Network Theory and Dynamic Systems

01. Introduction to Network Science

SOSE 2025

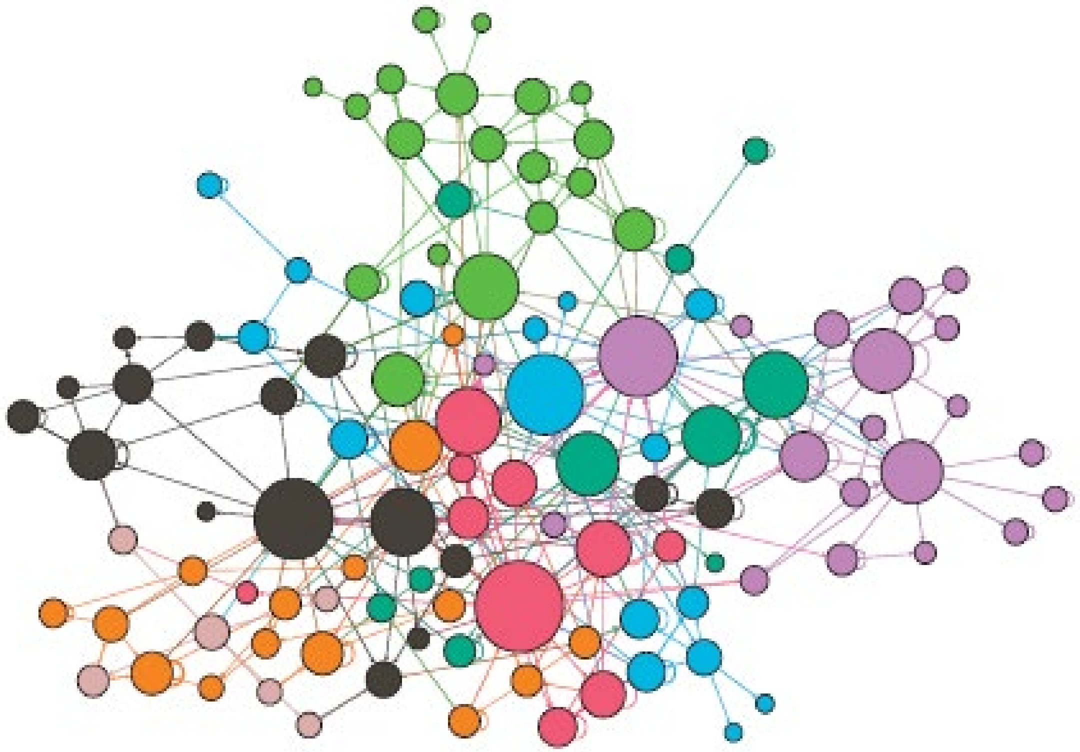
Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Networks Are Part of Everyday Life

universität koblenz
weiter:denken

- We use networks all the time — like when we chat on Facebook or Twitter, shop on Amazon, search on Google, or book flights
- Most of us don't even realize networks are working behind the scenes
- Knowing how networks work is important for many careers — in tech, marketing, business, design, biology, the arts, and more



A network map of chapters, sections, and subsections of Textbook [1], highlighting the interconnectedness through links, node colors, and sizes

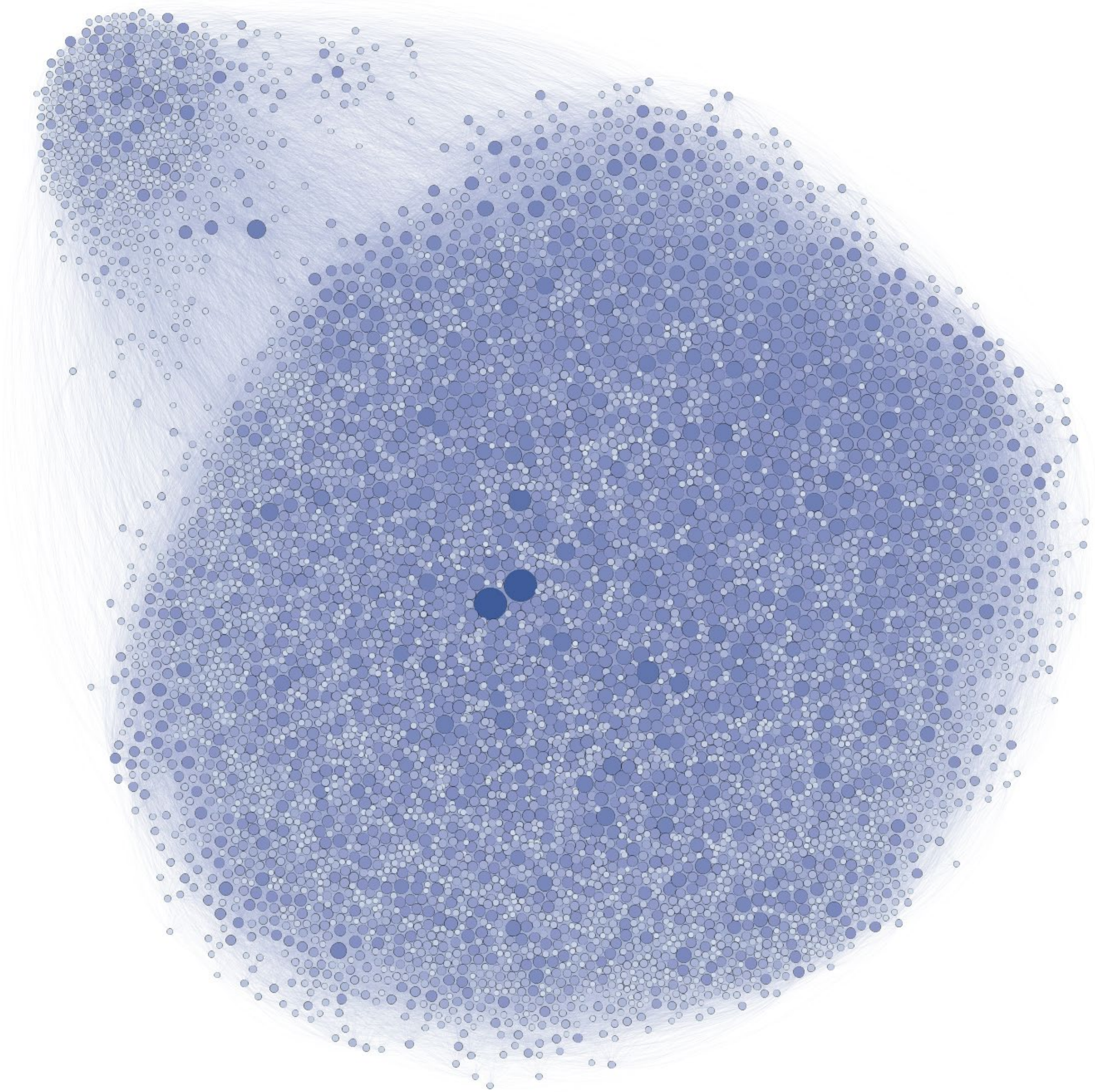
How Networks Help Us Understand Complexity

universität koblenz
weiter:denken

- Networks break down complex systems into simple parts: nodes (things) and links (connections)
- They ignore small details and focus on how things interact
- This makes networks useful in many fields to study complex systems

