)

Research interests: Data science, ML, graph mining, text mining and NLP

Office Hours

Instructor: Fragkiskos Malliaros

Office: CentraleSupélec, Gif Sur Yvette campus, CVN Lab, Room SC.217

Office hours: I will be available right after the lecture

Or, send me an email and we will find a good time to meet

Email: fragkiskos.me@gmail.com

TA: Abdulkadir Çelikkanat (Ph.D. student)

Office: CentraleSupélec, Gif Sur Yvette campus, CVN Lab

Email: abdcelikkanat@gmail.com

Acknowledgements

- Part of the lecture is based on material by
  - Jure Leskovec, Stanford University
  - Manos Papagelis, York University

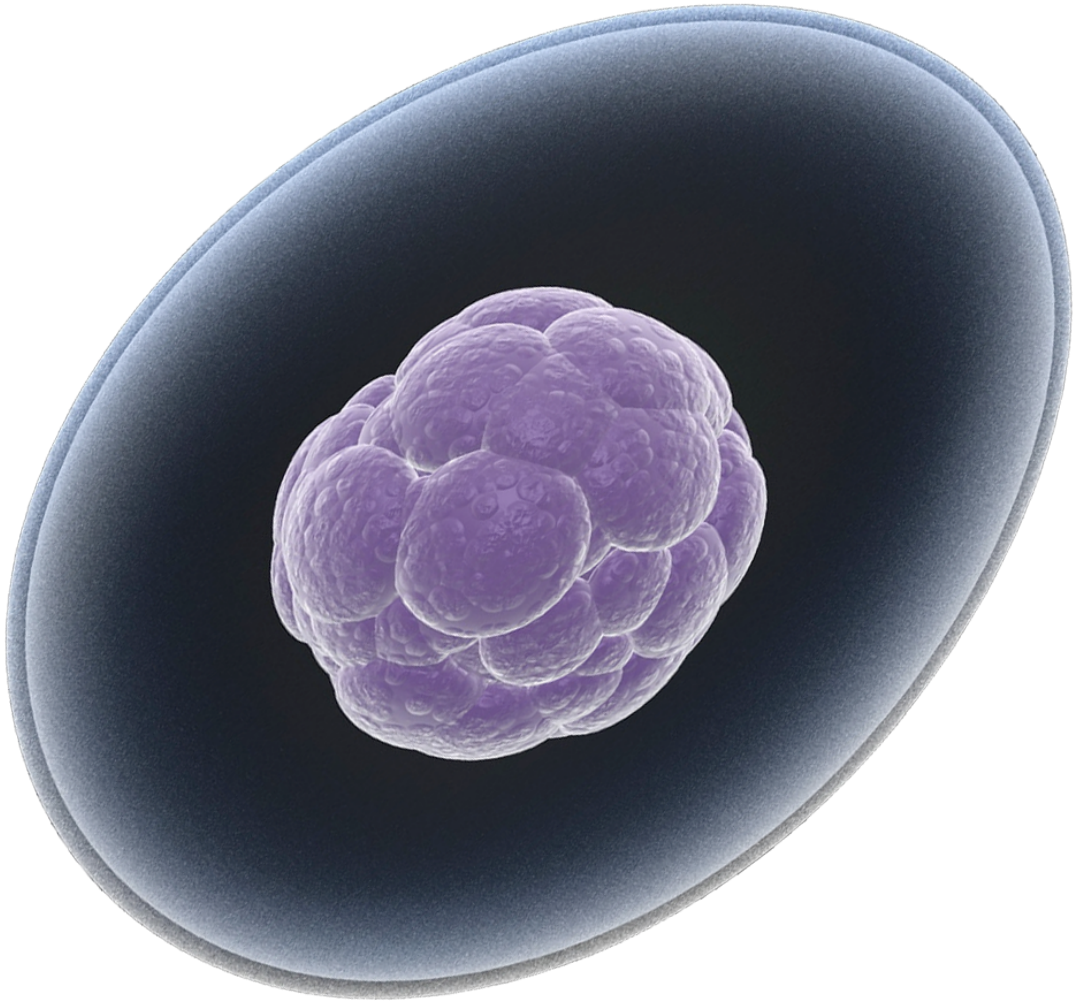
  Thank you!

Slides of Today's Lecture

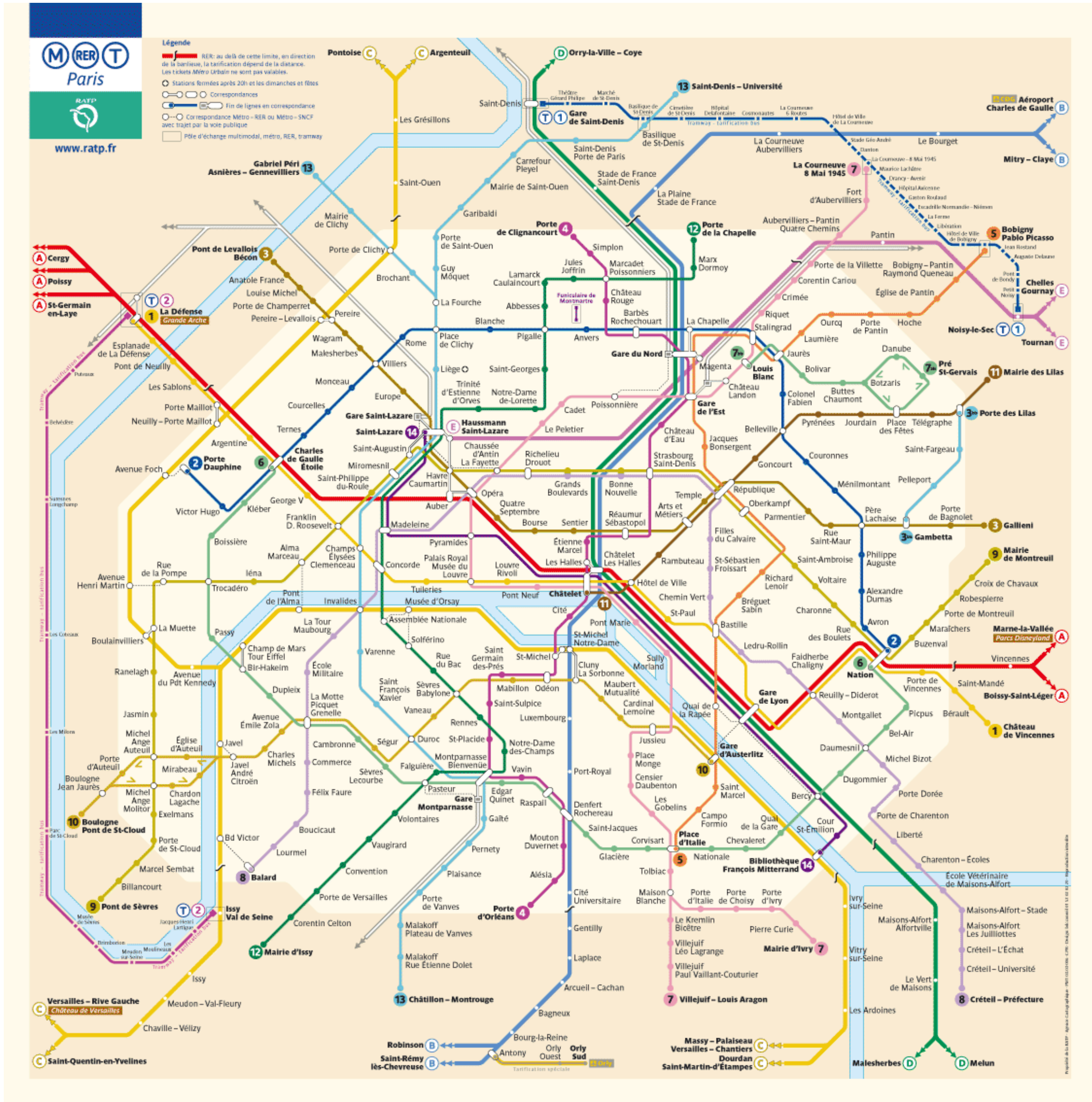
http://fragkiskos.me/ngsa-introA.pdf
http://fragkiskos.me/ngsa-introB.pdf

What do the following things have in common?

World economy

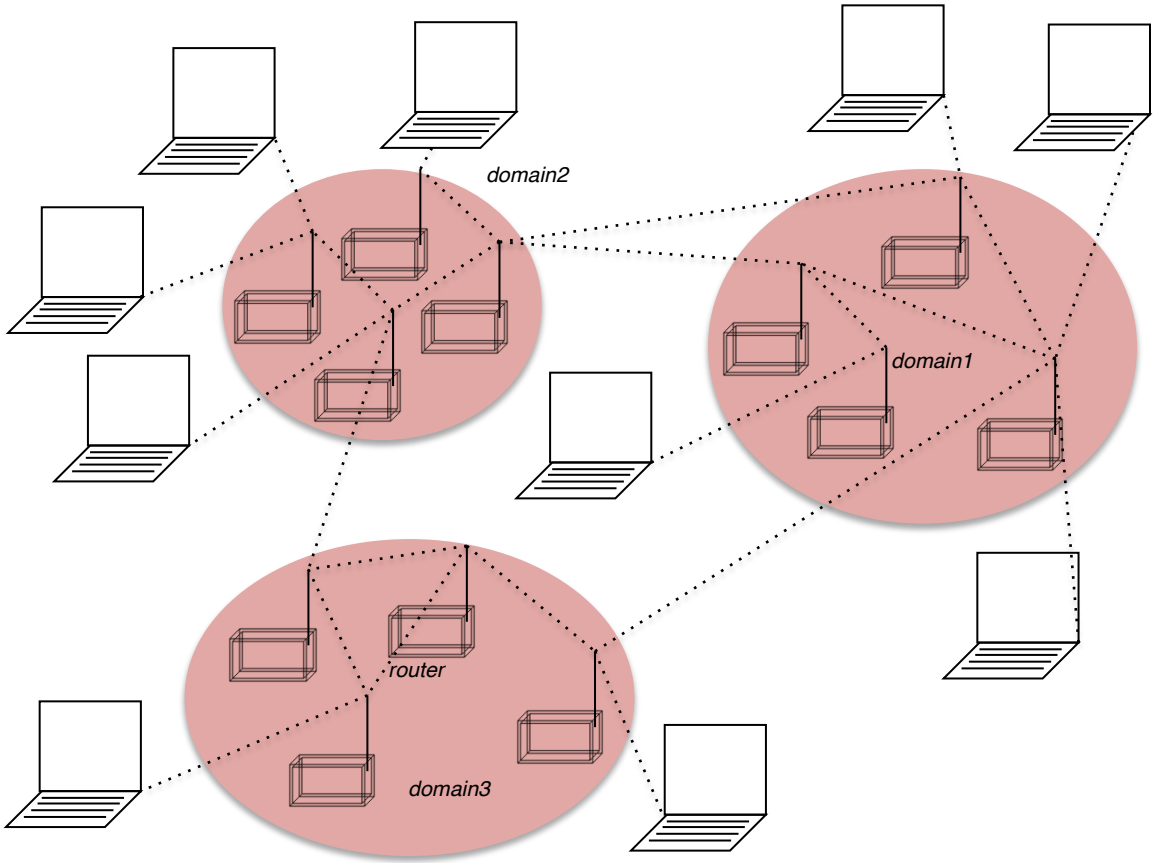


Human cell


Paris
RATP
www.ratp.fr
Légende
Pontoise
Argenteuil
Orry-la-Ville – Coye
Saint-Denis – Université
Aéroport Charles de Gaulle
Mitry – Claye
Gabriel Péri Asnières – Gennevilliers
Cergy
Poissy
St-Germain en-Laye
La Défense Grande Arche
Pont de Levallois Bécon
Porte de Clignancourt
Porte de la Chapelle
Bobigny Pablo Picasso
Chelles Gournay
Tournan
Mairie des Lilas
Porte des Lilas
Gallieni
Mairie de Montreuil
Marne-la-Vallée
Boissy-Saint-Léger
Château de Vincennes
Gare Saint-Lazare
Gare du Nord
Gare de l'Est
Charles de Gaulle Étoile
Porte Dauphine
Châtelet
Gare de Lyon
Gare d'Austerlitz
Gare Montparnasse
Nation
Boulogne Pont de St-Cloud
Pont de Sèvres
Issy Val de Seine
Balard
Mairie d'Issy
Châtillon – Montrouge
Porte d'Orléans
Villejuif – Louis Aragon
Mairie d'Ivry
Créteil – Préfecture
Bibliothèque François Mitterrand
Versailles – Rive Gauche
Saint-Quentin-en-Yvelines
Robinson
Saint-Rémy lès-Chevreuse
Massy – Palaiseau
Dourdan
Saint-Martin-d'Étampes
Malesherbes
Melun