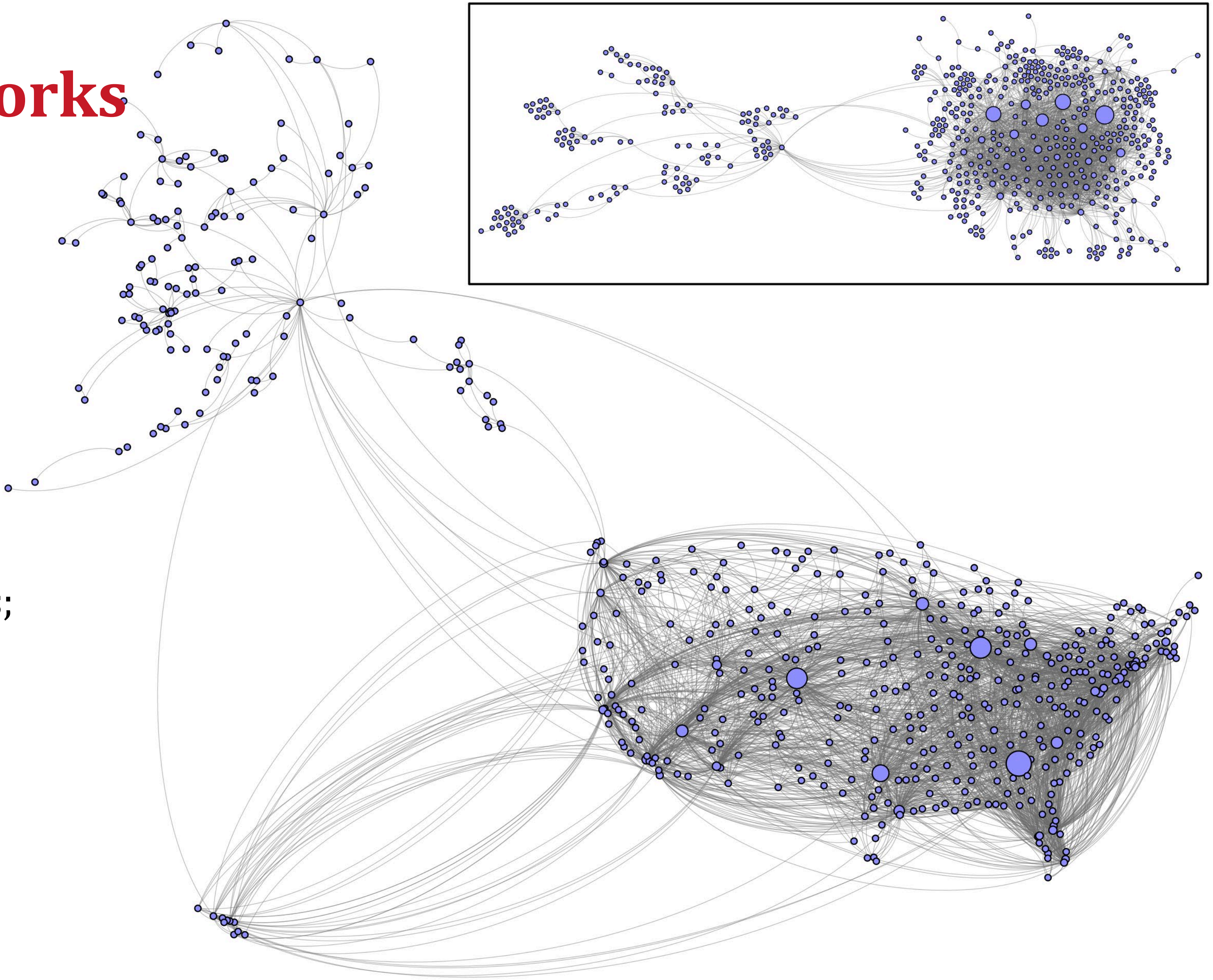
Social Networks

- What do nodes represent? People
- What do links represent? Facebook friend connections
- Do links have direction? No
- Do links have weights? No
- Larger, darker nodes have more connections; what does that represent? More connections
- What do the two clusters tell us? Undergraduate students are more likely to be friends with other undergraduates than with graduate students



Transportation Networks

- What do nodes represent? Airports/cities
- What do links represent?
- Do links have direction?
- Do links have weights?
- Larger nodes have more connections; what do they represent?
- What do the layouts represent? Air flight networks display a “hub and spoke” structure: a few hubs have huge numbers of links, while the majority of nodes have very few connections



Network Theory and Dynamic Systems

02. Network Elements

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Definitions: Network or Graph

- A **network** or **graph** *G* has two parts:
  - a set of *N* elements, called **nodes** or **vertices**, and
  - a set of *L* pairs of nodes, called **links** or **edges**
- The link *(i, j)* joins the nodes *i* and *j*
- Two nodes are **adjacent** or **connected** or **neighbors** if there is a link between them

Definitions: Undirected/Directed Networks

universität koblenz
weiter:denken

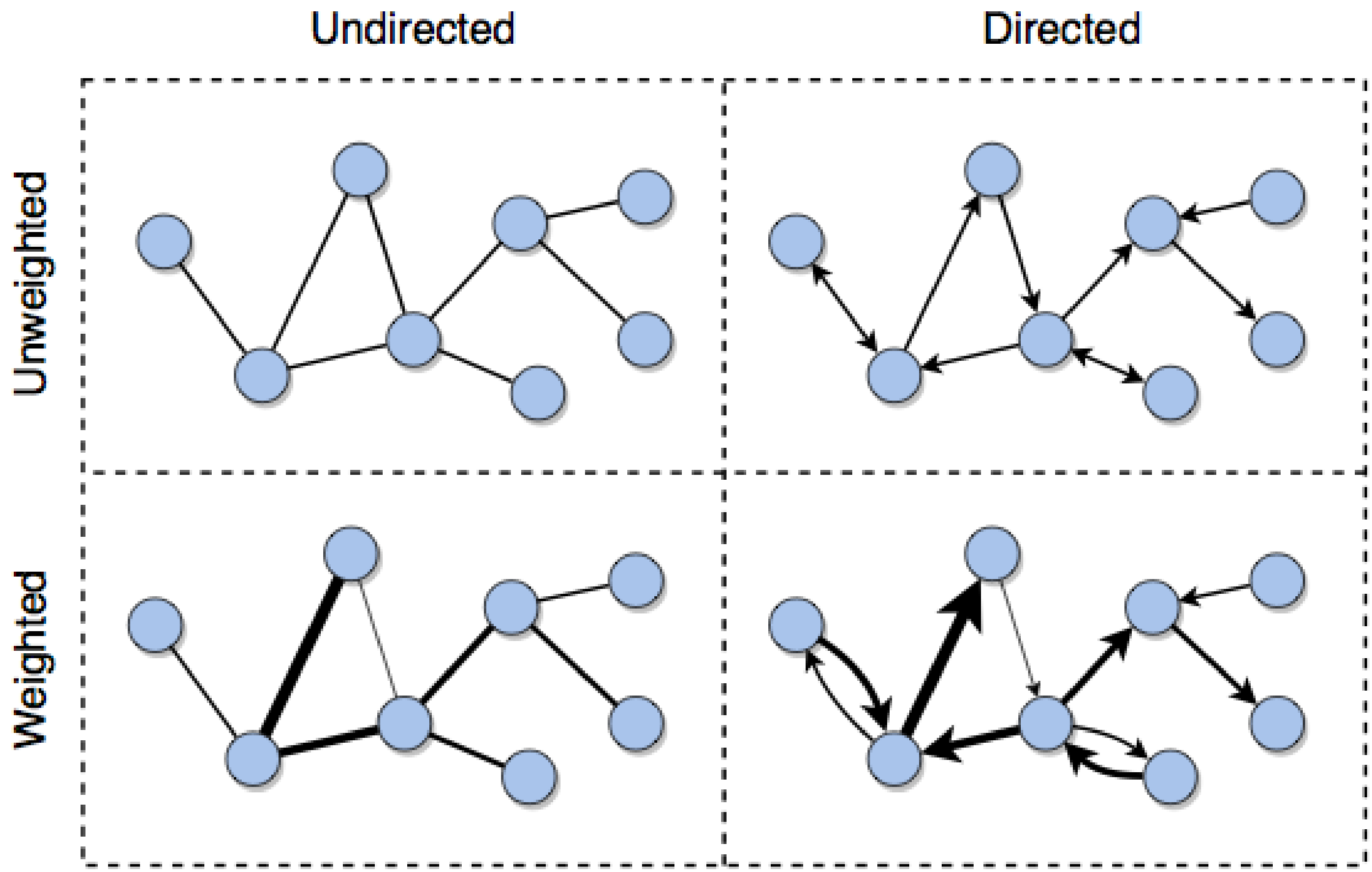
- A network can be **undirected** or **directed**
- **Directed Networks (Digraph)**
  - Links (directed edges) indicate directionality from a **source** node to a **target** node. The link (i, j) points from node i (source) to node j (target).
  - *Example*: Web hyperlink networks, where (i, j) means webpage i links to webpage j.
- **Undirected Networks:**
  - Links represent **bidirectional** relationships; the order of nodes is irrelevant
  - *Example*: Friendship networks, where connection (i, j) indicates mutual friendship