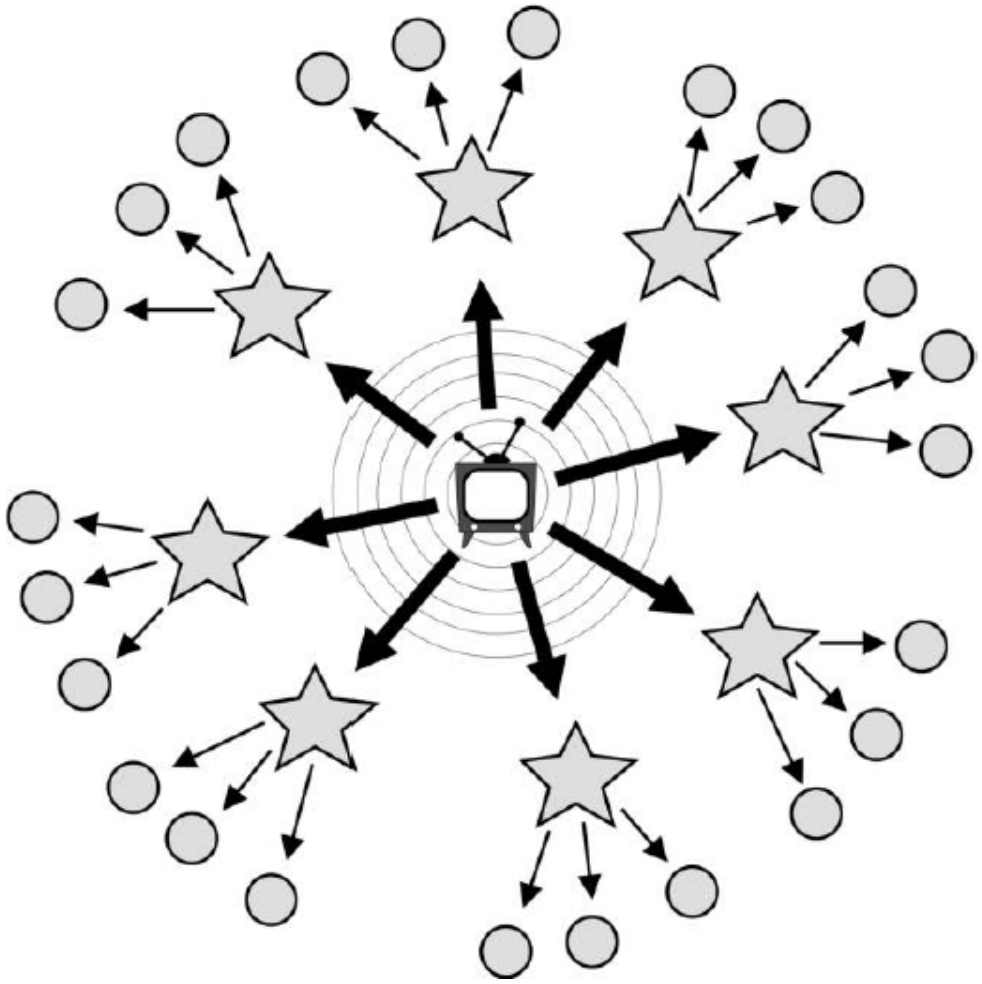
Railroads

Brain

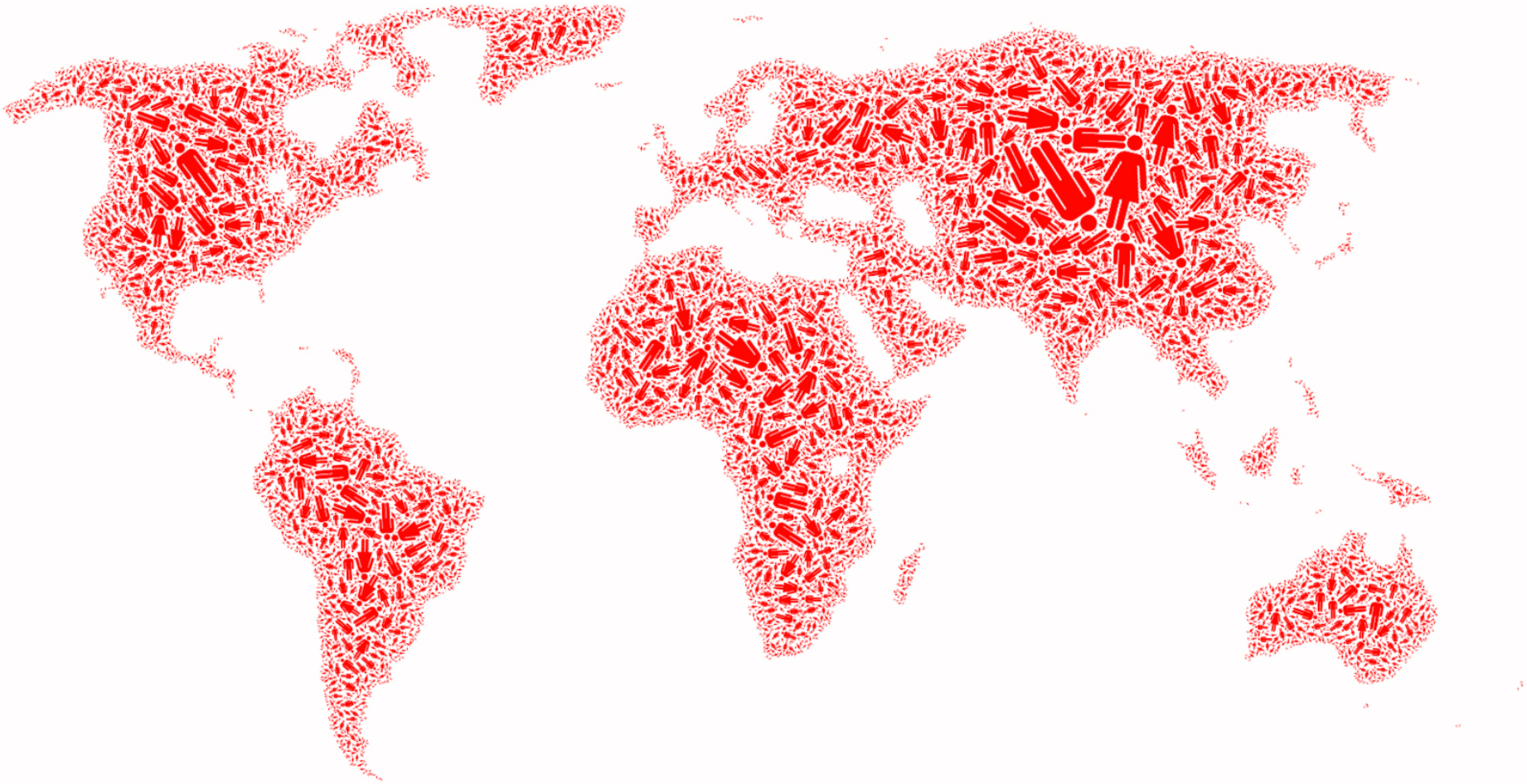


Internet

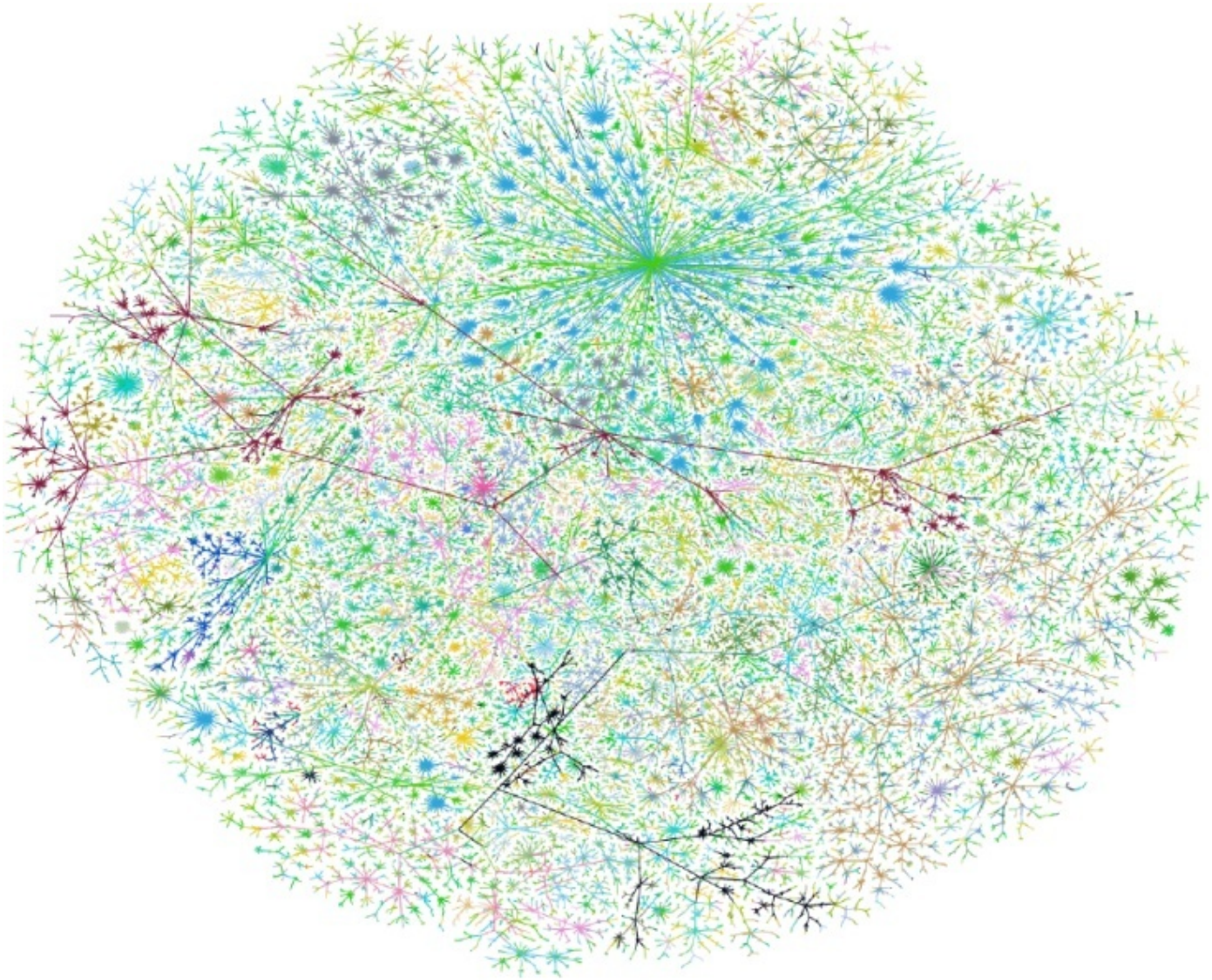
Friends & Family



Media & Information

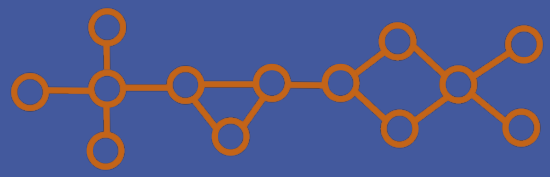


Society

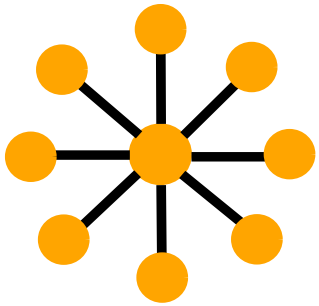


The Network!

Networks



Networks allow to model relationships between entities

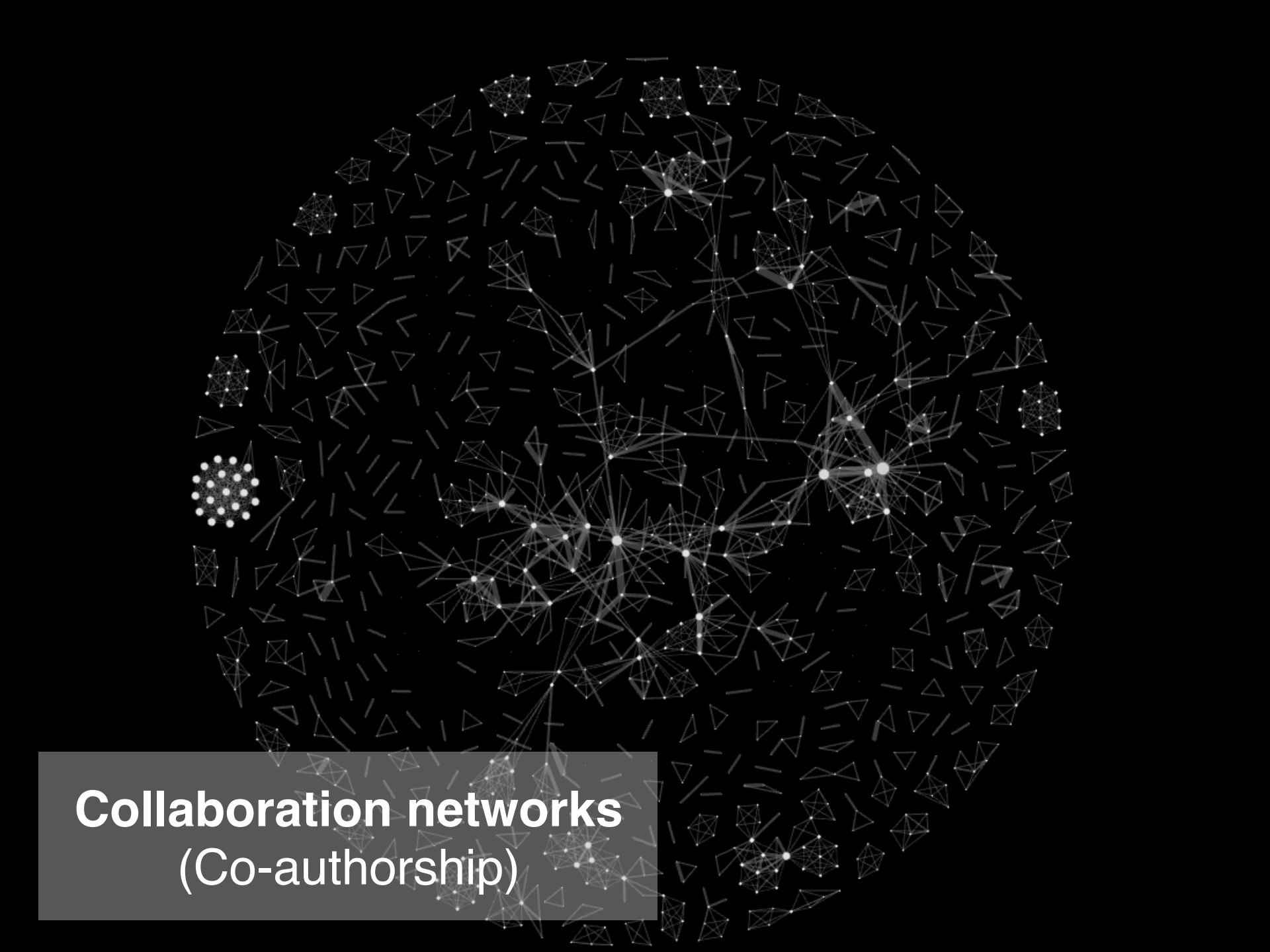


General-purpose language

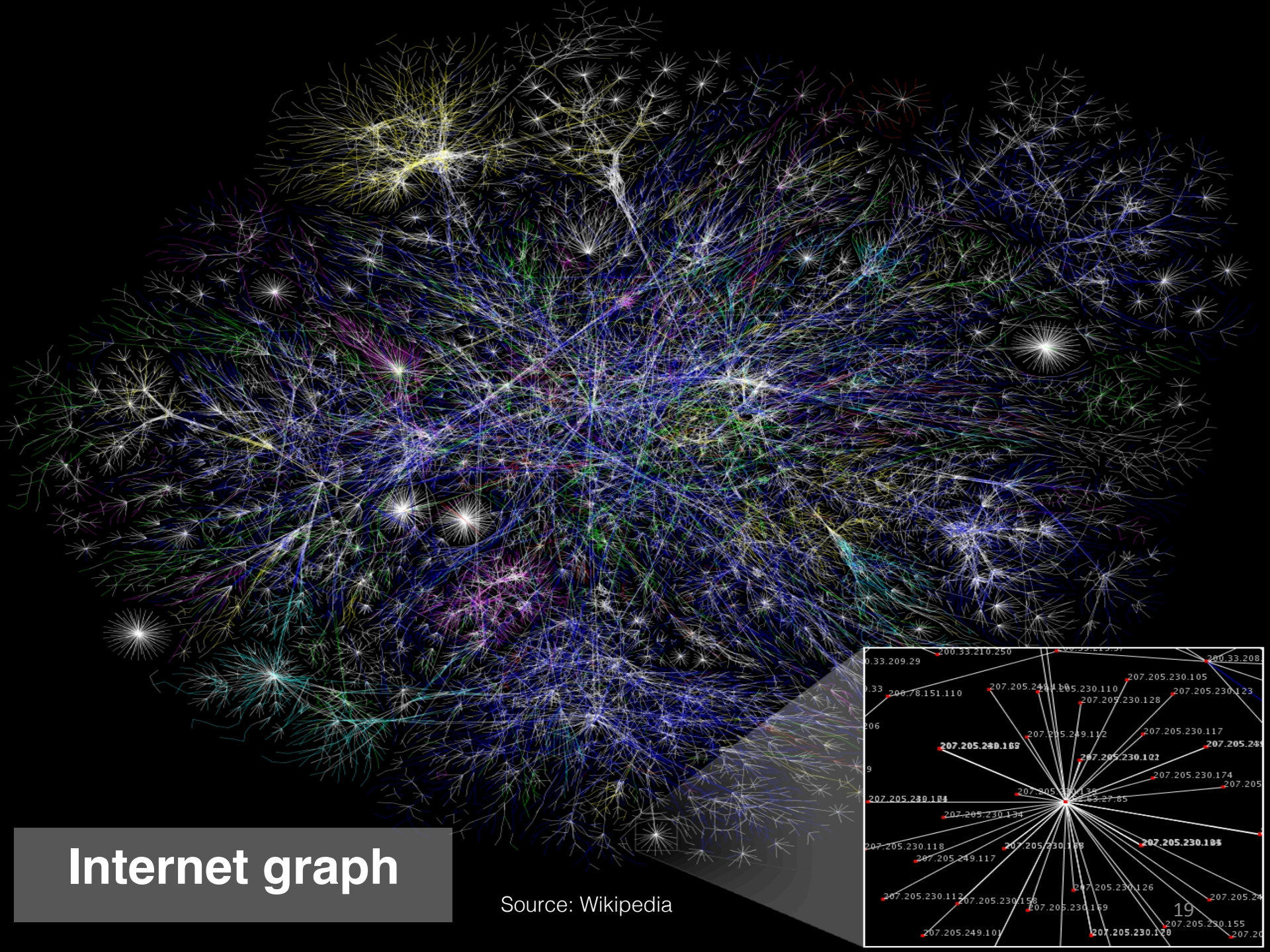
for describing real-world systems

Facebook
2.07 Billion users (Q2 of 2017)

Source: https://www.facebook.com/zuck

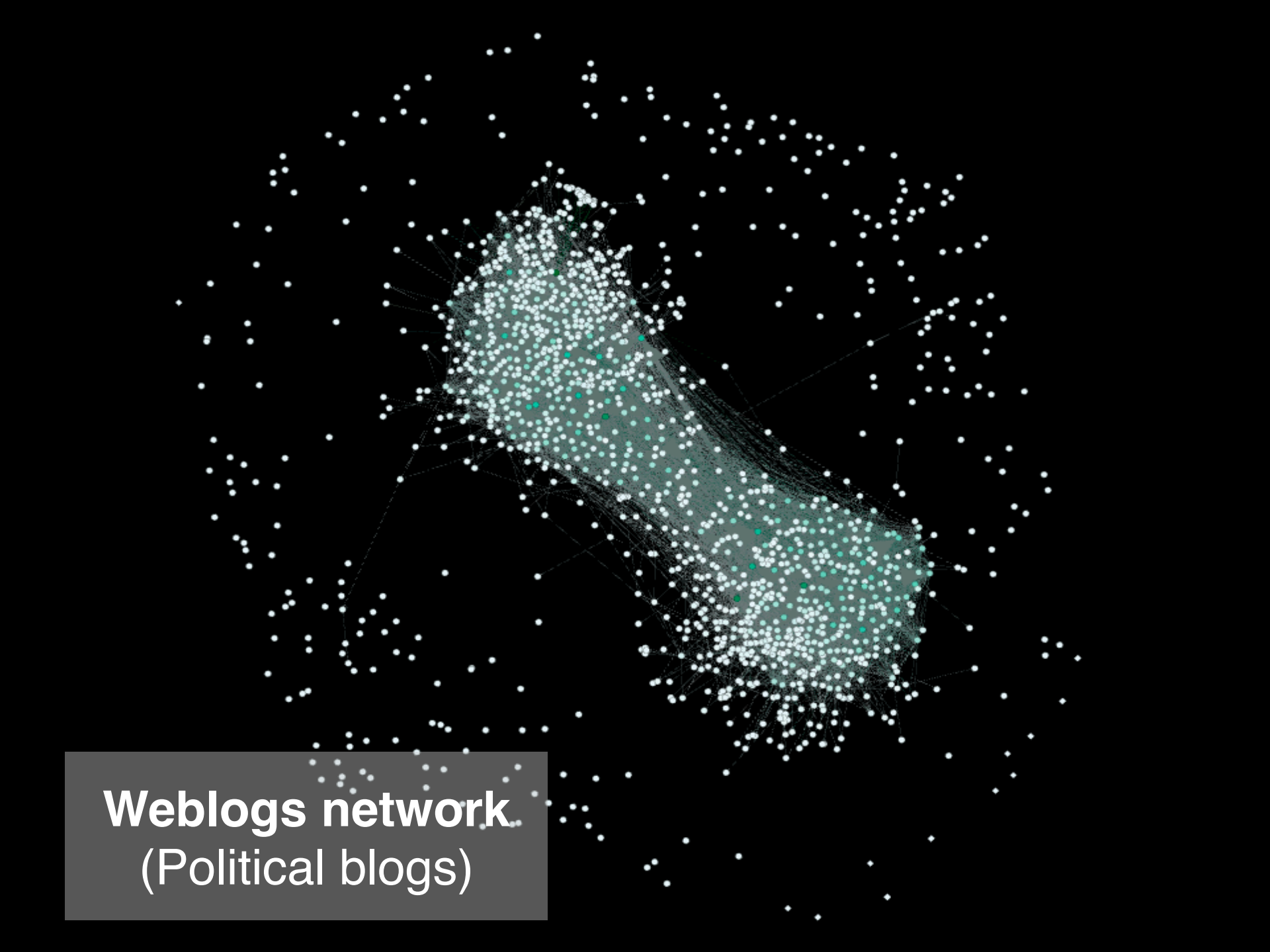


Collaboration networks
(Co-authorship)

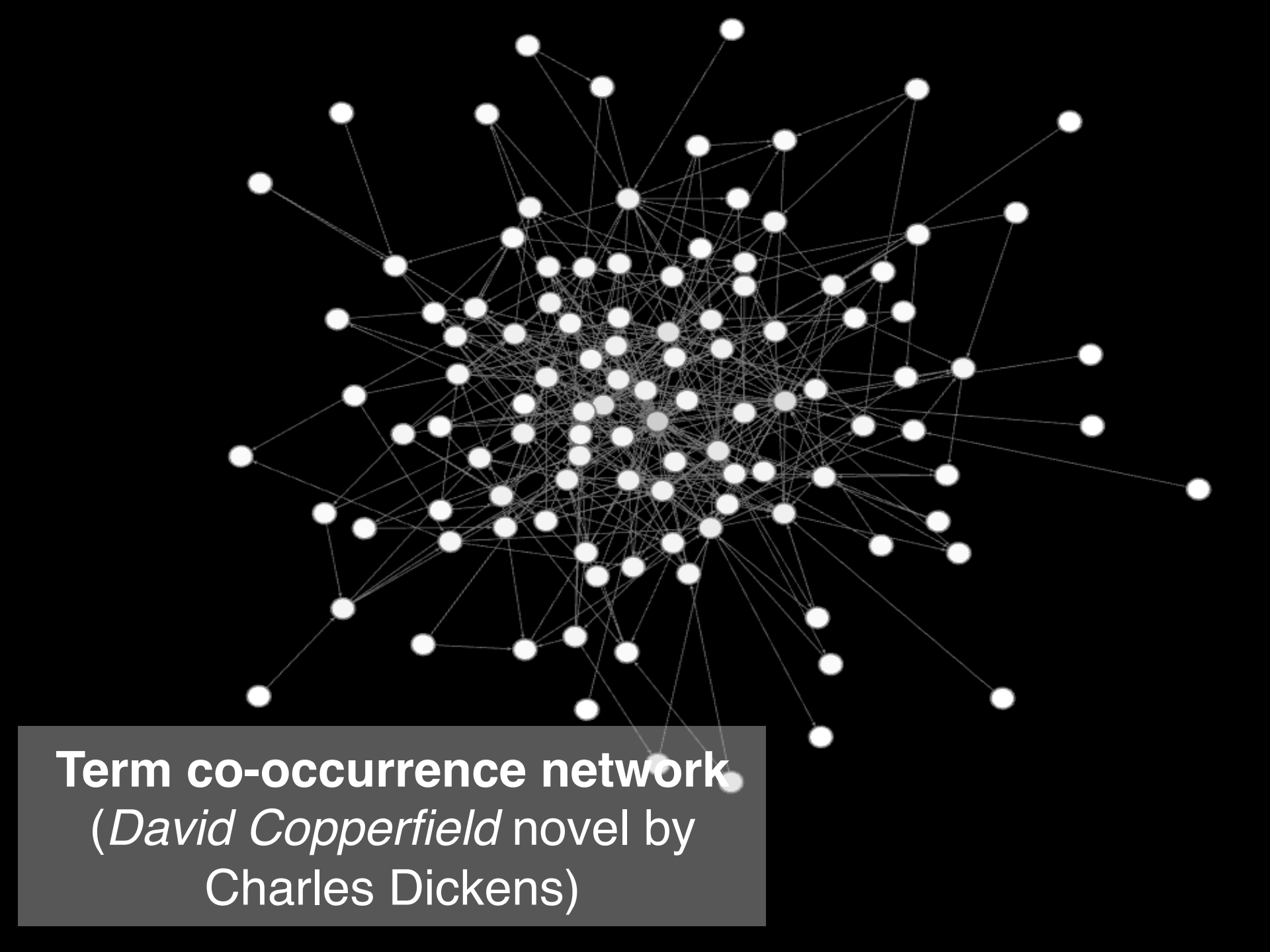


Internet graph

Source: Wikipedia

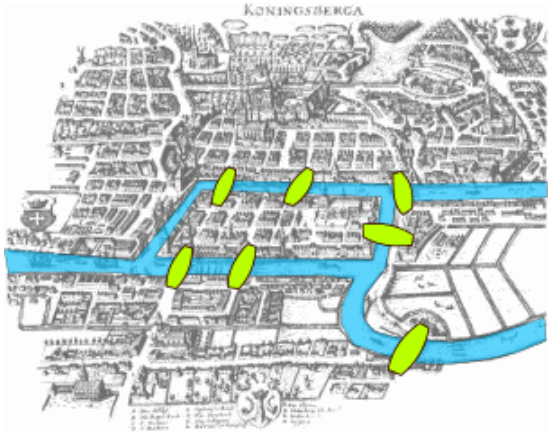


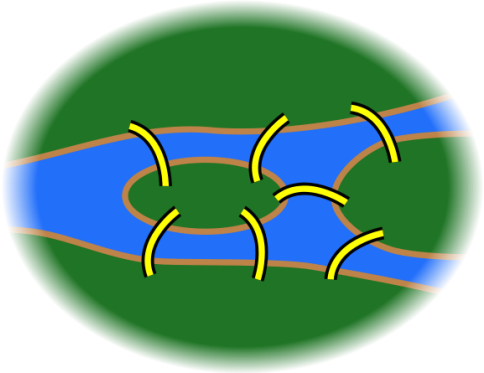
Weblogs network
(Political blogs)

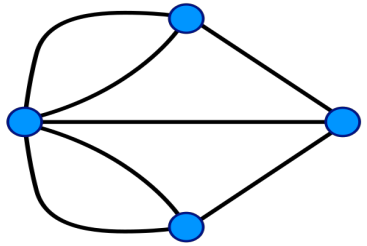


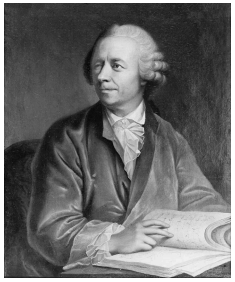
Term co-occurrence network
(*David Copperfield* novel by
Charles Dickens)

Infrastructure Networks







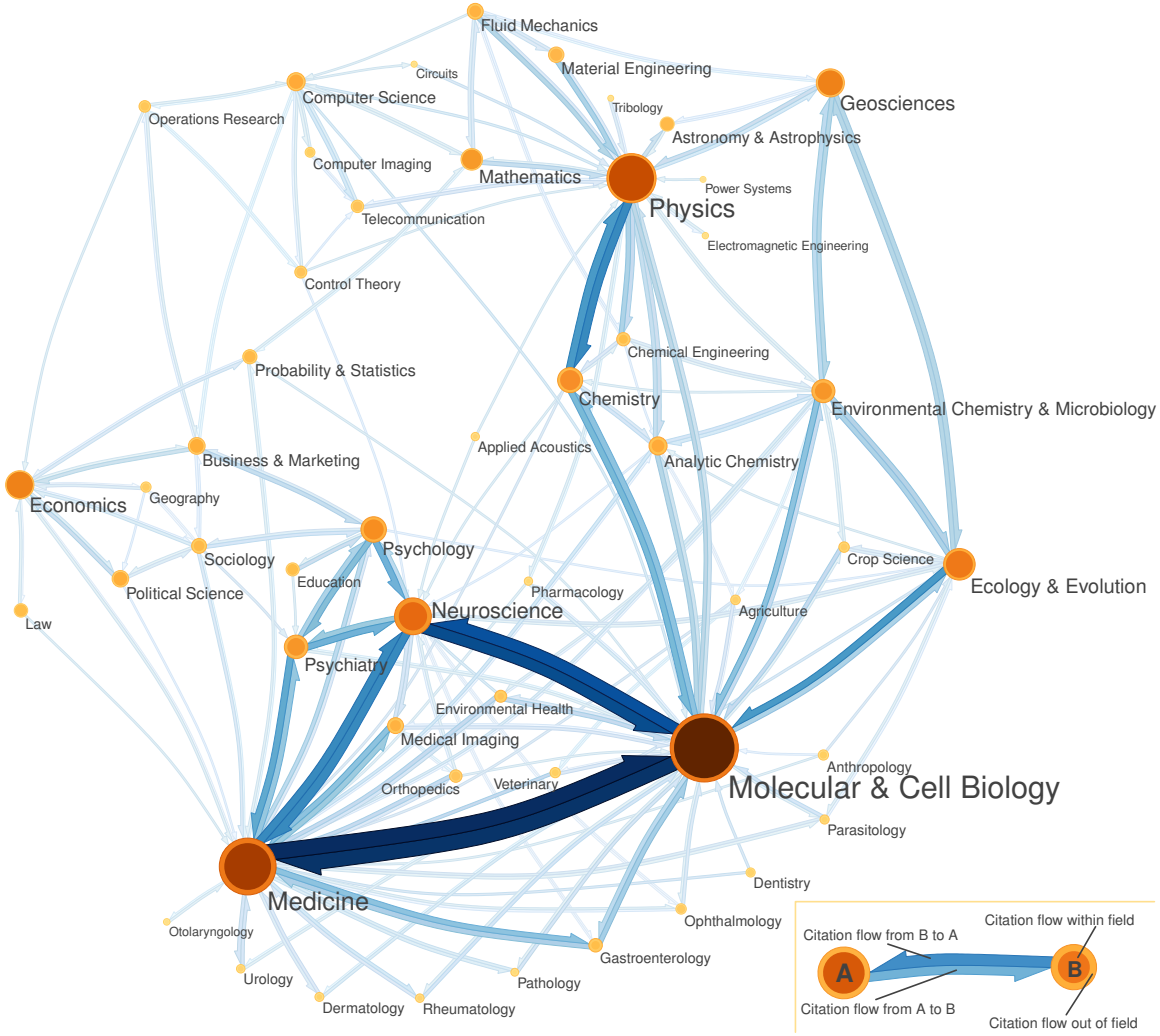


Seven Bridges of Königsberg (Kaliningrad) **[Euler, 1735]**

Devise a walk through the city that would cross each of those bridges once and only once

Source: https://en.wikipedia.org/wiki/Seven_Bridges_of_Konigsberg

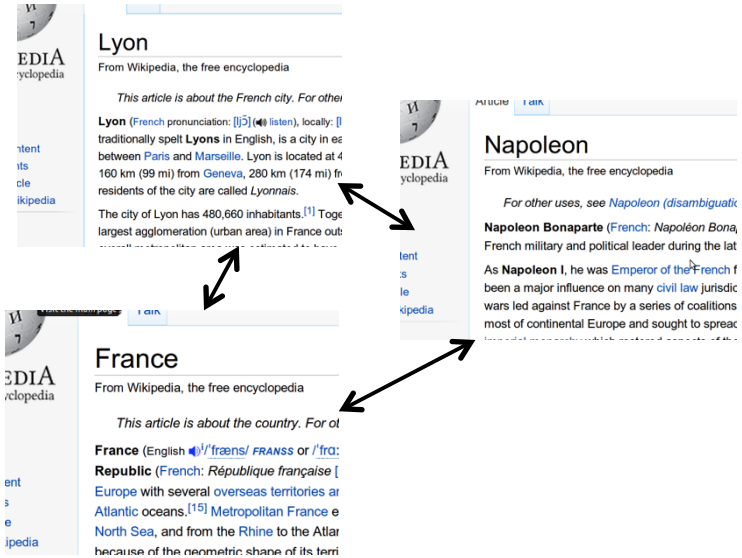
Information Networks

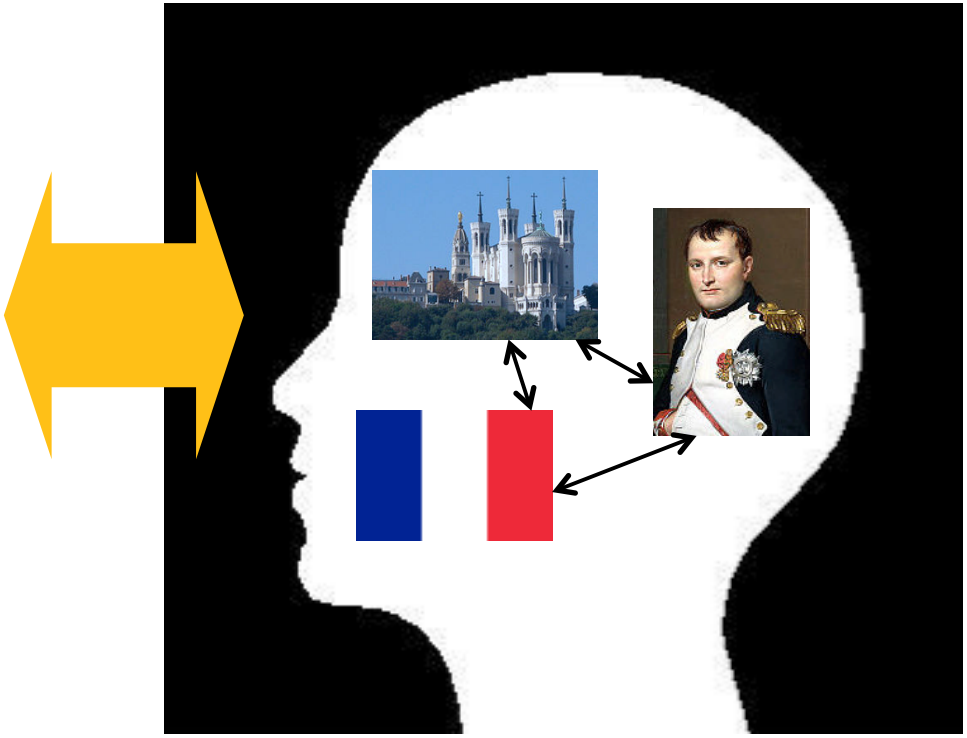


Citation networks and Map of science

[Rosvall and Bergstrom, 2008]

Knowledge Graphs

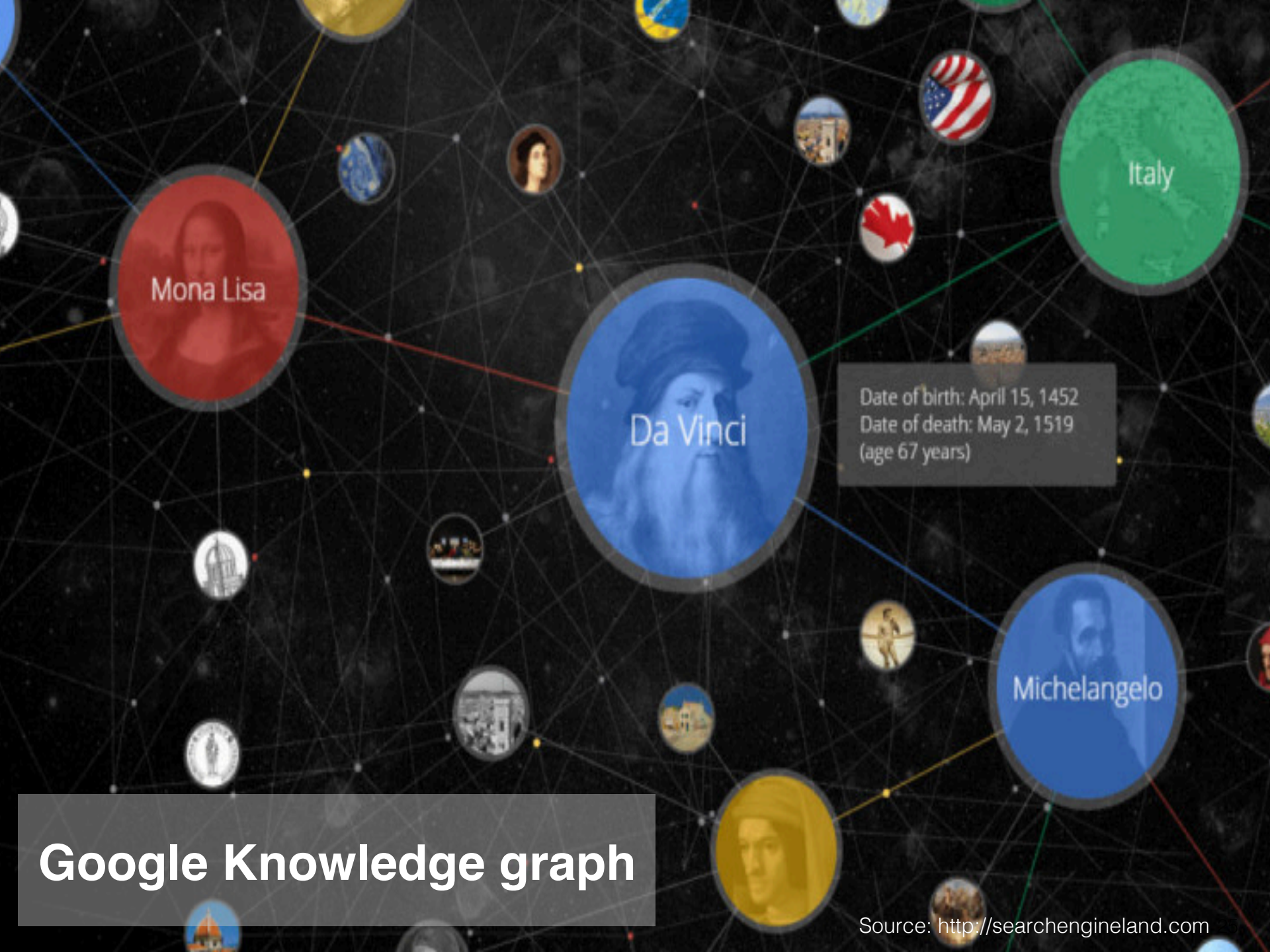




Understand how humans navigate Wikipedia

Get an idea of how people connect concepts

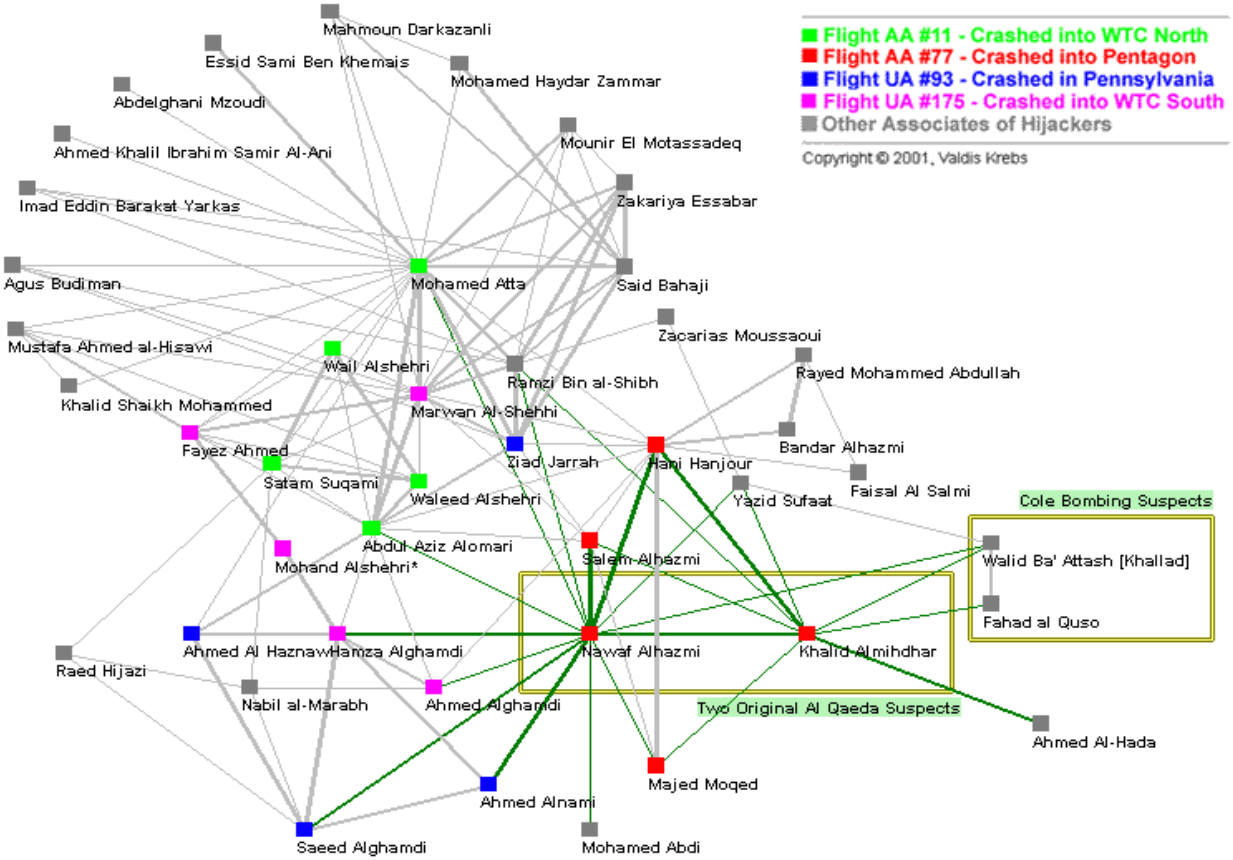
[West and Leskovec, 2012]



Google Knowledge graph

Source: http://searchengineland.com

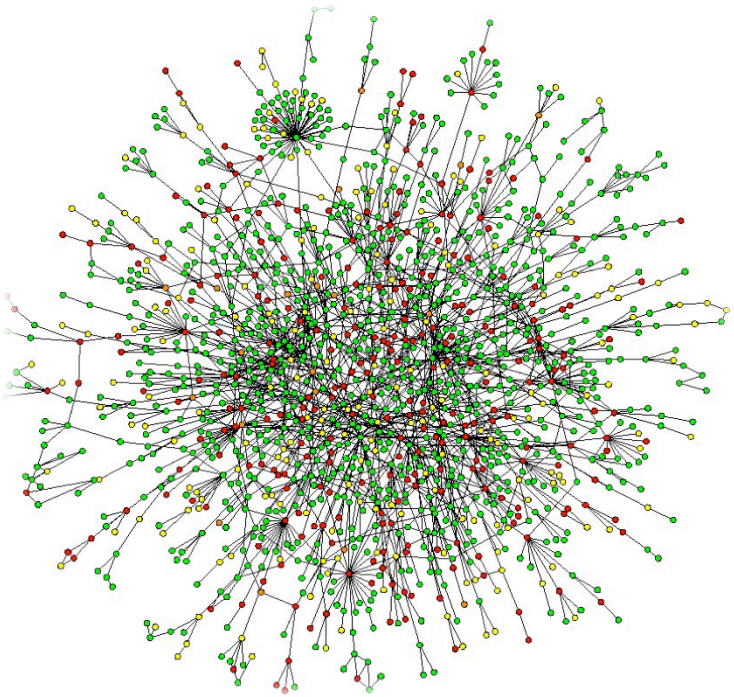
Networks of Organizations

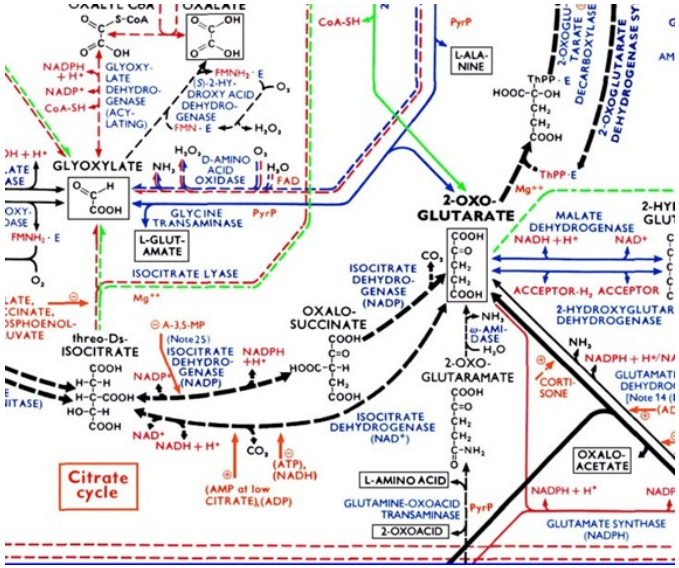


9/11 terrorist network

Source: http://www.orgnet.com/prevent.html

Biological Networks





Protein-Protein Interaction Networks:

Nodes: Proteins

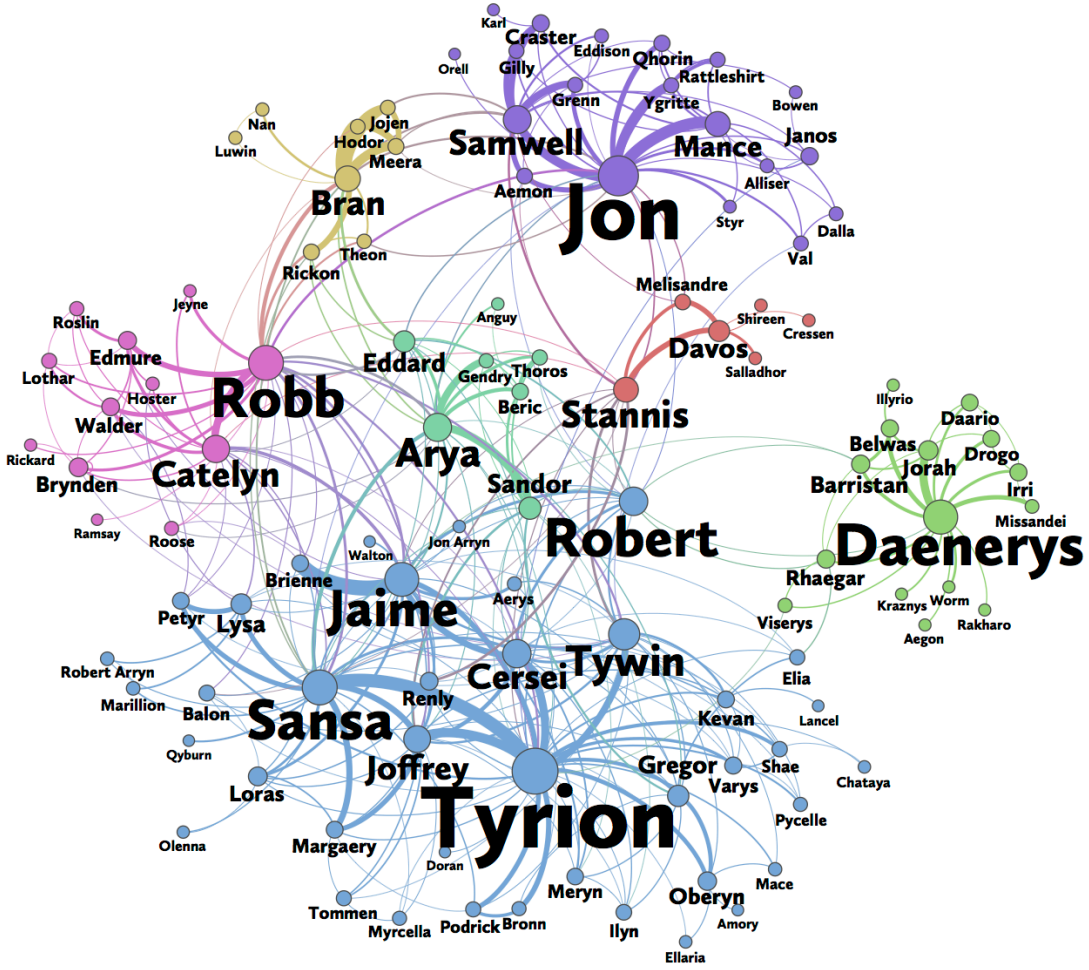
Edges: ‘physical’ interactions

Metabolic networks:

Nodes: Metabolites and enzymes

Edges: Chemical reactions

What Else?



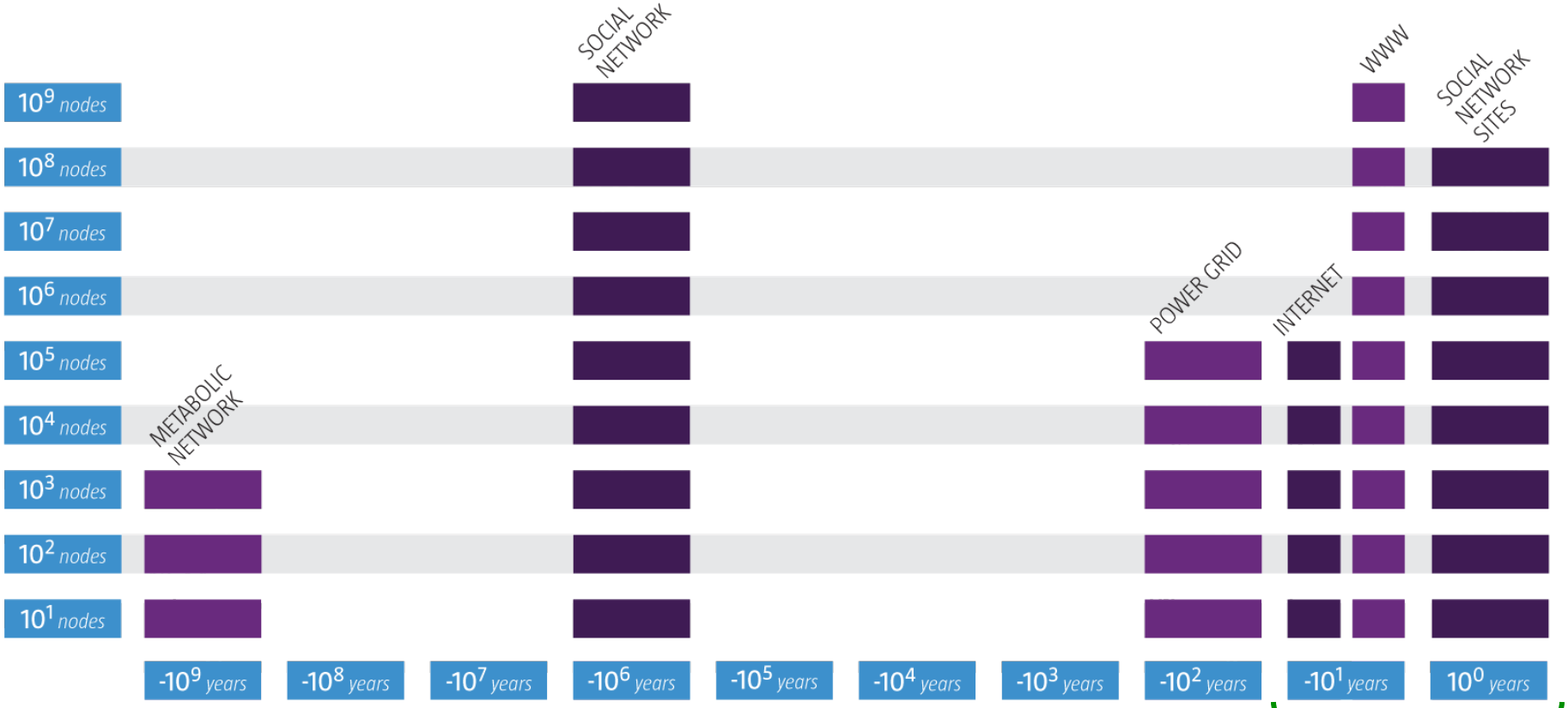
In fact, anything that captures relationships between entities can be modeled as graph (any 3-way join in the DB community)

Why should I care about networks?

Why Graphs? Why Now?

- **Universal language for describing complex data**
  - Networks from science, nature, and technology are more similar than one would expect
- **Shared vocabulary (representation) between fields**
  - Computer Science, Engineering, Social Sciences, Physics, Economics, Statistics, Biology, …
  - Cross-disciplinary topic
- **Availability of big and rich data**
  - Web/mobile, bio, health, and medical
  - Computational challenges
- **Impact**
  - Social networking and social media, recommender systems, drug design, neuroscience, epidemiology, …

Networks: Why Now?



Age and size of networks

Networks: Scale Matters

- **Network data: Orders of magnitude**
  - 436-node network of email exchange at a corporate research lab **[Adamic, Adar, SocNets '03]**
  - 43,553-node network of email exchange at an university **[Kossinets, Watts, Science '06]**
  - 4.4-million-node network of declared friendships on a blogging community **[Liben, Nowell et al., PNAS '05]**
  - 240-million-node network of communication on Microsoft Messenger **[Leskovec, Horvitz, WWW '08]**
  - 800-million-node Facebook network **[Backstrom et al. '11]**

Web – The Lab of Humanity

The Web is a "laboratory" for understanding the pulse of humanity

Tim Berners-Lee

MIT News
ON CAMPUS AND AROUND THE WORLD

Browse or Search

FULL SCREEN

Tim Berners-Lee was honored with the Turing Award for his work inventing the World Wide Web, the first web browser, and "the fundamental protocols and algorithms [that allowed] the web to scale."

Photo: Henry Thomas

Tim Berners-Lee wins $1 million Turing Award

CSAIL researcher honored for inventing the web and developing the protocols that spurred its global use.

Adam Conner-Simons | CSAIL
April 4, 2017

Press Inquiries

RELATED

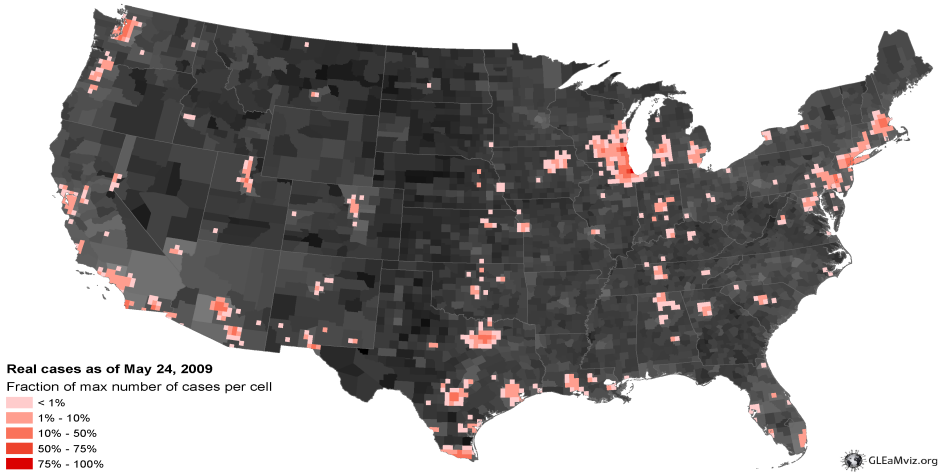
Networks: Economic Impact

licensed under CC Attribution-NonCommercial-ShareAlike 2.0 Germany | Ludwig Gatzke | http://flickr.com/photos/stabilo-boss/

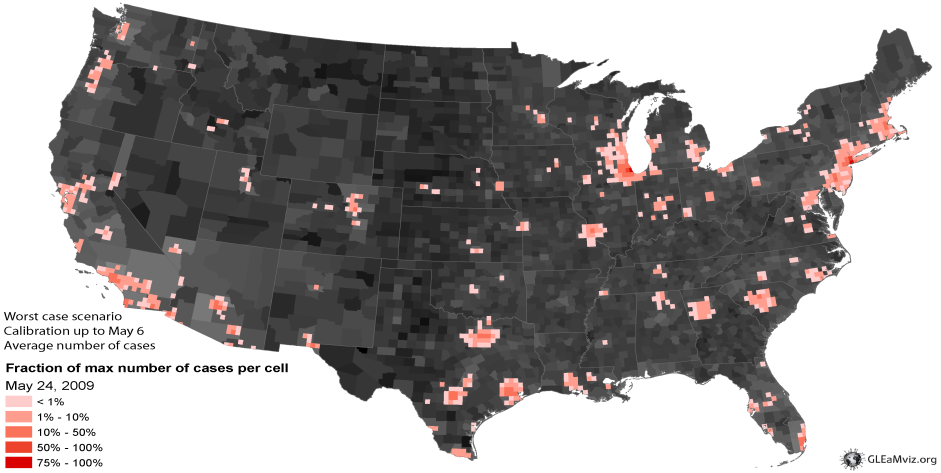
- **Google**
  Market cap: $394 billion
  (1y ago it was 300b)
- **Cisco**
  Market cap: $130 billion
  (1y ago it was 100b)
- **Facebook**
  Market cap: $201 billion
  (1y ago it was 114)

Networks: Healthcare Impact

Predicting epidemics (e.g., the 2009 H1N1 pandemic)



Real



Predicted

What Can we do With Networks?

- **Communication networks**
  - Intrusion detection, fraud detection
  - Churn prediction (e.g., telecommunication providers)

- **Social networks**
  - Link prediction, friend recommendation
    - E.g., Facebook, LinkedIn
  - Social circle detection, community detection
  - Social recommendations
  - Identifying influential nodes, information spreading, influence

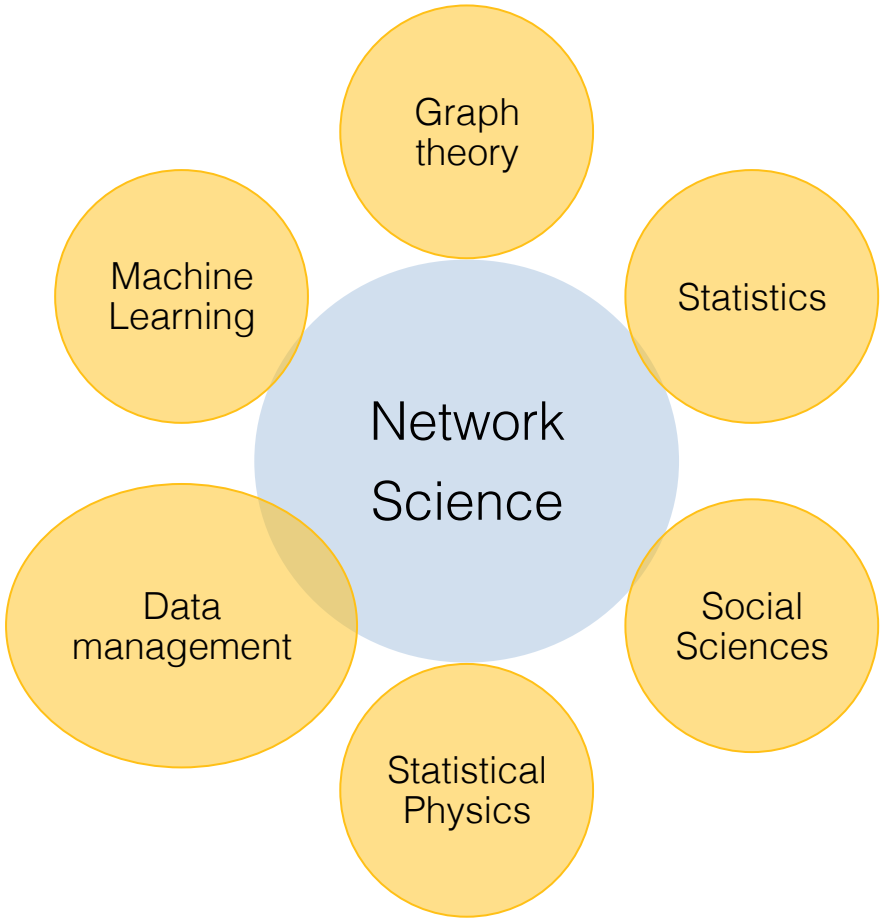
- **Information networks**
  - Navigational aids

Networks Really Matter

- If you want to understand the spread of diseases, can you do it without social networks?
- If you want to understand the structure of the Web, it is hopeless without working with the Web's topology
- If you want to understand dissemination of news, it is hopeless without considering the information networks (e.g., weblogs)
- If you are interested to suggest new connections in LinkedIn, you should take into account the underlying social graph

Network Science Analytics

Discovering, analyzing and making sense of graph data



About this course

Reasoning about Networks (1/2)

- **What do we hope to achieve from studying networks?**
  - Patterns and statistical properties of network data
  - Design principles and models
  - Understand why real networks are organized the way they are
    - Predict behavior of networked systems
  - Utilize the extracted knowledge in practical applications

Reasoning about Networks (2/2)

- **How do we reason about networks?**
  - Empirical analysis: Study network data to find organizational principles
    - How do we measure and quantify networks?
  - Mathematical models: Graph theory and statistical models
    - Models allow us to understand behaviors and distinguish surprising from expected phenomena
  - Algorithms for analyzing graphs
    - Hard computational challenges (scale and complexity of the underlying networks)
    - Unsupervised vs. supervised algorithms

How It All Fits Together – Sample of Topics

Properties	Models	Algorithms
Small diameter, Edge clustering	Small-world model, Erdös-Renyi model	Decentralized search
Scale-free	Preferential attachment, Copying model	PageRank, Hubs and authorities, Anomaly detection
Strength of weak ties, Core-periphery	Kronecker Graphs	Community detection
Densification power law, Shrinking diameters	Microscopic model of evolving networks	Link prediction, Supervised random walks
Information virality, Memetracking	Independent cascade model, SIR, SIS	Detection of influential nodes, Influence maximization

Learning Objectives

- Introduce students to the field of graph mining and network analysis
  - Cover a wide range of topics, methodologies and related applications
  - Hands-on experience on dealing with graph data and graph mining tasks

- By the end of the course, we expect
  - To have a thorough understanding of various graph mining and learning tasks
  - Be able to analyze large-scale graph data
  - Formulate and solve problems that involve graph structures

Course Logistics

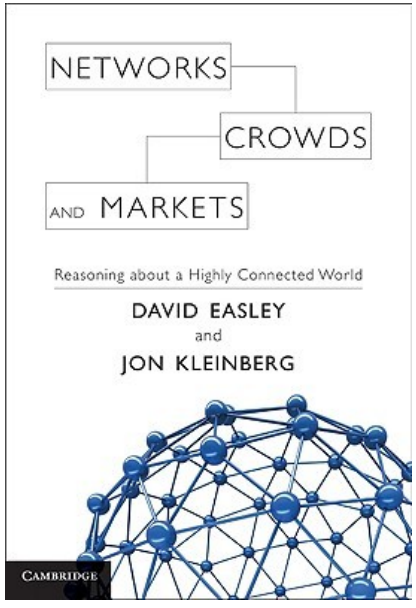
- Website
  - http://fragkiskos.me/teaching/NGSA-S18/
  - Information about the course, schedule, reading material
  - Resources (helpful for the project)

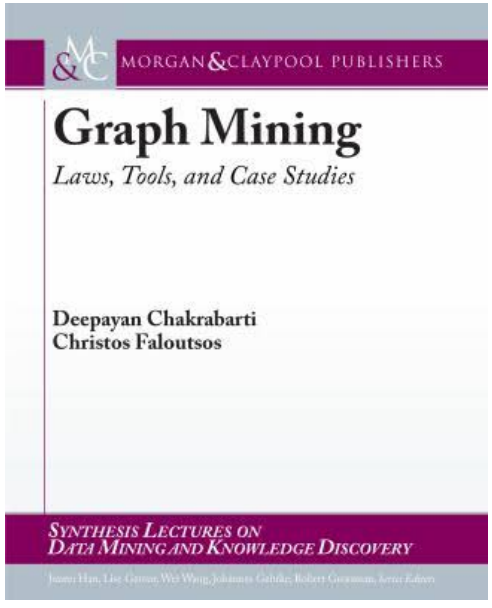
- Piazza for Q&A
  - http://piazza.com/centralesupelec/spring2018/ngsa
  - Please, participate and help each other
  - All announcements will be posted there
  - Also, lecture slides and assignments
  - Use key to enroll: ngsa-2018