Definitions: Unweighted/Weighted Networks

universität koblenz
weiter:denken

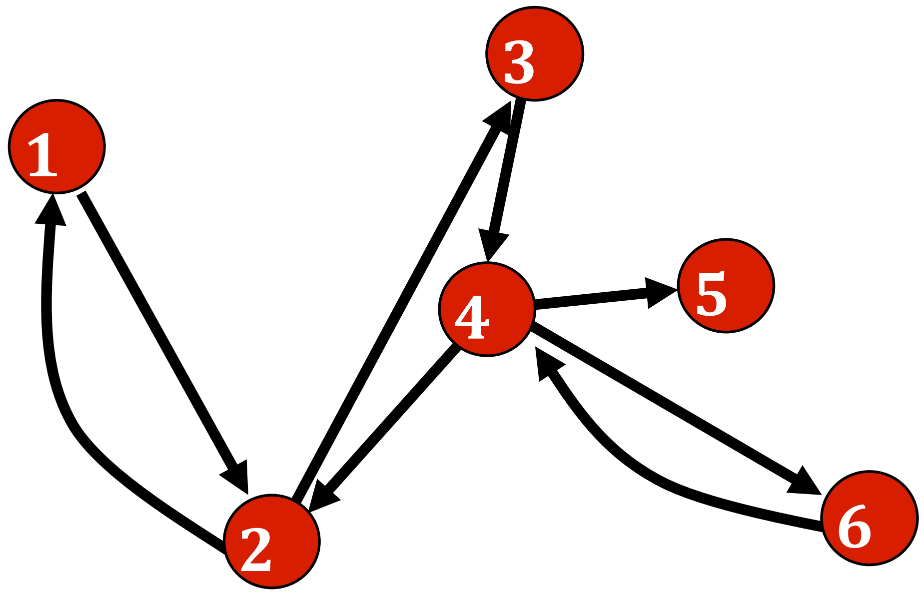
- A network can be **unweighted** or **weighted**
  - **Unweighted Network**:
    - Links have no assigned numerical value; they only represent the presence of a connection
    - *Example*: Social networks indicating simple friendships between individuals
  - **Weighted Network**:
    - Links carry numerical values ("weights"), denoted as (i,j,w), indicating the strength, capacity, or frequency of interactions between nodes i and j
    - *Example*: Traffic networks, where link weights represent traffic volumes between cities
  - Networks can simultaneously be **directed and weighted**, consisting of directed weighted links.
  - *Example*: Airline flight networks, where the directed weighted link (i,j,w) shows the number of flights from airport i (source) to airport j (target)

Graphical Representations of Undirected/Directed and Weighted Networks



- **Can you think of a few examples in each of these categories?**

Directed Networks




```
import networkx as nx # don't forget!

D = nx.DiGraph()
D.add_edge(1,2)
D.add_edge(2,1)
D.add_edges_from([(2,3),(3,4),…])
…
D.number_of_nodes()
D.number_of_edges()
D.edges()
D.successors(2)
D.predecessors(2)
D.neighbors(2)
```

Density and Sparsity (1/2)

- Network size N = number of nodes
- L = number of links
- Maximum possible number of links: $$L_{max} = \binom{N}{2} = \frac{N(N-1)}{2}$$
- Density: $$d = \frac{L}{L_{max}} = \frac{2L}{N(N-1)}$$
- The network is **sparse** if $d << 1$

Density and Sparsity (2/2)

- In a **directed** network things are a bit different
  - Maximum possible number of links: $L_{max} = N(N-1)$
  - Density: $d = \dfrac{L}{L_{max}} = \dfrac{L}{N(N-1)}$
- In a **complete** network, all pairs of nodes are connected and $d = 1$

```
G.number_of_nodes()
G.number_of_edges()
nx.density(G)
nx.density(D)

CG = nx.complete_graph(8471)
print(nx.density(CG)) # what does this print?
```

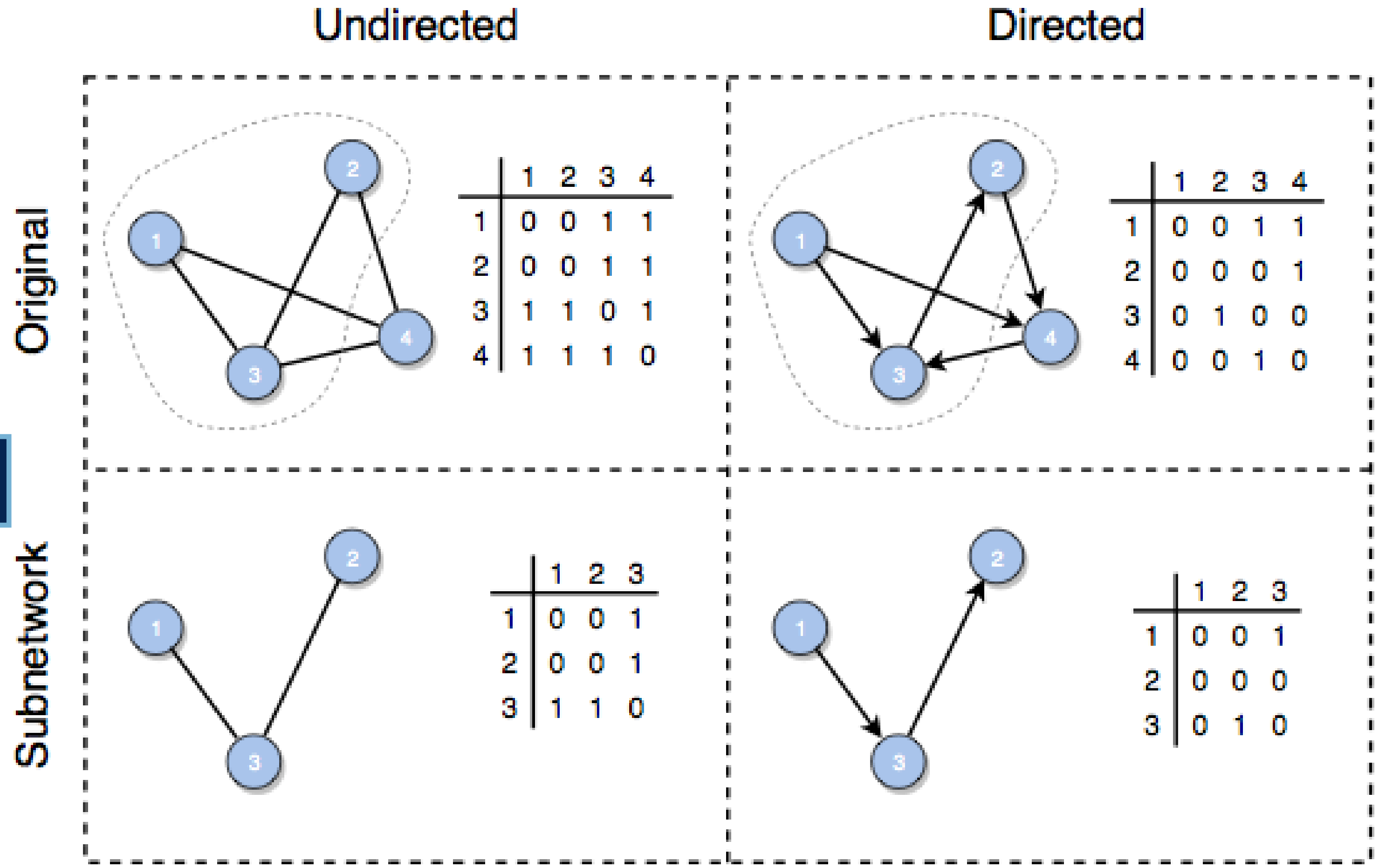
Subnetworks

universität koblenz
weiter:denken

- A **subnetwork** is a network obtained by selecting a subset of the nodes and all of the links among these nodes

```
S = nx.subgraph(G, node_list)
```

- A **clique** is a complete subnetwork



Original, undirected:

	1	2	3	4
1	0	0	1	1
2	0	0	1	1
3	1	1	0	1
4	1	1	1	0

Original, directed:

	1	2	3	4
1	0	0	1	1
2	0	0	0	1
3	0	1	0	0
4	0	0	1	0

Subnetwork, undirected:

	1	2	3
1	0	0	1
2	0	0	1
3	1	1	0

Subnetwork, directed:

	1	2	3
1	0	0	1
2	0	0	0
3	0	1	0

Degree

- The **degree** of a node is its number of links, or neighbors
- We typically use k_i to denote the degree of node i
- A node without neighbors is called a **singleton** (k=0)

```
G.degree(2) # returns the degree of node 2
G.degree()  # dict with the degree of all nodes of G
```

Degree: Directed Networks

- In a directed network we have
  - **in-degree** of a node = number of incoming links $k^{in}{}_i$
  - **out-degree** of a node = number of outgoing links $k^{out}{}_i$

```
D.in_degree(4)
D.out_degree(4)
D.degree(4)
```

Strength or Weighted Degree

- In a weighted network we have **strength** $s_i = \sum_j w_{ij}$ (a.k.a. **weighted degree**)

- In a weighted directed network we have
  - **in-strength** $s_i^{in} = \sum_j w_{ji}$
  - **out-strength** $s_i^{out} = \sum_j w_{ij}$

```
W.degree(4) # degree
W.degree(4, weight='weight') # strength
```

Average Degree

- The **average degree** of a network is $\langle k \rangle = \frac{\sum_i k_i}{N}$
- We can connect network size, number of links, density, and average degree
- In undirected networks:

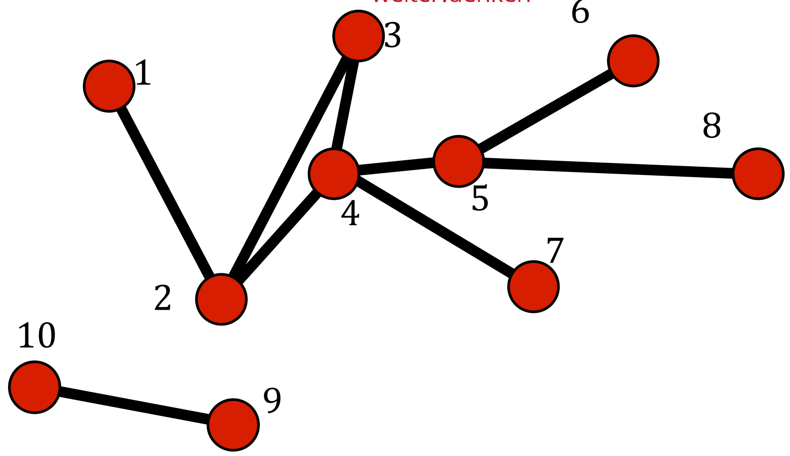
$$\langle k \rangle = \frac{2L}{N} = \frac{dN(N-1)}{N} = d(N-1)$$

$$d = \frac{\langle k \rangle}{N-1} = \frac{\langle k \rangle}{k_{max}}$$

Network Representations

- **Adjacency Matrix**: N x N matrix where each element $a_{ij} = 1$ if i and j are adjacent, zero otherwise
- The diagonal elements are zero because we have no self-loops
- In undirected networks, the matrix is symmetric: $a_{ij} = a_{ji}$

```
nx.adjacency_matrix(G)
print(nx.adjacency_matrix(G))
G.edge[3][4]
G.edge[3][4]['color']='blue'
G.edge[3][4]
G.edge[4]
```

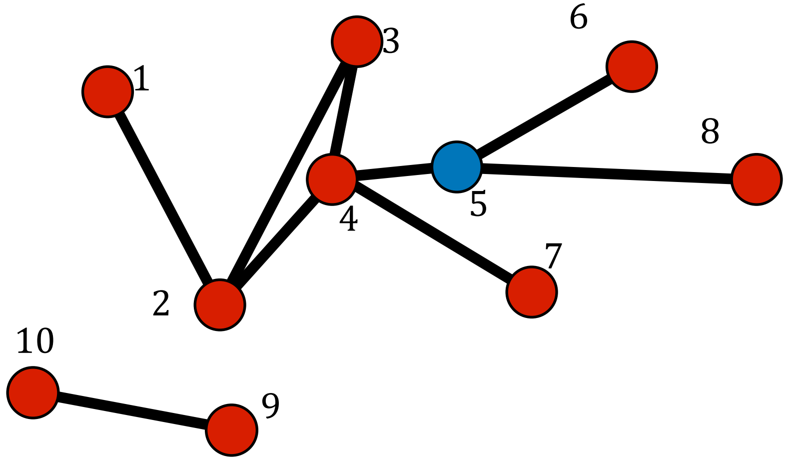


	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0
4	0	1	1	0	1	0	1	0	0	0
5	0	0	0	1	0	1	0	1	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	1	0

Network Representations: Undirected Nets

- In undirected networks, the degree is obtained by summing adjacency matrix elements across rows or columns:

$$k_i = \sum_j a_{ij} = \sum_j a_{ji}$$



	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0
4	0	1	1	0	1	0	1	0	0	0
5	0	0	0	1	0	1	0	1	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	1	0