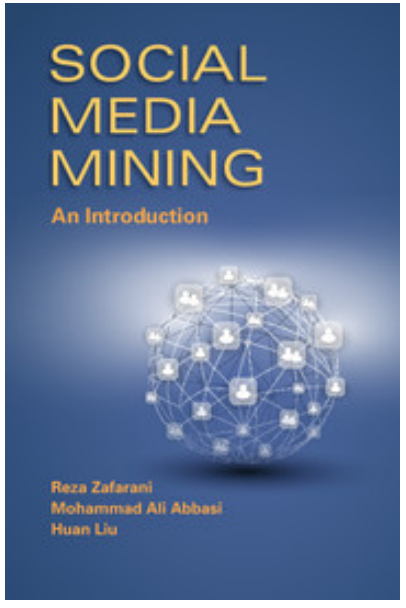
Prerequisites

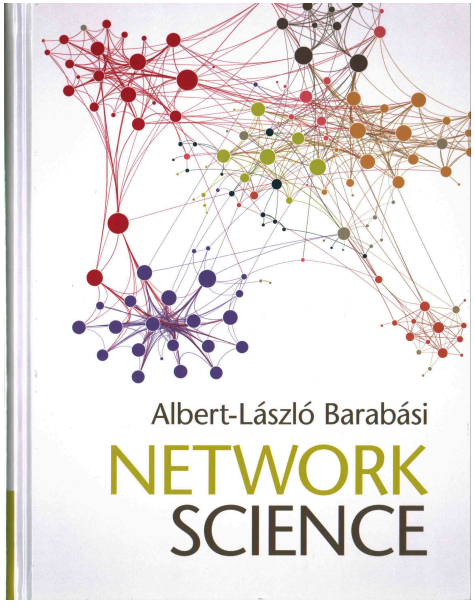
- Basic knowledge of
  - Graph theory and algorithms
  - Linear algebra
  - Probabilities and statistics
- Be familiar with basic tasks in machine learning / data mining
  - E.g., Clustering, classification
- Programming is necessary
  - Python (or any other language of your preference)

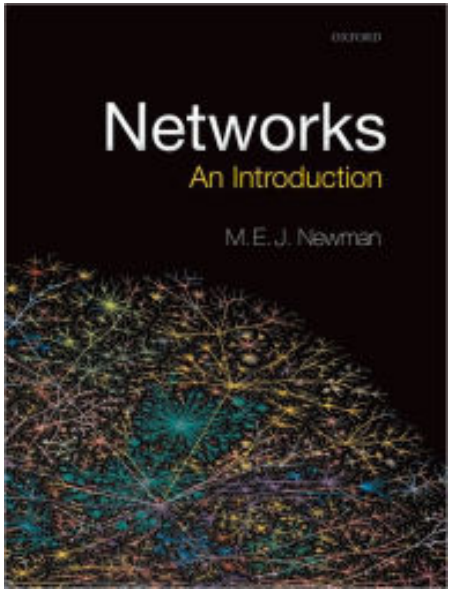
Reading Material











- The books are publicly available in electronic form (except the "Networks: An Introduction)
  - Pointers to chapters for every lecture (see the website)
- Research articles for some topics

Coursework and Grading

	Weight	Details
Assignment 1 (individually)	20%	• Many short questions • E.g., analyze a network, compute some properties, etc. • Theoretical and programming questions
Assignment 2 (teams of 3-4 students)	30%	• Deal with a real network science task • Kaggle competition • Deliver short report and code
Project (teams of 3-4 students)	50%	• Proposal (5%) • Final report + poster presentation (45%)

- Small adjustments may be done in the weights of the coursework
- A detailed description of the project will be provided soon

Course Project (1/2)

- Main component of the evaluation of the course
  - Experimental evaluation of algorithms on an interesting graph dataset
  - Empirical comparison of algorithms for a specific task
  - A theoretical project that considers an algorithm / model and derives rigorous results about it
  - A new algorithm for a graph mining task and evaluation
  - Formulation of an interesting problem using graph mining techniques, algorithm and evaluation
  - New network dataset, exploration and an application (make the dataset available)
  - Efficient implementation of an algorithm and experimental evaluation
  - Your ideas along the lines of your own research
- Ideally, combination of experimentation on real/artificial data and theoretical analysis
- The final report should have the structure of a research article

Course Project (2/2)

- Please see the project section on piazza [will be added soon]
- Check the Resources section on the website for ideas about interesting datasets, tasks, data challenges
  - Start thinking about the project as early as possible
  - We will help with ideas and mentoring
- Poster session open to other students and faculty

It will be fan ☺

Software Tools

- We strongly advise to use Python
  - NetworkX library
  - igraph library (also for C++ and R)
- Snap library
  - C++ and Python
- Gephi, Jung and graph-tool for network visualization
- See the Resources section

Some Personal Notes ☺

- Come to class
  - Please ask questions, participate in discussions on piazza
- Check out the additional suggested material on the website
  - Special topics, research-oriented course
  - Search the web, google is your friend!
  - For every topic covered in the class, you can find the original research articles – take a look on them
  - Typically, the suggested reading material is overlapping – read selectively
- Play with software tools
  - This is the actual goal of the assignments
- Give us your feedback!

Topics that will be covered

Schedule (Subject to Change)

http://fragkiskos.me/teaching/NGSA-S18/

Schedule and Lectures

The topics of the lectures are subject to change (the following schedule outlines the topics that will be covered in the course). The slides for each lecture will be posted in piazza just before the start of the class. **The due dates of the assignments/project are subject to change.**

Week	Date	Topic	Material	Assignments/Project
1	January 19	○ Introduction to network science and graph mining ○ Graph theory and linear algebra recap; basic network properties	Lecture 1A Lecture 1B	
2	January 26	○ Random graphs and the small-world phenomenon ○ Power-law degree distribution and the Preferential Attachment model		
3	February 2	○ Time-evolving graphs and network models ○ Centrality criteria and link analysis algorithms		**Assignment 1 out**
4	February 9	○ Graph clustering and community detection		**Project proposal due on February 10**
5	March 2	○ Node similarity and link prediction ○ Graph similarity and graph classification		**Assignment 1 due on February 25** **Assignment 2 out**
6	March 9	○ Representation learning in graphs ○ Graph sampling and summarization		
7	March 16	○ Epidemic processes and cascading behavior in networks ○ Influence maximization in social networks		
8	March 23	○ Core decomposition in networks ○ Graph-based methods in NLP		**Assignment 2 due on March 25**
9	April 6	Project presentations		**Project final report due** **Project poster session or presentations**

Networks or Graphs?

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

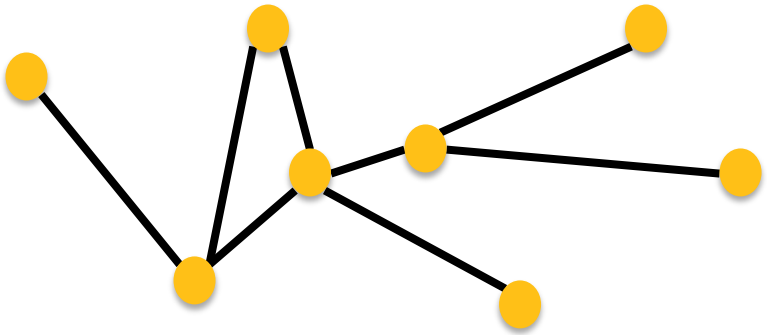
  Language: network, node, link

- Graph is mathematical representation of a network (a model)
  - 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

Basics in Graph Theory and Linear Algebra



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Adjacency matrix

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

Graph-theoretic algorithms

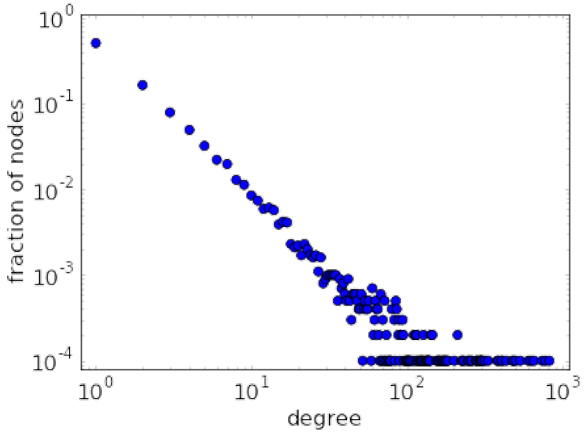
- E.g., graph concepts, types of graphs, subgraphs, traversal, shortest paths, connectivity, complexity issues

Linear algebra

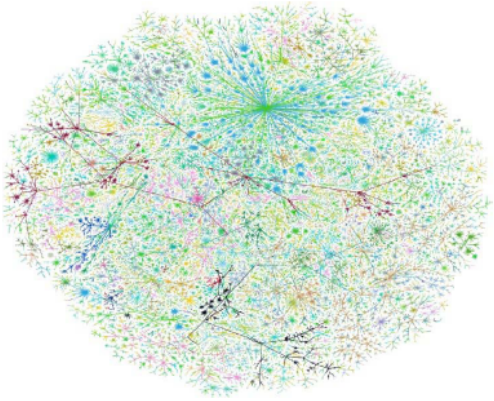
- E.g., matrix-based graph representations, matrix decomposition, properties of the adjacency matrix, Laplacian matrix, spectral graph theory, graph spectrum

Patterns and Graph Generative Models

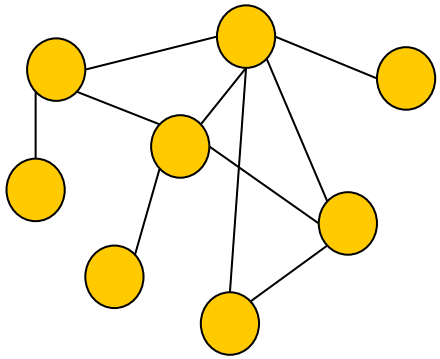
Q1: How does a real-network look like?
Q2: Properties, patterns, deviation from randomness?



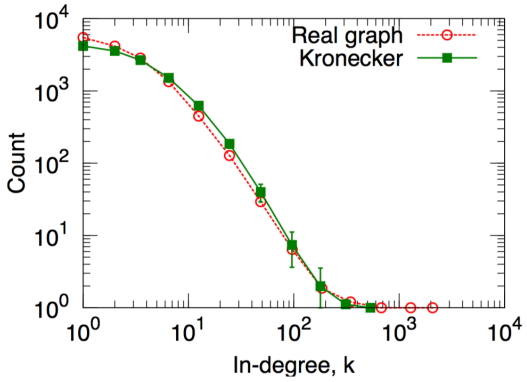
Q3: Can we generate artificial networks that are similar to real ones?



Given a real graph
(e.g., Facebook social graph)

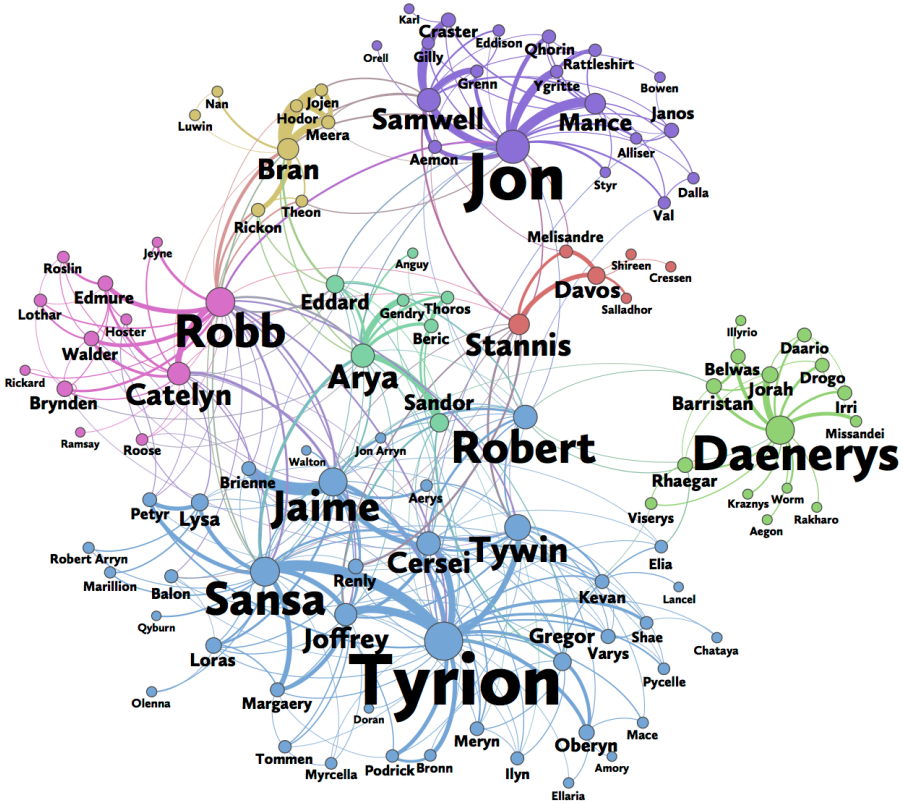


Generate an artificial graph
(e.g., model the formation process)



Fit some properties
(e.g., same degree distribution)

Centrality and Ranking in Networks

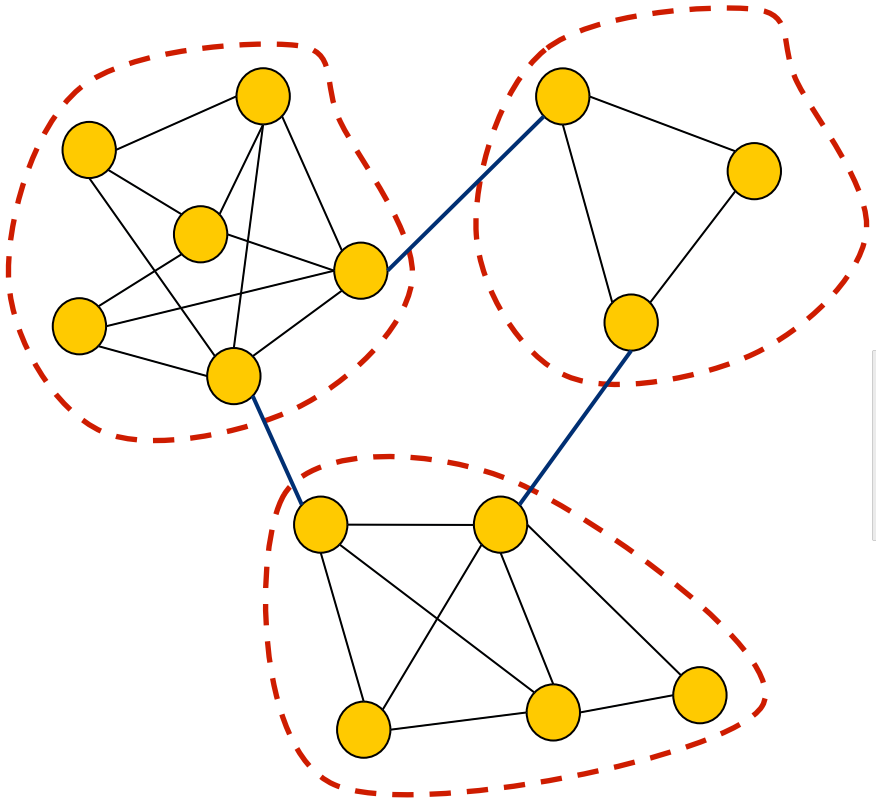


Q: How to determine the importance of a node in the graph?

- Centrality criteria (e.g., degree, closeness, betweenness)
- HITS and PageRank algorithms
- Scalability issues

Source: https://www.macalester.edu/~abeverid/thrones.html

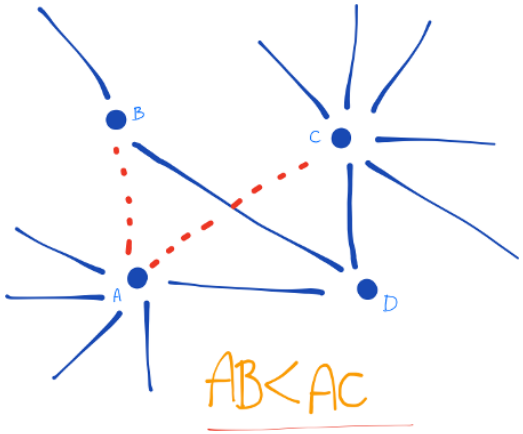
Graph Clustering - Community Detection



Example graph with three communities

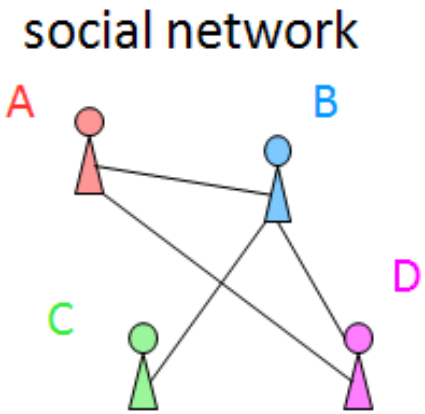
- [Models and definitions] How a community in graphs looks like?
- [Algorithms] How can we extract the inherent communities?
- [Patterns in large networks] What is happening in large-scale networks?

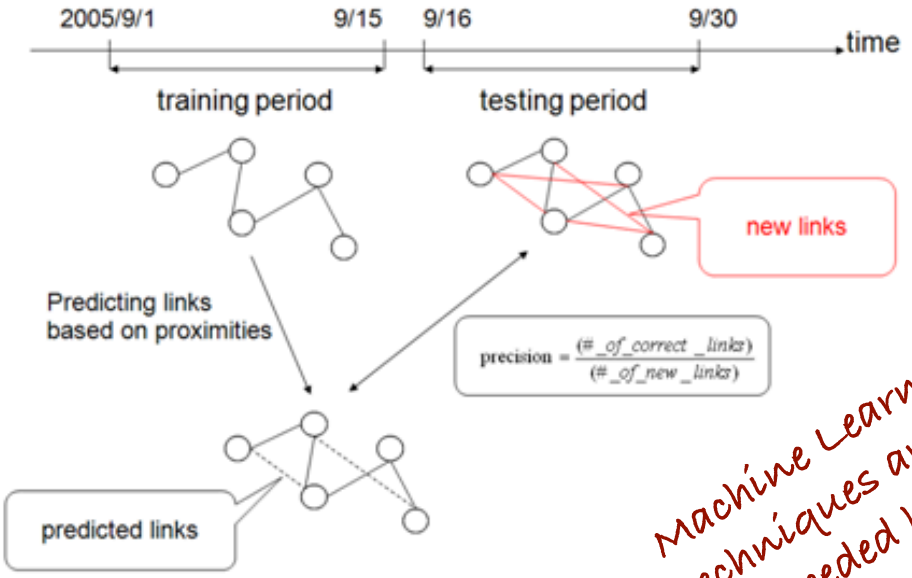
Node Similarity and Link Prediction



- How similar are two nodes in the graph?
- Can we utilize this similarity to predict missing edges (links)?

Applications?