Network Representations: Directed Nets

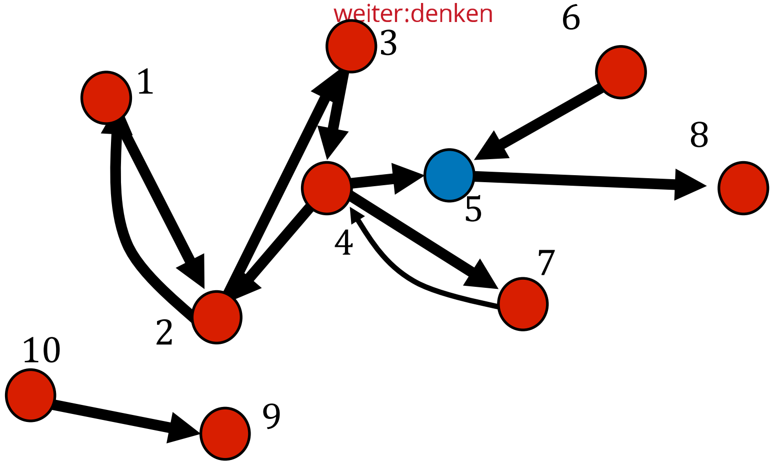
- In directed networks, the adjacency matrix is **not** symmetric
- The **out-degree** is obtained by summing adjacency matrix elements across **rows**:

$$k_i^{out} = \sum_j a_{ij}$$

- The **in-degree** is obtained by summing adjacency matrix elements across **columns**:

$$k_i^{in} = \sum_j a_{ji}$$

```
print(nx.adjacency_matrix(D))
D.edge[3][4]
D.edge[4][3]
D.edge[4]
```

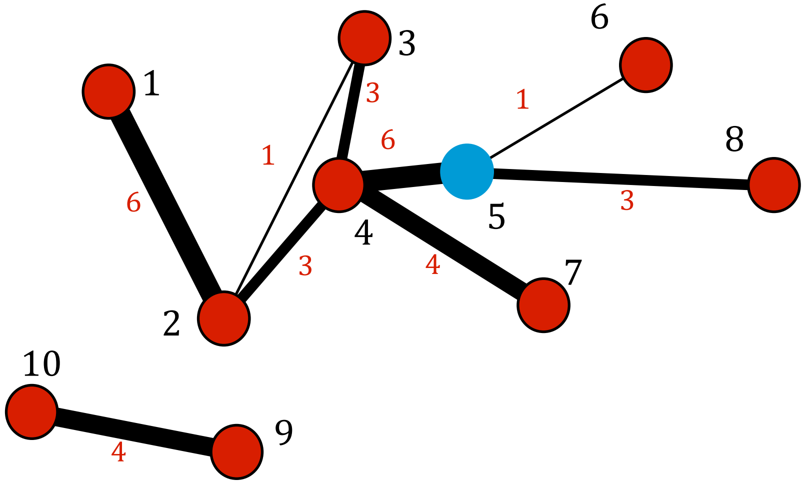


	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	1	0	1	0	0	0
5	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0

Network Representations: Weighted Nets

- In weighted networks, each element w_{ij} represents the weight of the link between *i* and *j*, zero if there is no link
- If undirected, the strength is obtained by summing adjacency matrix elements across rows or columns
- If directed, the in/out-strength is obtained by summing adjacency matrix elements across columns/rows

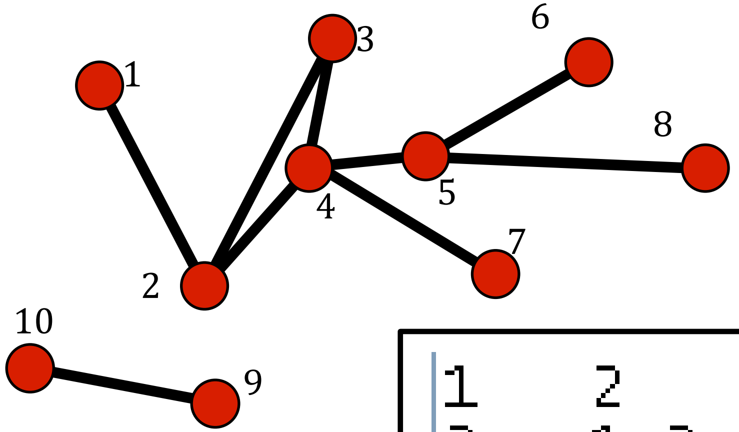
```
print(nx.adjacency_matrix(W))
W.edge[2][3]
W.edge[2]
W.edge[2][3]['weight'] = 2
W.edge[2][3]
W.edge[2]
```



	1	2	3	4	5	6	7	8	9	10
1	0	6	0	0	0	0	0	0	0	0
2	6	0	1	3	0	0	0	0	0	0
3	0	1	0	3	0	0	0	0	0	0
4	0	3	3	0	6	0	4	0	0	0
5	0	0	0	6	0	1	0	3	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	4	0	0	0	0	0	0
8	0	0	0	0	3	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	4
10	0	0	0	0	0	0	0	0	4	0

Adjacency List

- List of neighbors for each node
- In undirected networks, each link is listed twice
- In weighted networks, each neighbor is replaced by a pair (neighbor, weight)




```
1    2
2    1 3 4
3    2 4
4    2 3 5 7
5    4 6 8
6    5
7    4
8    5
9    10
10   9
```

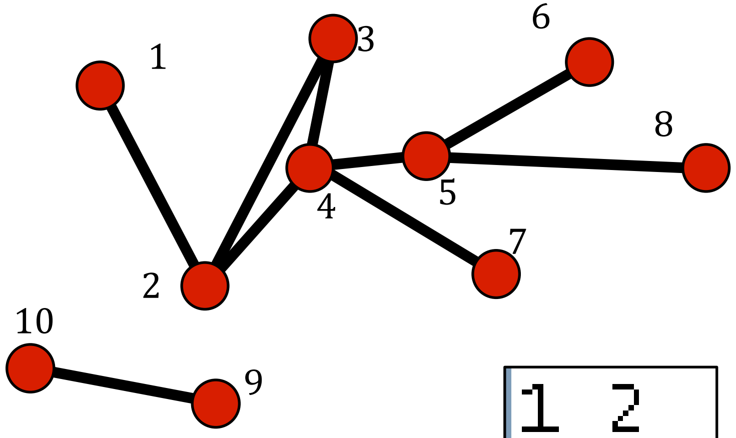
```
G.neighbors(2)

for n,neighbors in G.adjacency():
    for neighbor,link_attributes in neighbors.items():
        print('(%d, %d)' % (n,neighbor))

nx.write_adjlist(G, "netfile.adjlist")
G2 = nx.read_adjlist("netfile.adjlist") # G and G2 are
isomorphic
```

Edge List

- List of node pairs that are connected
- In weighted networks, each pair is replaced by a triplet (i, j, weight)




```
for i,j in G.edges:
    print('%d %d' %(i,j))

nx.write_edgelist(G, "netfile.edgelist")
G3 = nx.read_edgelist("netfile.edgelist") # G and G3 are isomorphic

nx.write_weighted_edgelist(W, "wf.edges") # store  weights
W2 = nx.read_weighted_edgelist("wf.edges") # W and W2 are isomorphic
```

```
1 2
2 3
2 4
3 4
4 5
4 7
5 6
5 8
9 10
```