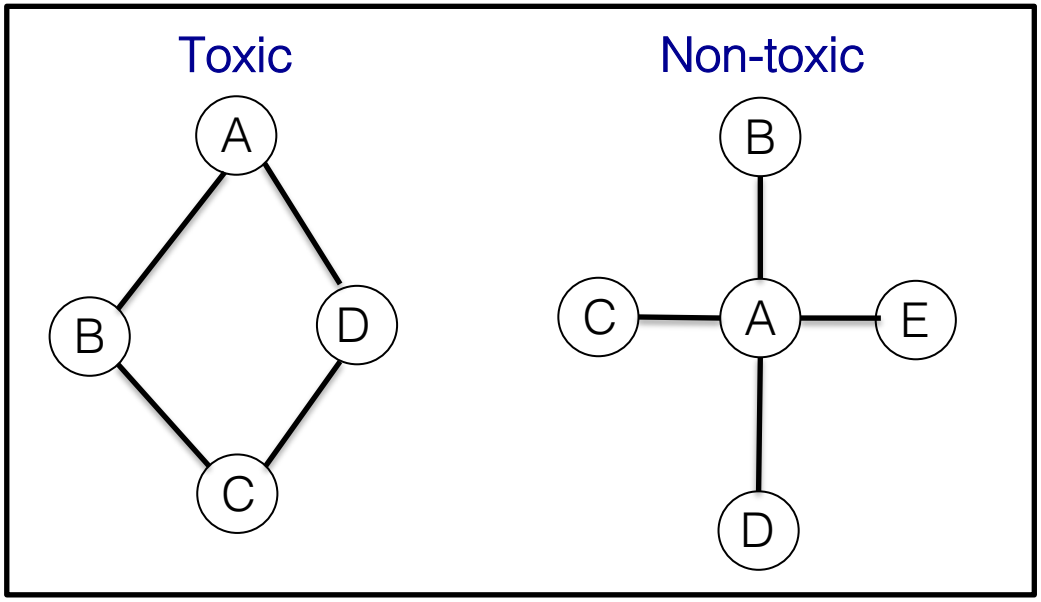


Machine Learning techniques are also needed here!

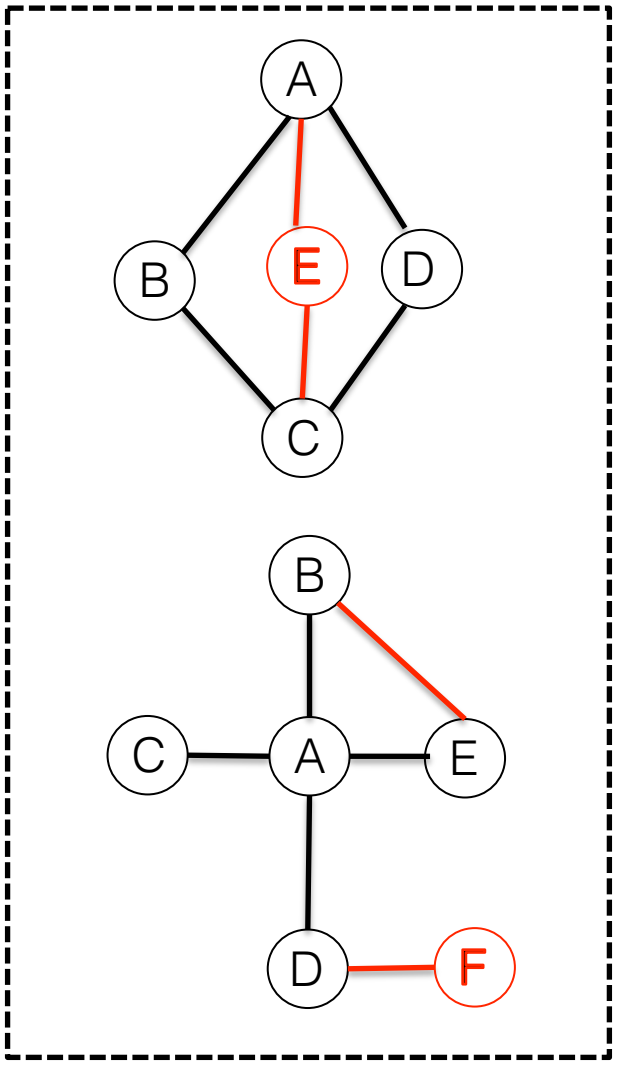
Source: http://be.amazd.com/link-prediction/
http://www.net.c.titech.ac.jp/research.html

Graph Similarity and Classification

Dataset of known molecules

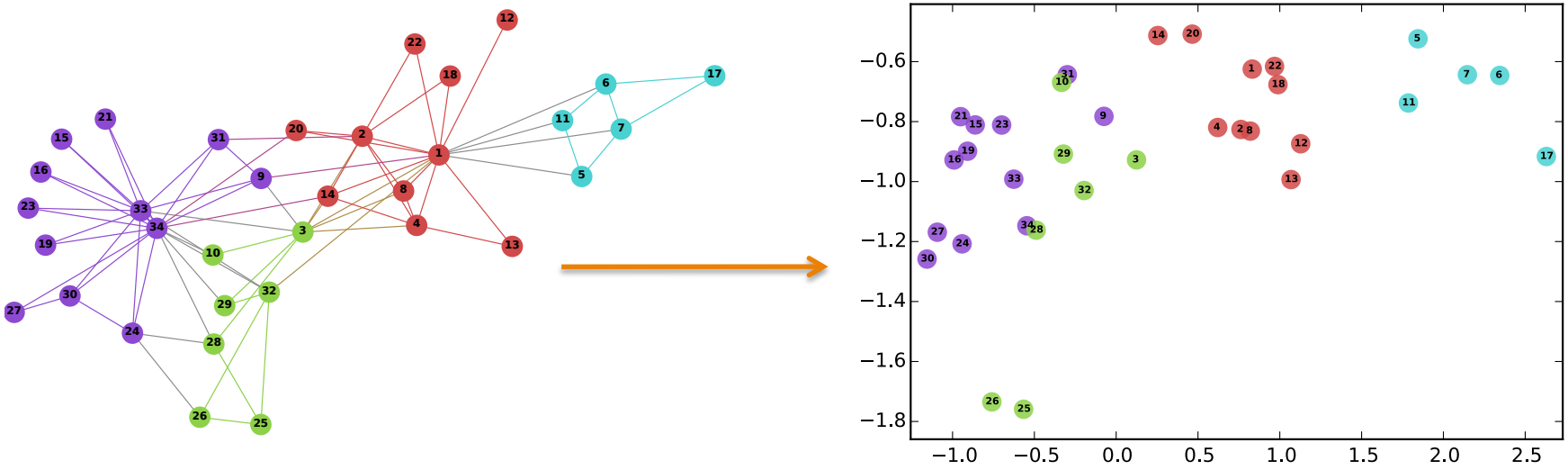


Unknown molecules



Task:
Given a set of molecules that are either toxic or non-toxic
Predict the class of unknown molecules

Representation Learning in Graphs



Input graph

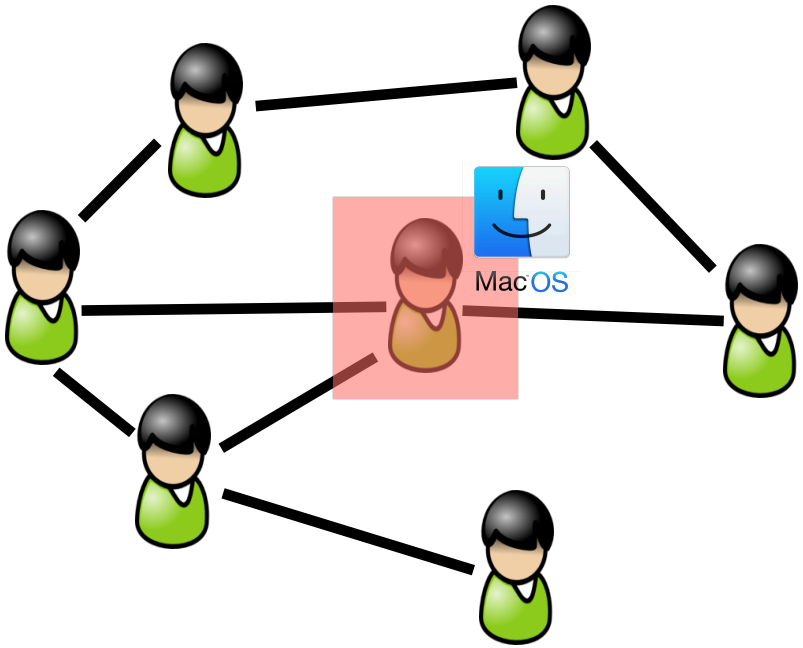
Learn latent representation

Applications: clustering, classification, link prediction, …

Also known as node embedding techniques (e.g., DeepWalk, node2vec)

[Perozzi et al., KDD '14]

Influential Nodes and Influence Maximization



[Viral marketing] How to organize an effective product promotion campaign?

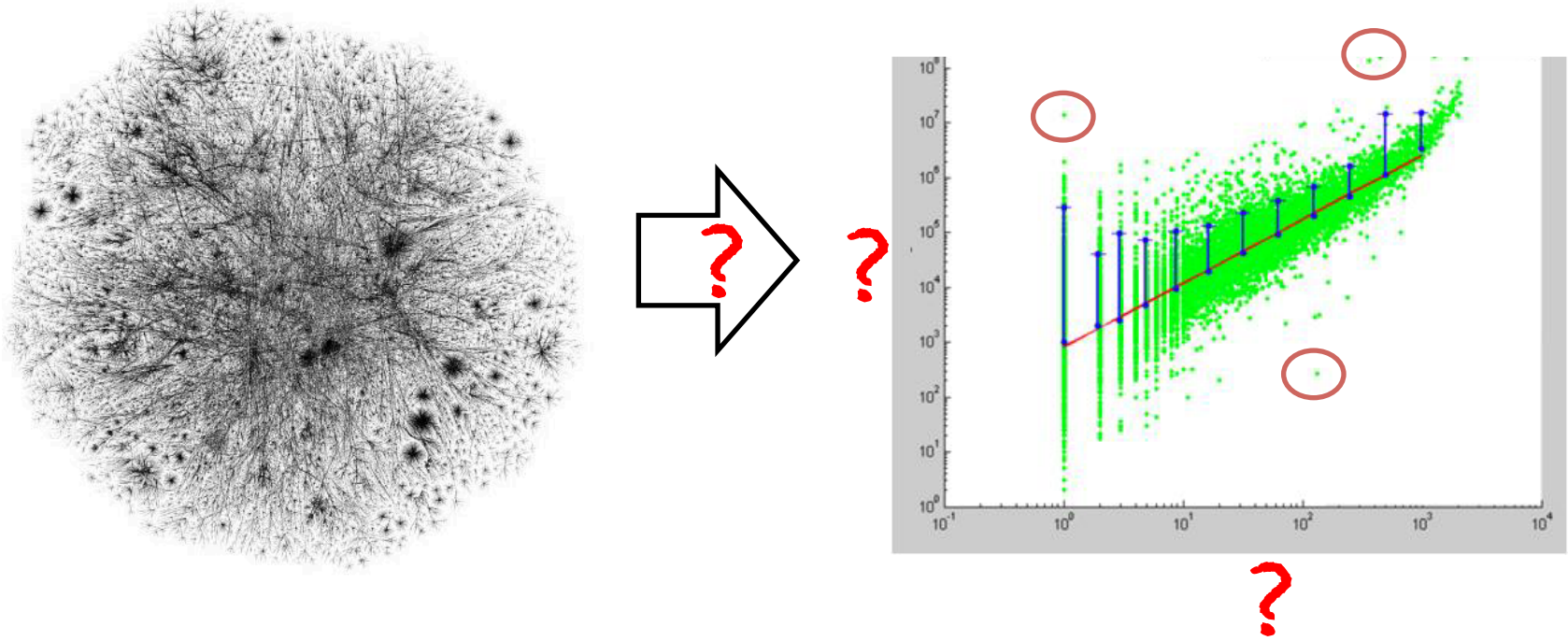
[Opinion dynamics] How do opinions/rumors spread?

[Epidemiology] How do viruses/ diseases propagate?

- Detection of influential nodes (spreaders) in networks
- Influence maximization algorithms
- Epidemic processes in networks

[Prakash, Ramakrishnan, KDD '16]

Anomaly Detection in Graphs

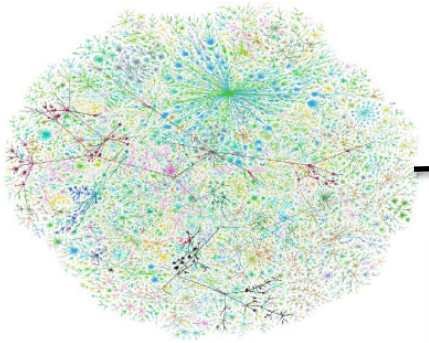


- Given a graph, how can we spot anomalies (e.g., strange, abnormal nodes)?
- Can we assign an 'anomaly' score at each node of the graph?
- Can we explain why those nodes are characterized as anomalous?

[Akoglu, Faloutsos, WSDM '13]

Graph Sampling and Summarization

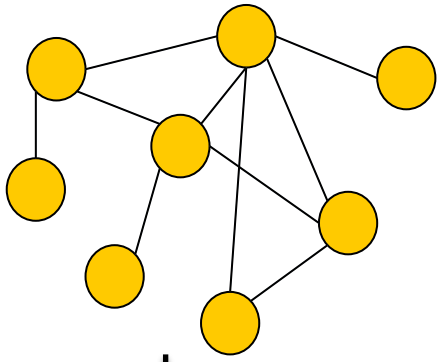
Original graph G



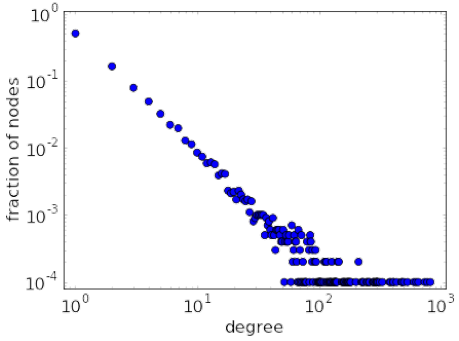
Sampling

Reduce size

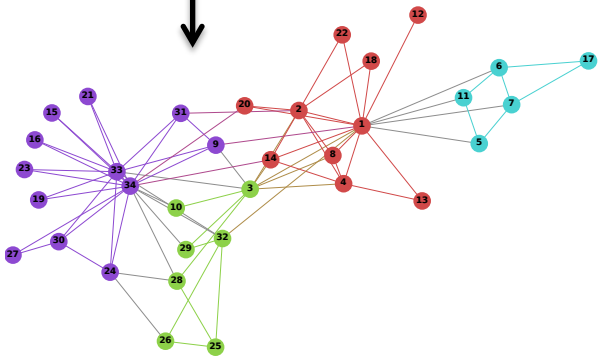
Sample subgraph G_S
(or sample of nodes/edges)



- Estimate graph properties quickly
- Extract smaller subgraph that preserves the properties of the original graph
- Algorithms run faster

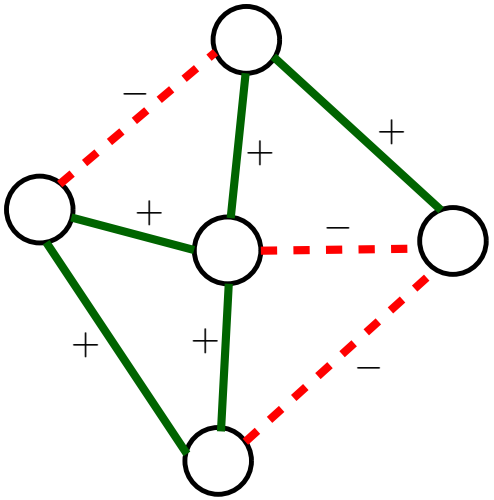


Estimate characteristics of the graph



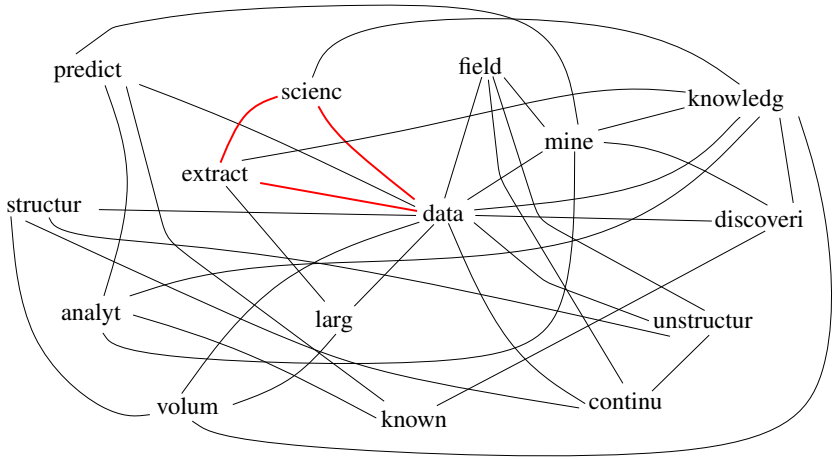
Extract the community structure

Rich Network Structures



Signed graphs

Data Science ~~is the~~ extraction of knowledge from large volumes of data that are structured or unstructured which is a continuation of the field of data mining and predictive analytics, also known as knowledge discovery and data mining.



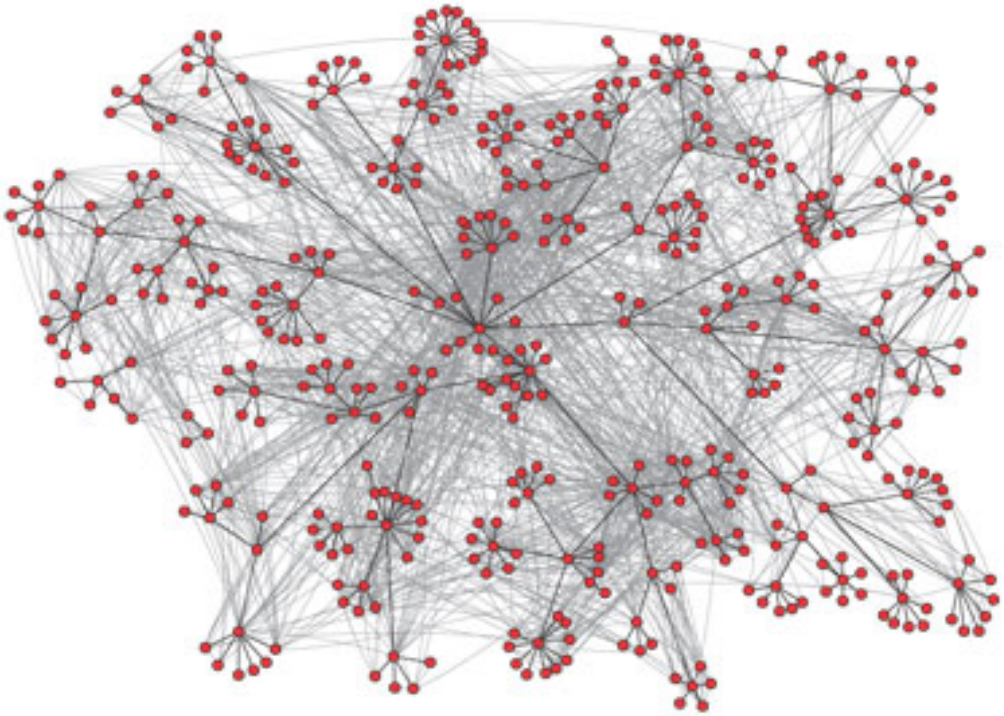
Text-based information networks

Other topics:

- Probabilistic (uncertain) graphs (probabilities on the edges)
- Multilayer networks (various types of relationships among nodes)
- Location-based systems (e.g., Foursquare)
- Biological networks

Structure of the Web Graph

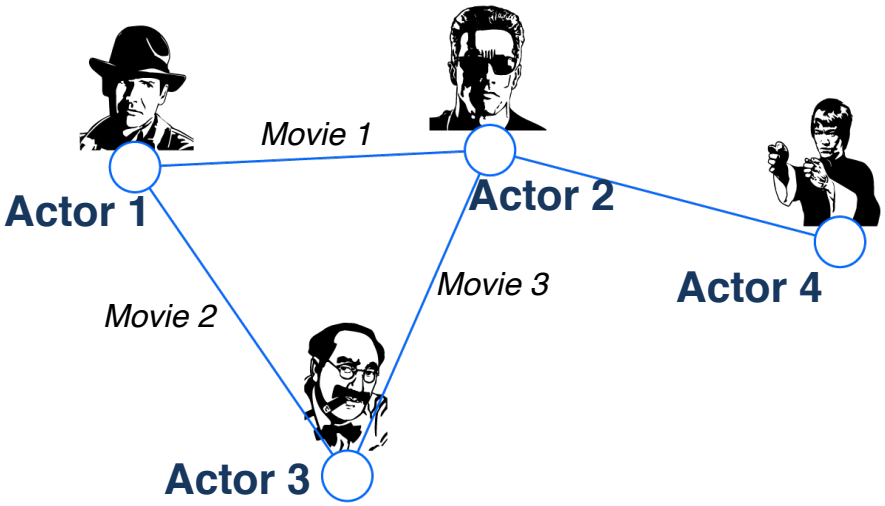
Structure of Networks?