Network Theory and Dynamic Systems

03. Small Worlds

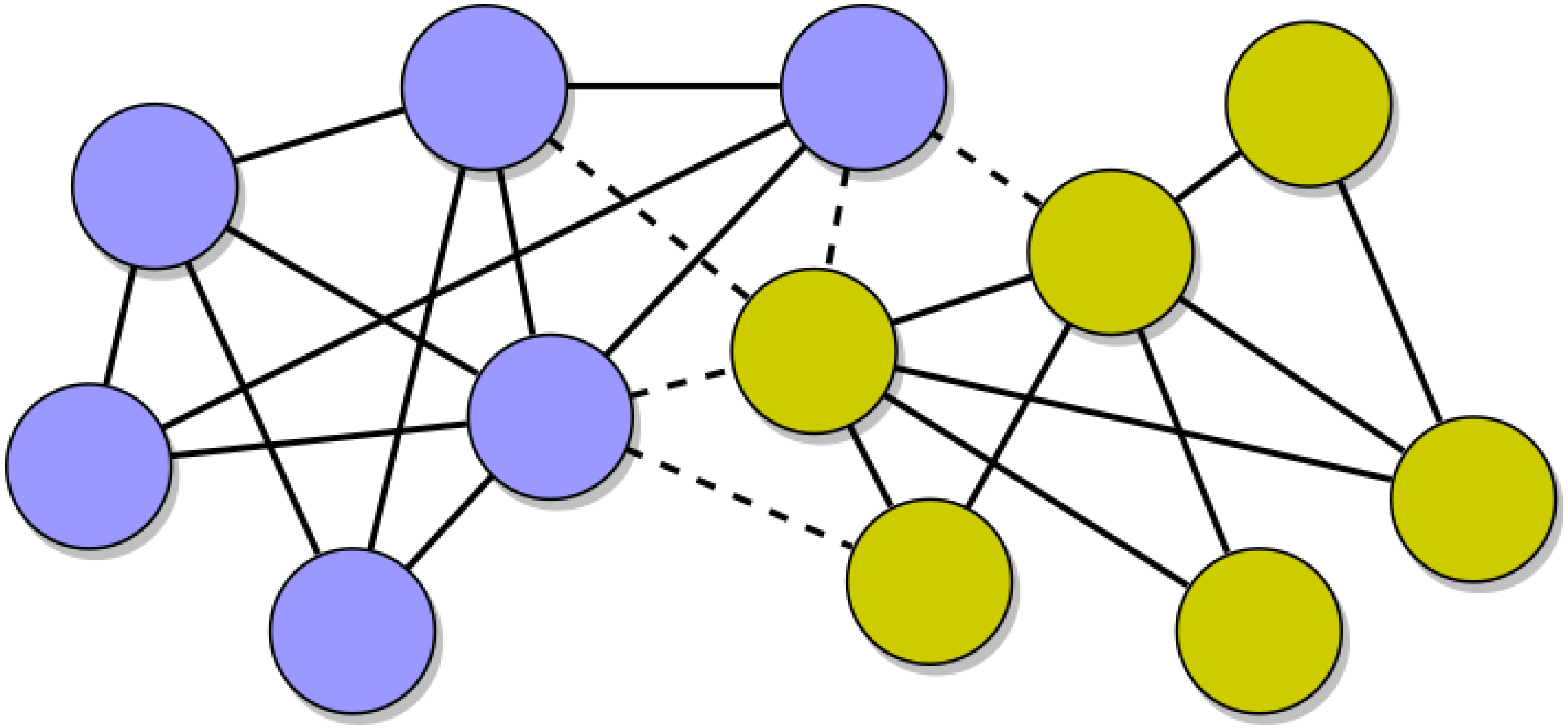
SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Birds of a Feather

Understanding the Tendency of Similar Individuals to Form Connections



- This network graph illustrates how nodes—each representing an individual with a specific trait (such as political views or musical preferences)—tend to connect more frequently with others who share similar attributes
- The color of each node denotes a particular characteristic
- As shown, nodes of the same color are more likely to be linked, highlighting the principle of homophily: the tendency for individuals to associate with those who are similar to themselves

Introduction to Assortativity

- Assortativity in social networks describes the tendency of nodes (such as individuals or entities) to form connections with others who share similar attributes
  - These attributes may include age, geographic location, interests, or social background
- It quantifies the extent to which similar nodes are more likely to be connected compared to dissimilar ones
- Understanding assortativity provides insight into how cohesive subgroups emerge within broader social structures, and reveals the underlying dynamics that drive these patterns

Assortativity

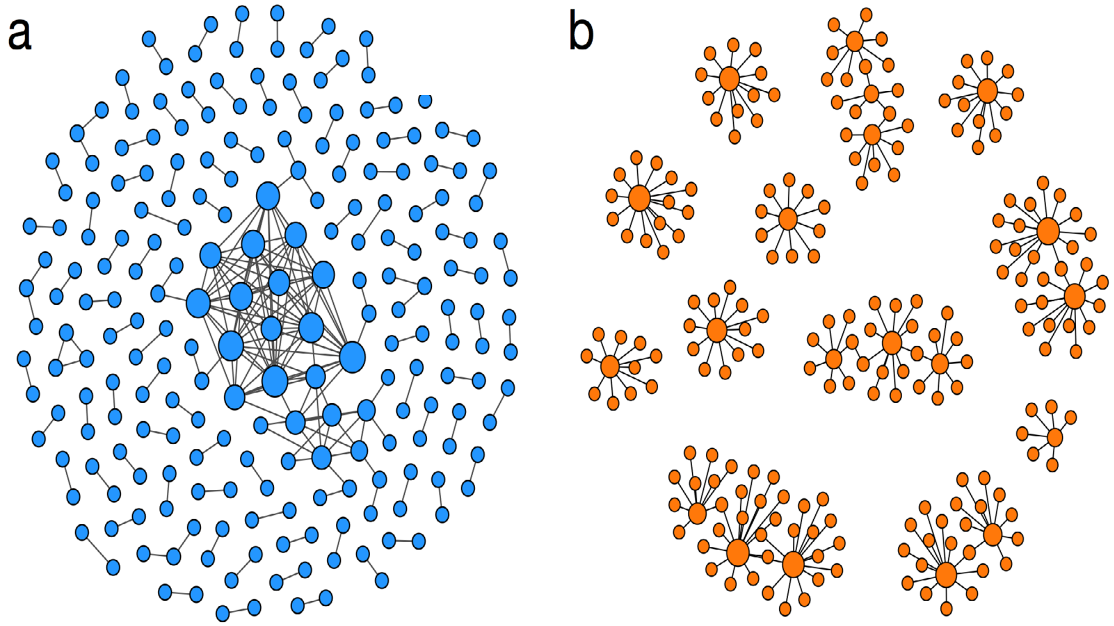
universität koblenz
weiter:denken

- Two possible mechanisms by which assortativity emerges naturally:
  1. **Selection** or **homophily**: similar nodes become connected
  2. (Social) **influence**: connected nodes become more similar
- It can also be a bad thing
  - For example "**echo chambers**" and "**groupthink**" are situations where your friends are like you, diversity is killed, and you are only exposed to opinions that reinforce your pre-existing beliefs…

We're going to explain it in more detail later

Degree Assortativity

- A.k.a. **degree correlation**
- Assortative networks have a **core-periphery** structure with hubs in the core
  - Example: social networks
- Disassortative networks have **hub-and-spoke** (or **star**) structure
  - Example: Web, Internet, food webs, bio networks



The Dark Side of Homophily (2/2)

- **Homophily**—the tendency of individuals to form ties with others who are similar to them in key dimensions (e.g., interests, beliefs, or demographics)— can also have negative consequences:

the formation of homogeneous groups or clusters within larger networks, which can significantly impact the flow and integrity of information

⬇

Polarization, Misinformation, and Social Manipulation

Polarization

The process by which opinions within groups become increasingly extreme, driving groups further apart in values and beliefs.

- Homophily reduces exposure to differing perspectives and reinforces *confirmation bias*, where individuals predominantly encounter information that supports their existing views
- This dynamic contributes to a fragmented society in which shared understanding diminishes, compromise becomes more difficult, and the risk of social conflict and instability increases

Spread of Misinformation

- **Role of Echo Chambers**
  - **Echo Chamber Effect:** Social networks with high homophily foster environments where the same ideas, beliefs, or misinformation are continuously reinforced, reducing exposure to alternative or corrective viewpoints
  - **Vulnerability to Fake News:** These echo chambers accelerate the spread of misinformation, as individuals are more likely to accept content that aligns with their pre-existing beliefs without questioning its validity.
- **Exploitation by Malicious Actors**
  - Manipulative agents (e.g., state-sponsored groups, malicious organizations) can exploit homophilous networks by strategically injecting misinformation that spreads rapidly within like-minded communities, deepening social divides
  - **Example**: Coordinated misinformation campaigns during election periods that target specific ideological or demographic groups to influence public opinion or suppress voter turnout

Social Bots and Manipulation Tactics

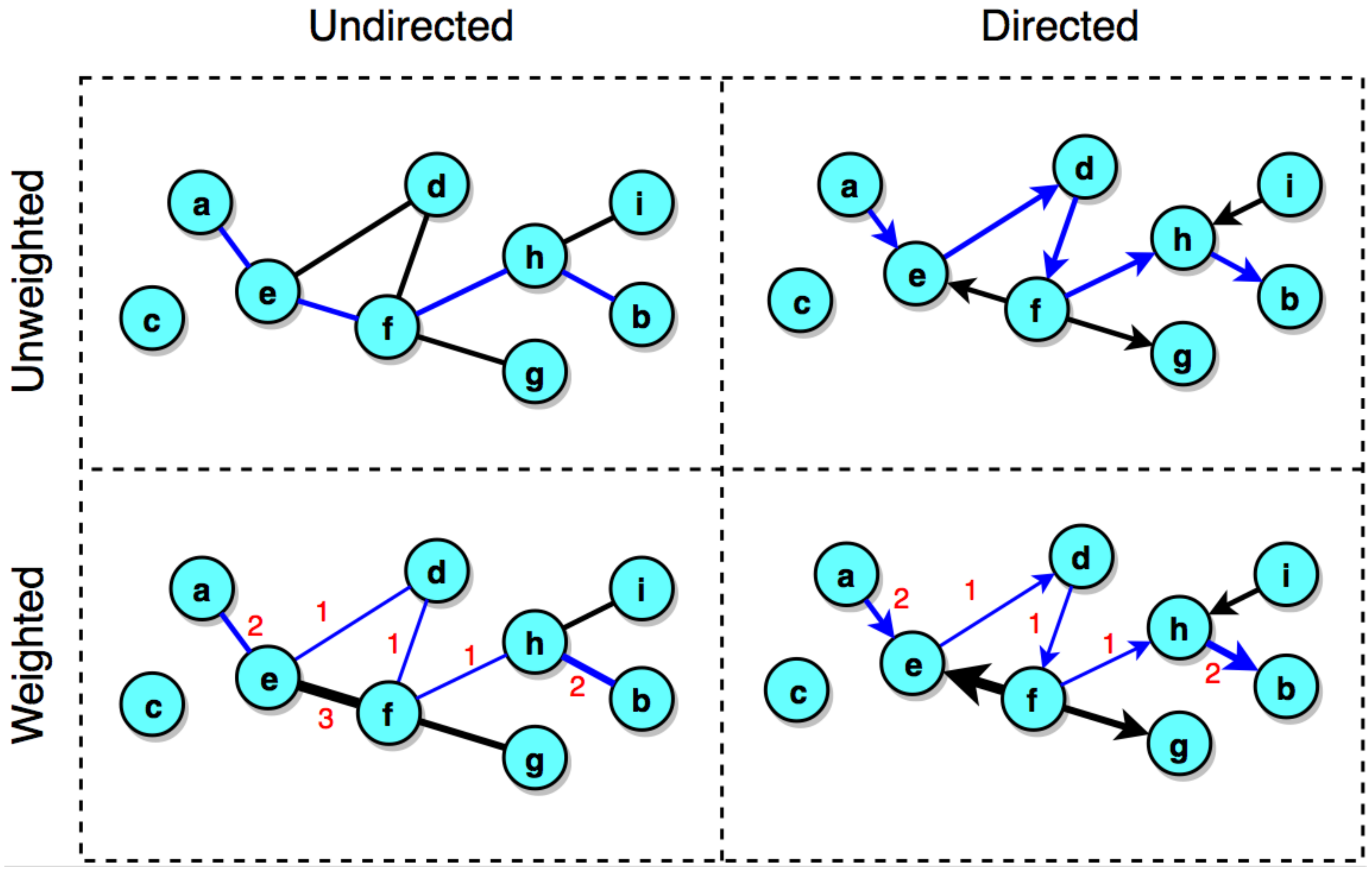
- **Social Bots**
  - Automated accounts designed to imitate human behavior in online spaces
  - They are used to manipulate discourse, amplify particular viewpoints, and disseminate misinformation at scale
- **Sybil Attacks:**
  - The creation of numerous fake identities to fabricate consensus or propagate specific narratives, distorting public perception and potentially influencing real-world outcomes
- The orchestrated use of such tactics can undermine democratic processes by fostering political polarization and misleading the public on critical issues

Paths: Definitions

- **Path**: sequence of links traversed to go from a **source** to a **target** node
  - In a directed network, links must be traversed according to their direction
  - Note: A path may not always exist between two nodes
- **Cycle**: path where source and target node are the same
- **Simple path**: A path that does **not** revisit any link
  - We will only deal with simple paths
- **Path length**: number of links in path
- Finding paths was the earliest problem studied in network science

Shortest Paths

- **Shortest path** between two nodes: minimal length (there may be more than one)
  - In weighted networks, weights may represent distances
- **Shortest path length** or **distance**: length of shortest path
  - Undefined (∞) if there is no path



APL and Diameter (1/2)

- We can use the shortest paths to characterize a network:
  - The **diameter** is the longest shortest-path length, or the maximum of the shortest path lengths across all pairs of nodes:

$$\ell_{max} = \max_{i,j} \ell_{ij}$$

  - The **average path length** (APL) is the average of the shortest path lengths across all pairs of nodes
    - Undirected network: $$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{\binom{N}{2}} = \frac{2\sum_{i,j} \ell_{ij}}{N(N-1)}$$
    - Directed network:

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{N(N-1)}$$

APL and Diameter (2/2)

- What if there is not a path between one or more pairs of nodes?
  - We can say APL and diameter are undefined (as NetworkX does)
  - We can measure APL and diameter within the largest connected component (defined later)
  - We can use a mathematical trick:

$$\langle \ell \rangle = \left(\frac{\sum_{i,j} \frac{1}{\ell_{ij}}}{\binom{N}{2}} \right)^{-1}$$