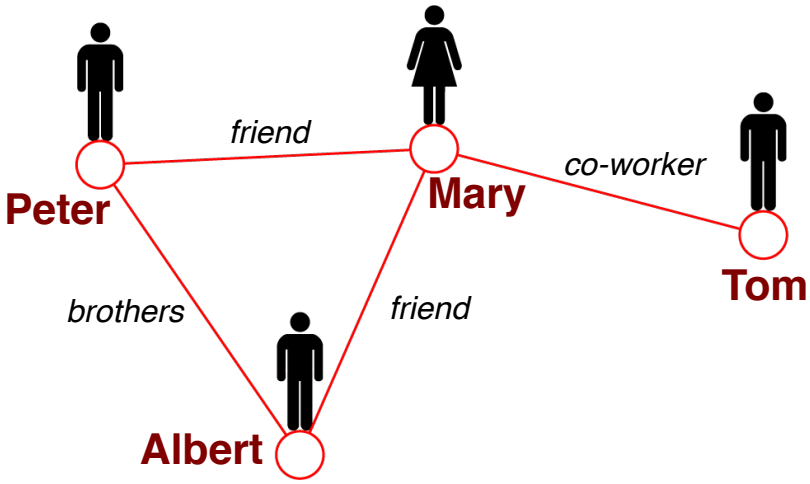


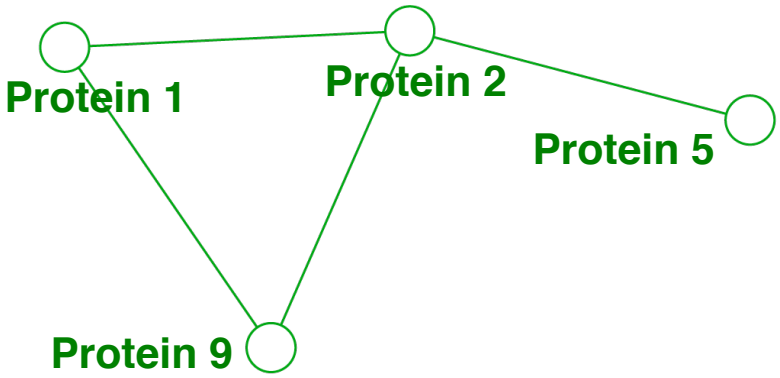
Network is a collection of objects where some pairs of objects are connected by links

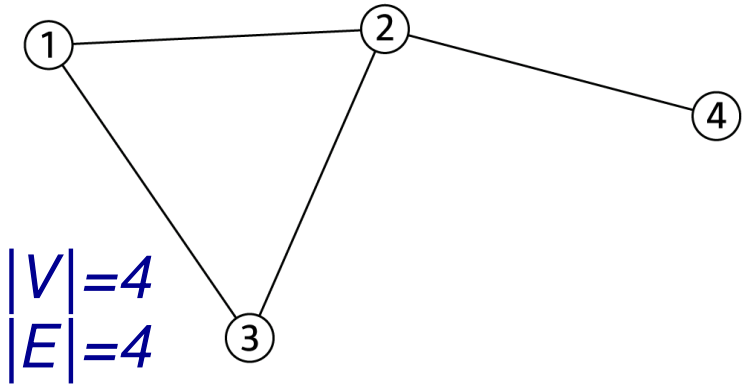
What is the structure of the network?

Networks: Common Language









Choosing Proper Representation

- How to build the graph?
  - What are the nodes?
  - What are the edges?

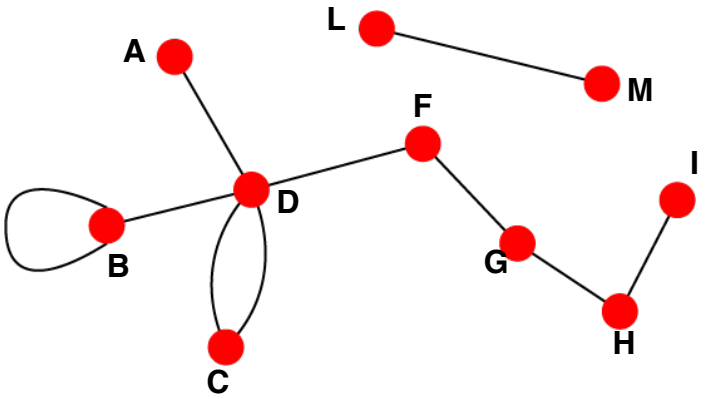
- Choice of the proper network representation of a given domain/ problem determines our ability to use networks successfully
  - In some cases there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature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

Very important step
Everything else follows!

Undirected vs. Directed Networks

Undirected

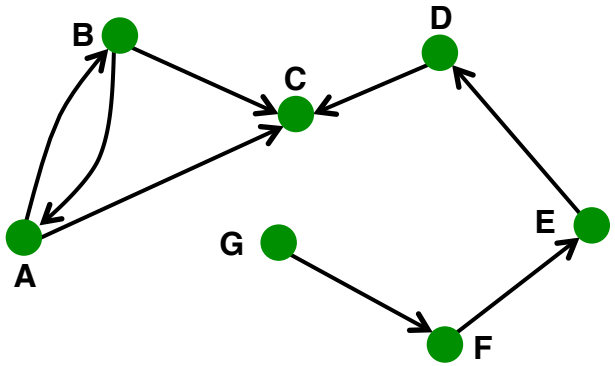
- **Links:** undirected (symmetrical, reciprocal)



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

Directed

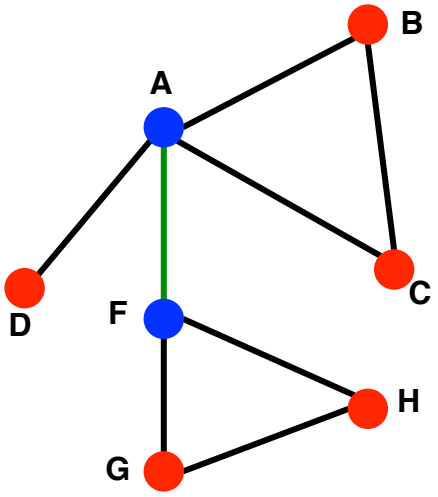
- **Links:** directed (arcs)

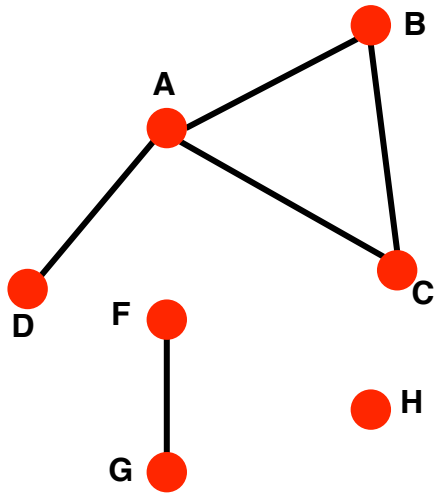


- Examples
  - Phone calls
  - Following on Twitter

Connectivity of 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





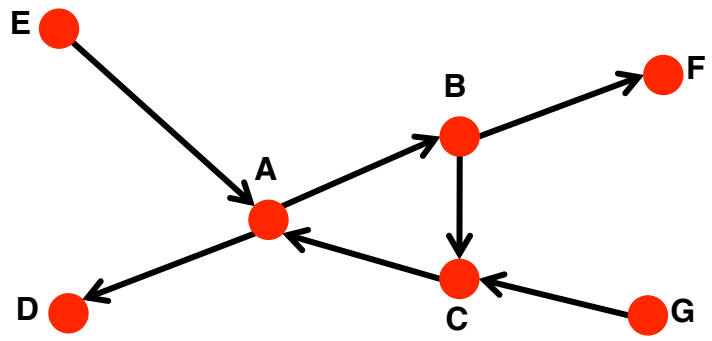
Largest Component:
Giant Component

Isolated node (node H)

Bridge edge: If we erase it, the graph becomes disconnected
Articulation point: If we erase it, the graph becomes disconnected

Connectivity of Directed Graphs

- Strongly connected
  - Has a directed path from each node *u* to every other node *v* **and** vice versa (e.g., *u*-*v* path and *v*-*u* path)
- Weakly connected
  - Is connected if we disregard the edge directions



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

Web as a Graph (1/2)

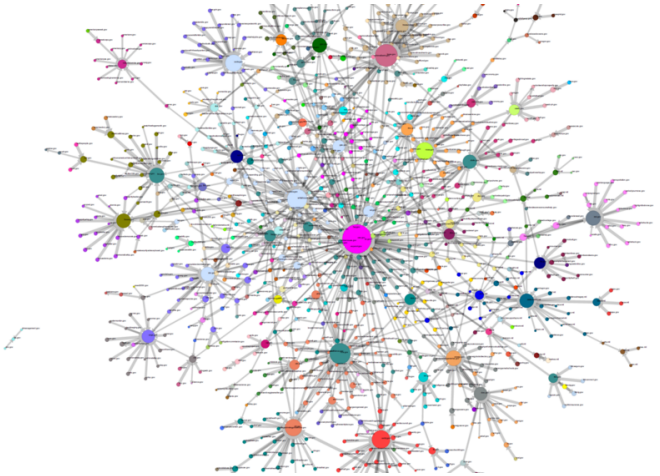
- Q: How does the Web “look like”?

- Here is what we will do next
  - We will take a real system (i.e., the Web)
  - We will represent the Web as a graph
  - We will use graph theory to reason about the structure of the graph
  - We will do a computational experiment on the Web graph
  - Learn something about the structure of the Web!

Web as a Graph (2/2)

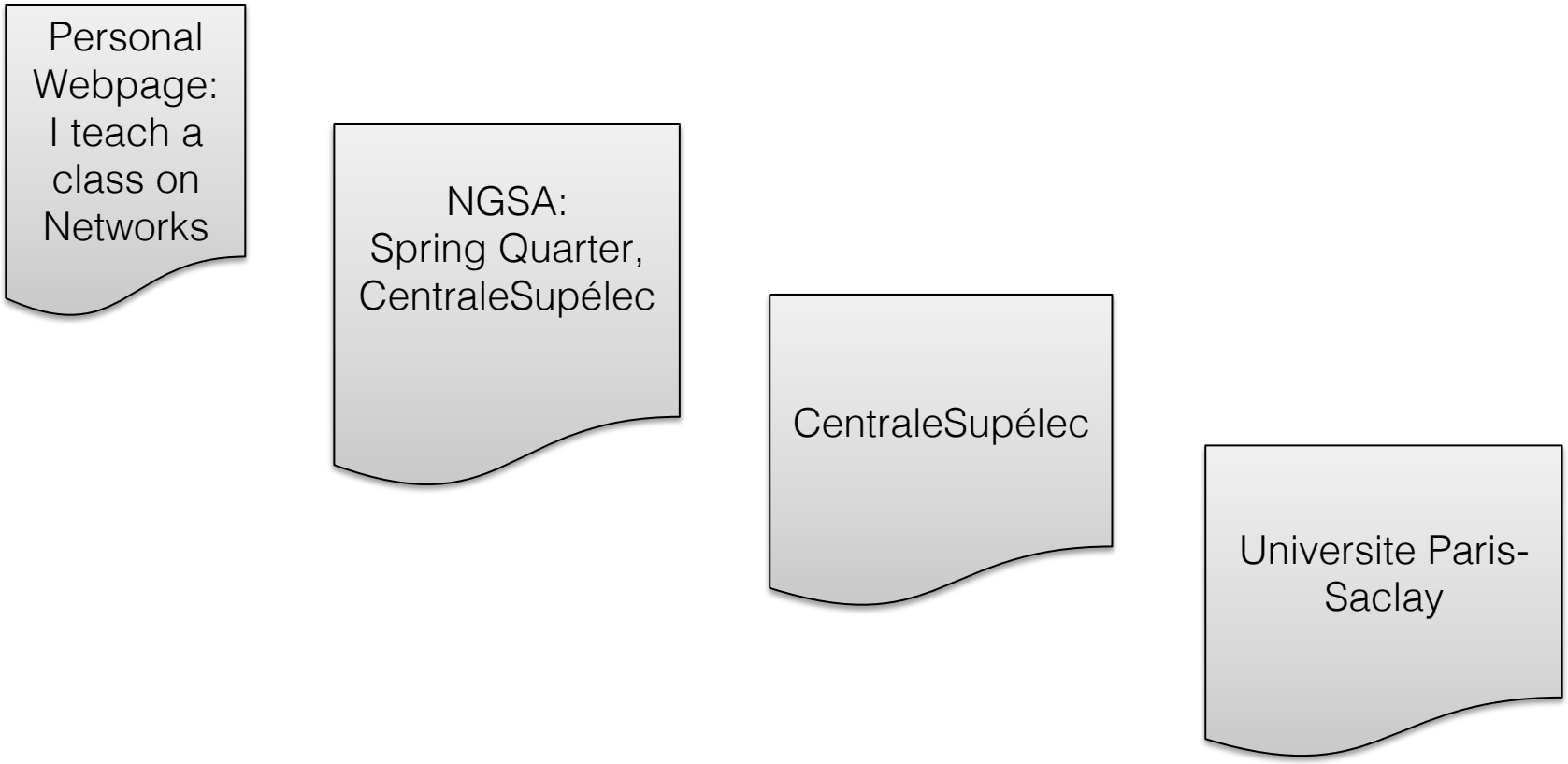
Q: What does the Web "look like" at a global level?

- Web as a graph
  - Nodes = web pages
  - Edges = hyperlinks

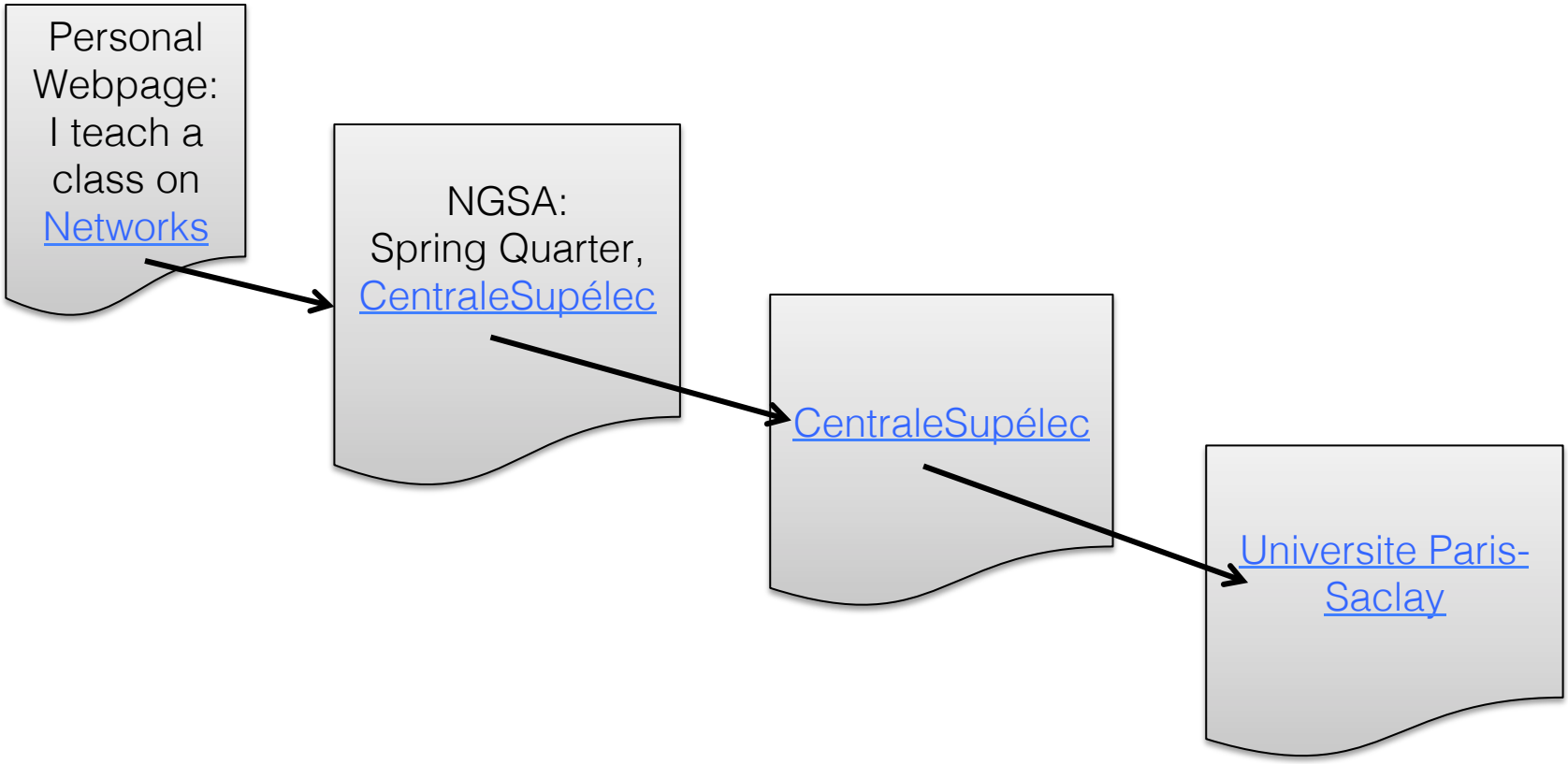


  - Side issue: What is a node?
    - Dynamic pages created on the fly
    - "dark matter" – inaccessible database generated pages

Example

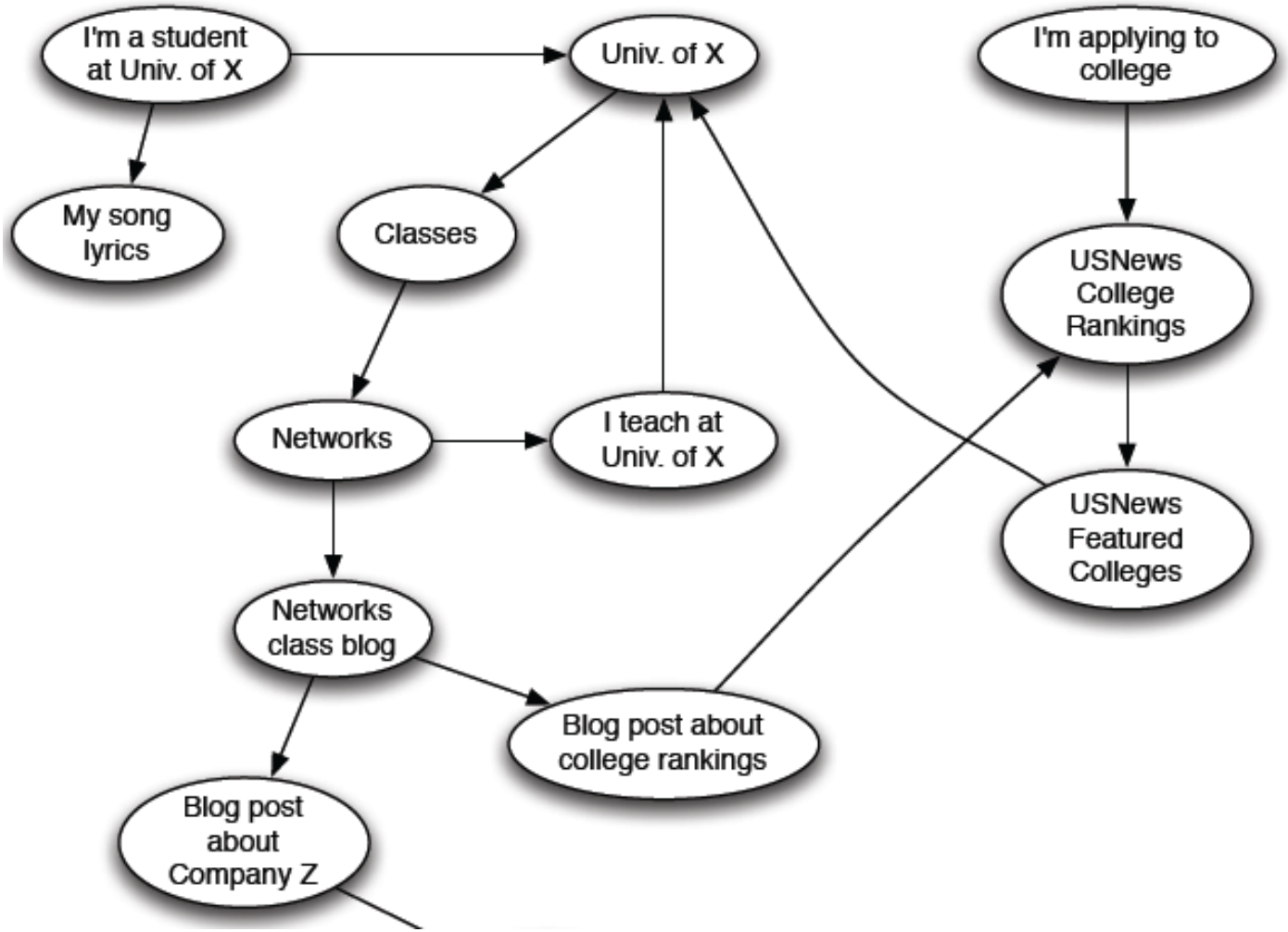

Personal Webpage: I teach a class on Networks
NGSA: Spring Quarter, CentraleSupélec
CentraleSupélec
Universite Paris-Saclay

Example



- In early days of the Web links were navigational
- Today many links are transactional

The Web as a Directed Graph

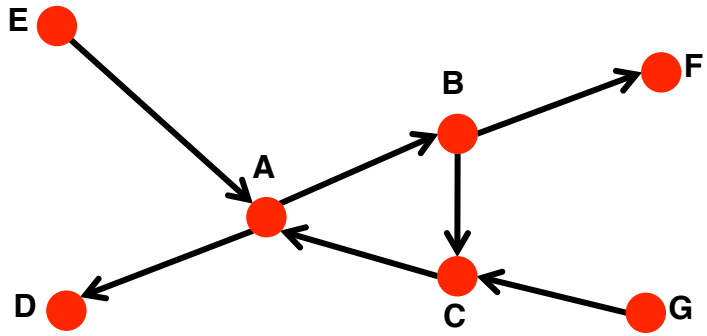


What Does the Web Look Like?

- How is the Web linked?
- What is the "map" of the Web?

Web as a directed graph **[Broder et al. 2000]**

- Given node v, what can *v* reach?
- What other nodes can reach *v* ?



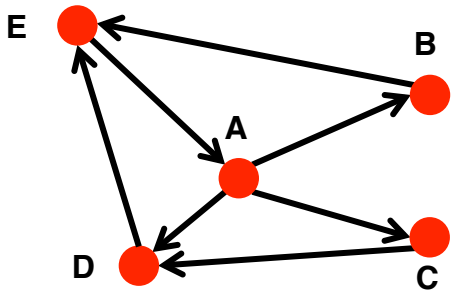
In(v) = {w | w can reach v}
Out(v) = {w | v can reach w}

For example:
In(A) = {A,B,C,E,G}
Out(A)={A,B,C,D,F}

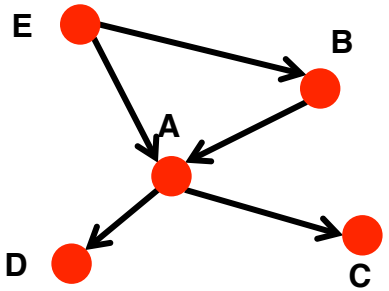
Directed Graphs

- Two types of directed graphs:
  - Strongly connected
    - Any node can reach any other node via a directed path

      In(A)=Out(A)={A,B,C,D,E}



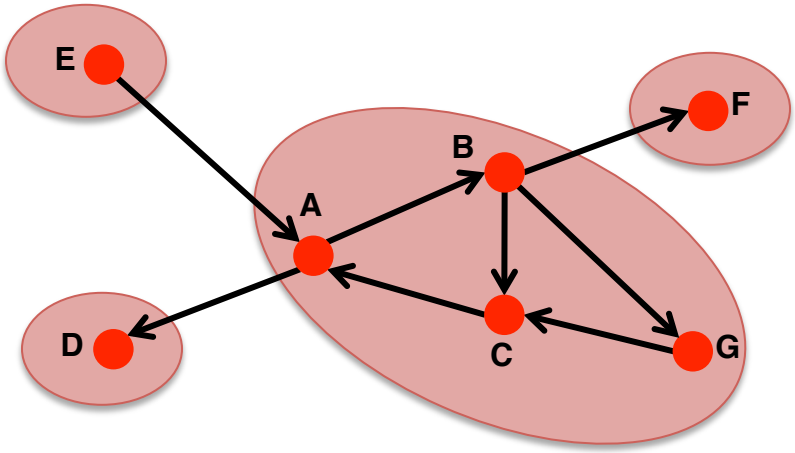
  - DAG – Directed Acyclic Graph
    - Has no cycles: if *u* can reach *v*, then *v* can not reach *u*



- Any directed graph can be expressed in terms of these two types!

Strongly Connected Component

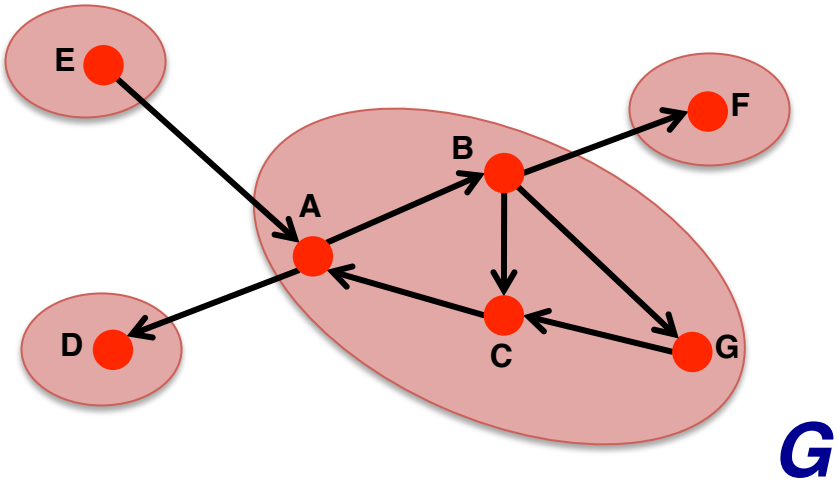
- Definition: Strongly connected component (SCC) is a set of nodes S so that:
  - Every pair of nodes in S can reach each other
  - There is no larger set containing S with this property



Strongly connected components of the graph: *{A,B,C,G}, {D}, {E}, {F}*

Strongly Connected Component

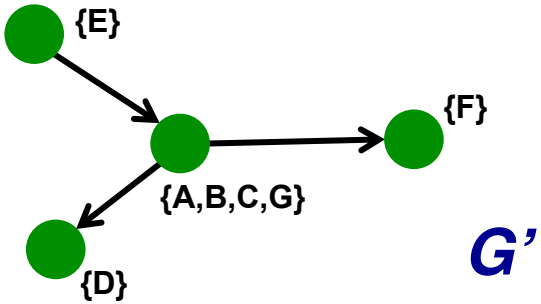
- Fact: Every directed graph is a DAG on its SCCs
  - (1) SCCs partition the nodes of *G*
    - That is, each node is in exactly one SCC
  - (2) If we build a graph *G'* whose nodes are SCCs, and with an edge between nodes of *G'* if there is an edge between corresponding SCCs in *G*, then *G'* is a DAG



G

(1) Strongly connected components of graph *G: {A,B,C,G}, {D}, {E}, {F}*

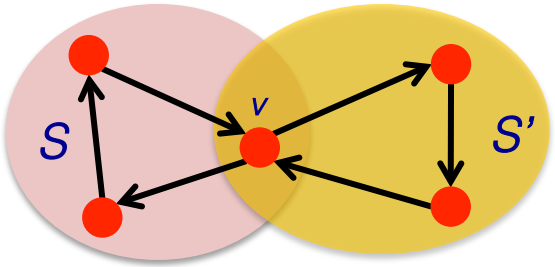
(2) G' is a DAG:



G'

Proof of (1)

- Claim: SCCs partitions nodes of *G*
  - This means: Each node is member of exactly one SCC
- Proof by contradiction:
  - Suppose there exists a node *v* which is a member of two SCCs *S* and *S'*

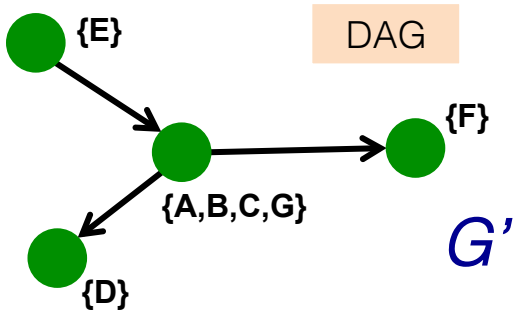


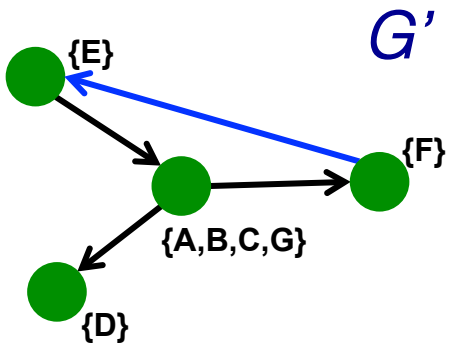
  - But then *S ∪ S'* is one large SCC!
    - Contradiction!

Proof of (2)

- Claim: *G'* (graph of SCCs) is a DAG
  - This means: *G'* has no cycles

- Proof by contradiction:
  - Assume *G'* is not a DAG
  - Then *G'* has a directed cycle
  - Now all nodes on the cycle are mutually reachable, and all are part of the same SCC
  - But then *G'* is not a graph of connections between SCCs (SCCs are defined as maximal sets)
    - Contradiction!



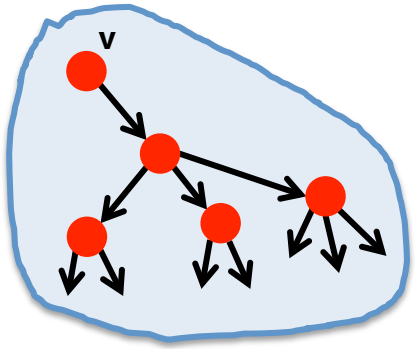


Now *{A,B,C,G,E,F}* is a SCC!

Graph Structure of the Web

- Goal: Take a large snapshot of the Web and try to understand how its SCCs "fit together" as a DAG
- Computational issue
  - Want to find a SCC containing node *v*?
  - Observation:
    - *Out(v)* … nodes that can be reached from *v*
    - SCC containing *v* is: *Out(v) ∩ In(v)*

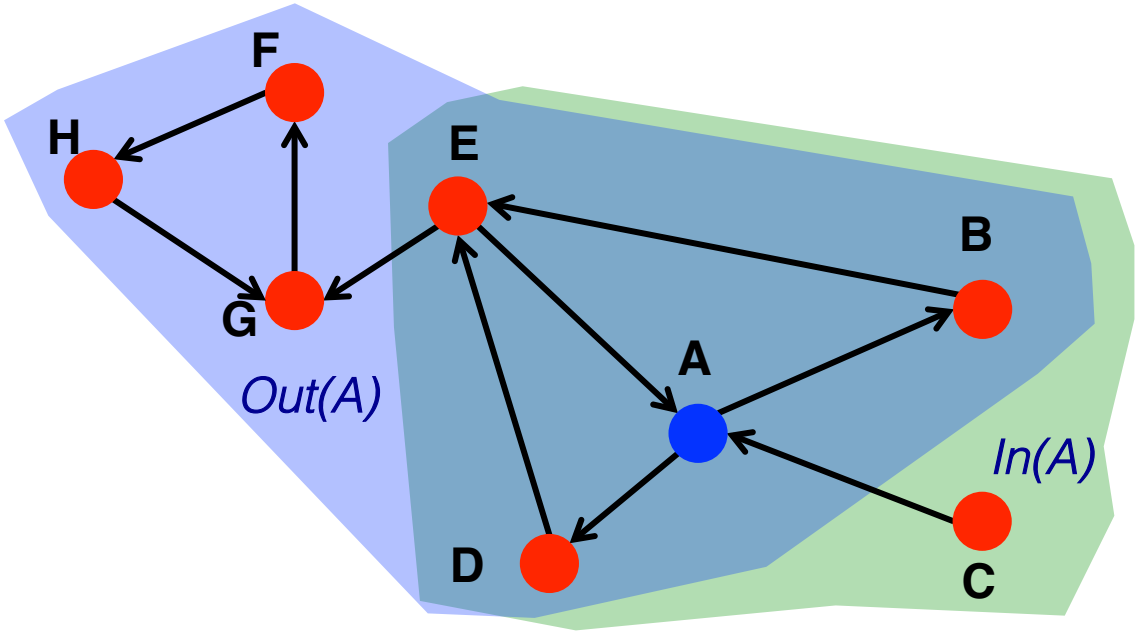
    = *Out(v,G) ∩ Out(v,G')*, where *G'* is *G* with all edge directions flipped



Out(v)

Out(A) ∩ In(A) = SCC

- Example:



  - *Out(A) = {A, B, D, E, F, G, H}*
  - *In(A) = {A, B, C, D, E}*
  - So, *SCC(A) = Out(A) ∩ In(A) = {A, B, D, E}*

Structure of the Web

- [Broder et al., 2000]:
  - Altavista crawl from October 1999
    - 203 million URLS
    - 1.5 billion links
  - Computer: Server with 12GB of memory

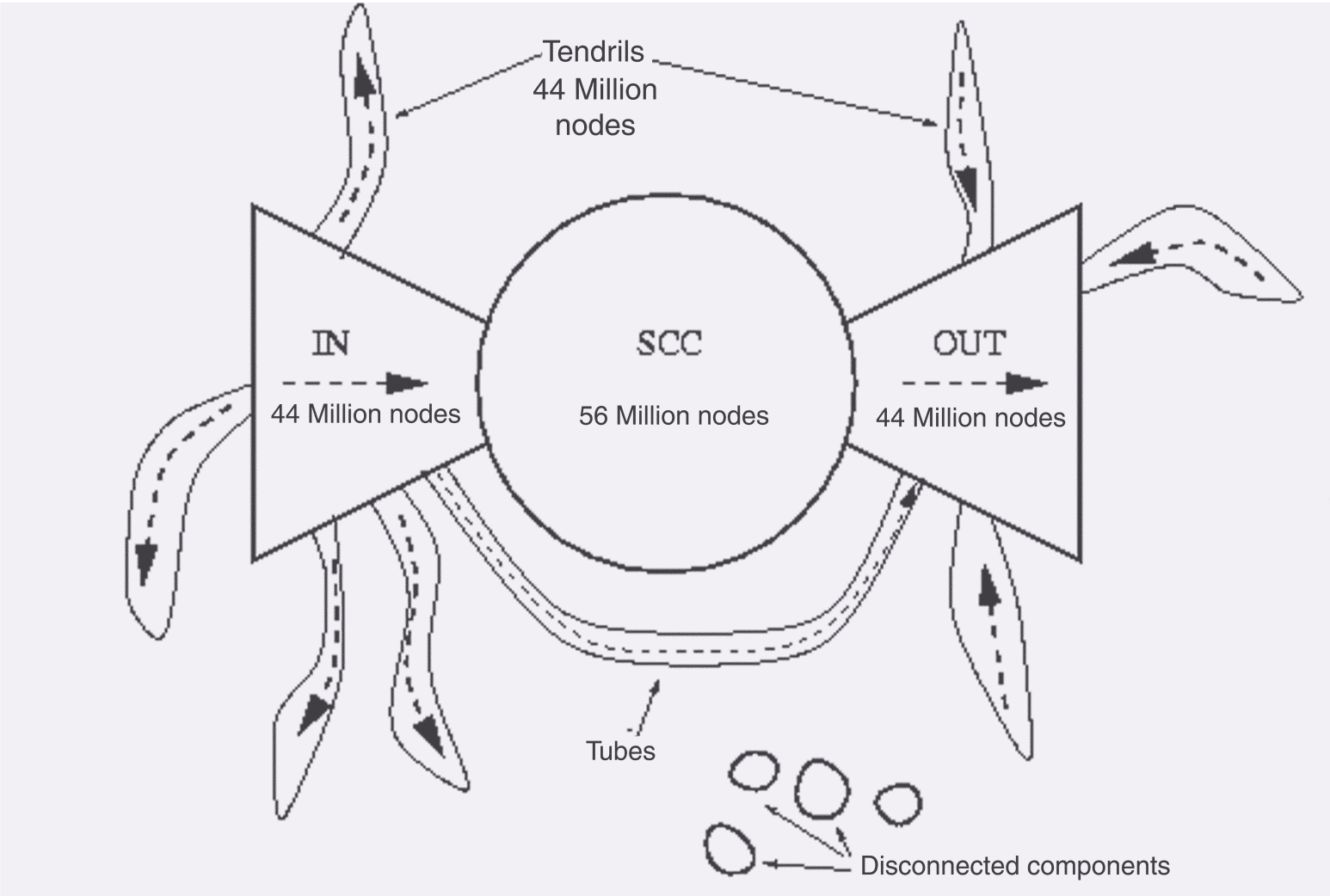
- Undirected version of the Web graph
  - 91% nodes in the largest Weakly Connected Component (WCC)

Structure of the Web

- Directed version of the Web graph
  - Largest SCC: 28% of the nodes (56 million)
  - Taking a random node *v*
    - *Out(v) ≈ 50%* (100 million)
    - *In(v) ≈ 50%* (100 million)

What does this tell us about the conceptual picture of the Web graph?

Bow-tie Structure of the Web



203 million pages, 1.5 billion links **[Broder et al. 2000]**

What did We Learn/Not Learn ?

- What did we learn:
  - Some conceptual organization of the Web (i.e., the bowtie)
- What did we not learn
  - Treats all pages as equal
    - Google's homepage == my homepage
  - What are the most important pages
    - How many pages have k in-links as a function of k?

      The degree distribution: $\sim k^{-2}$
    - Link analysis ranking - as done by search engines (e.g., PageRank)
  - Internal structure inside giant SCC
    - Clusters, implicit communities?
  - How far apart are nodes in the giant SCC:
    - Distance = # of edges in shortest path
    - Avg = 16 **[Broder et al.]**

Lecture 1 – Part B

- Basics in
  - Graph theory
  - Linear algebra and spectral graph theory

Thank You!