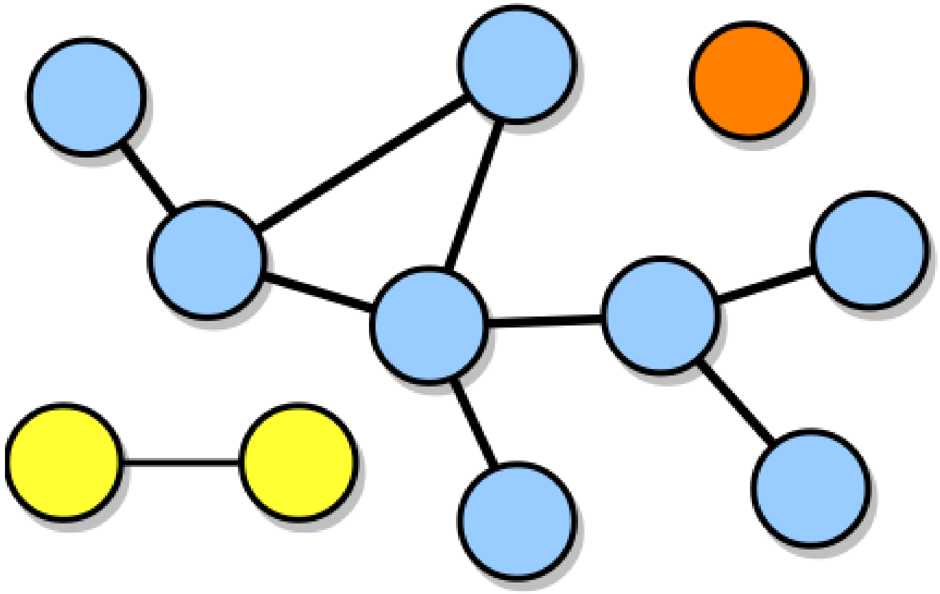
Concept of Connectedness

- **Connectedness** in a network refers to the ability to reach any node from any other node via paths of intermediate nodes and links
- It is a critical measure that defines the integrity and usability of the network, influencing how information or processes flow through the network
- **Density and Connectivity**: Higher density generally implies better connectedness, meaning that the network can more likely sustain connectivity even if some links are removed or fail

Connectedness and Components (1/3)

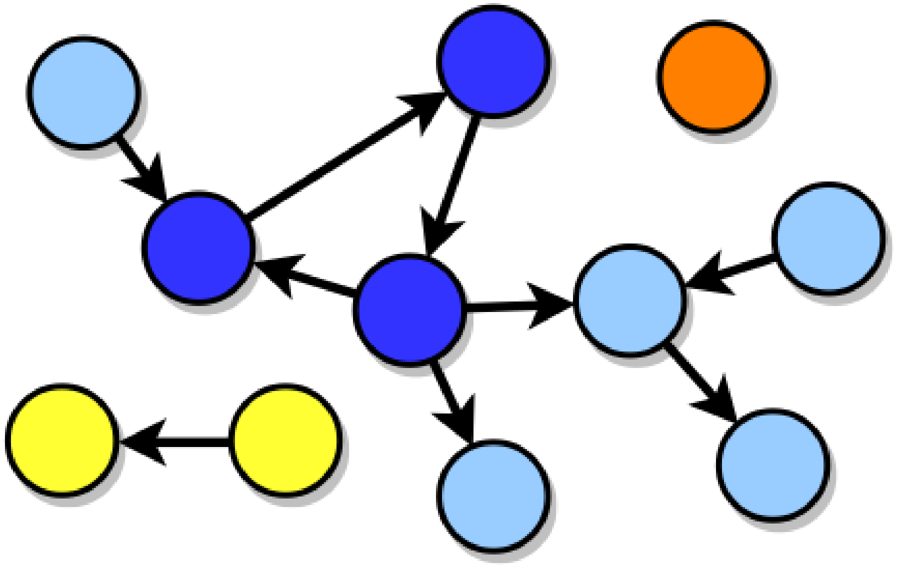
universität koblenz
weiter:denken

- A network is **connected** if there is a path between any two nodes
- If a network is not connected, it is **disconnected** and has multiple connected components
- A **connected component** is a connected subnetwork
  - The largest one is called **giant component**; it often includes a substantial portion of the network
  - A **singleton** is the smallest-possible connected component



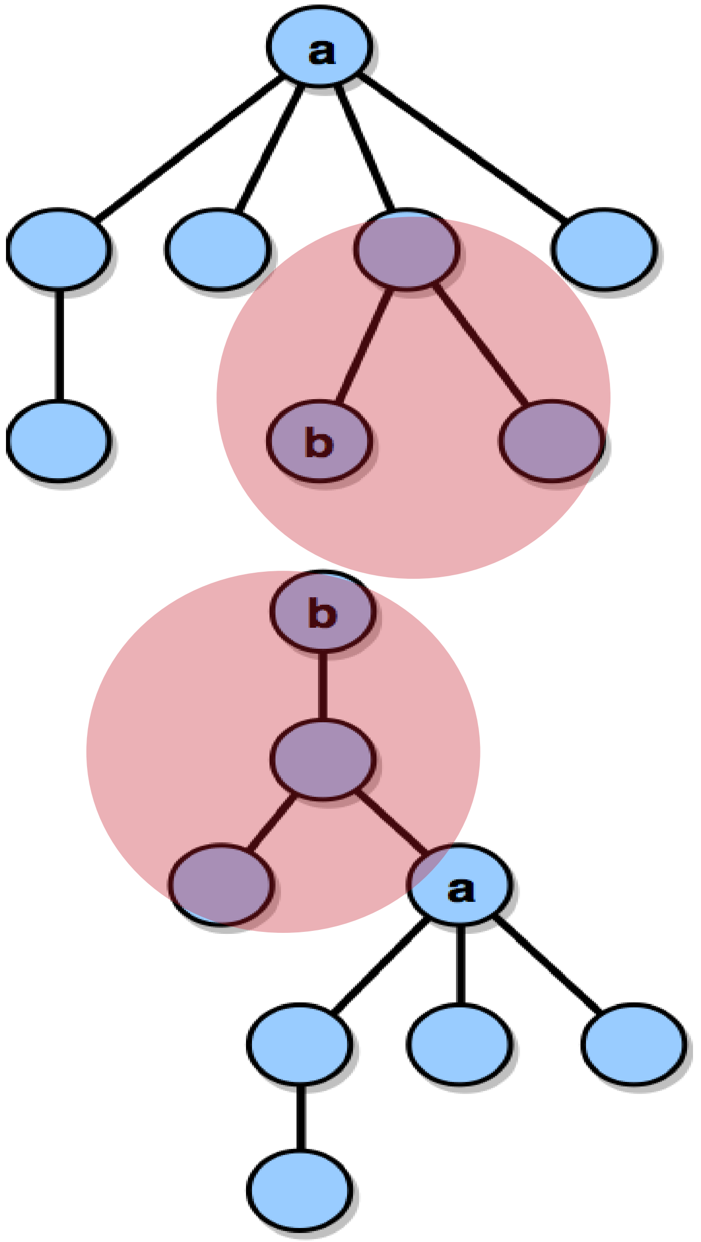
Connectedness and Components (2/3)

- **A directed network** can be **strongly connected** or **weakly connected** if there is a path between any two nodes, respecting or disregarding the link directions, respectively
- Similarly for **strongly connected** or **weakly connected components**
- The **in-component** of a strongly connected component S is the set of nodes from which one can reach S, but that cannot be reached from S
- The **out-component** of a strongly connected component S is the set of nodes that can be reached from S, but from which one cannot reach S



Trees (1/2)

- A **tree** is a **connected** network **without cycles**
- A **tree** is a **connected** network **with *N*-1 links**
  - Exercise: prove that these two definitions are equivalent
- In a tree there is a **single path** between any two nodes
- Trees are **hierarchical**: you can pick a node as the **root**. Each node is connected to a **parent** node (toward the root) and to one or more **children** nodes (away from the root). Exceptions:
  - The root has no parent
  - The **leaves** have no children



Introduction to Finding Shortest Paths

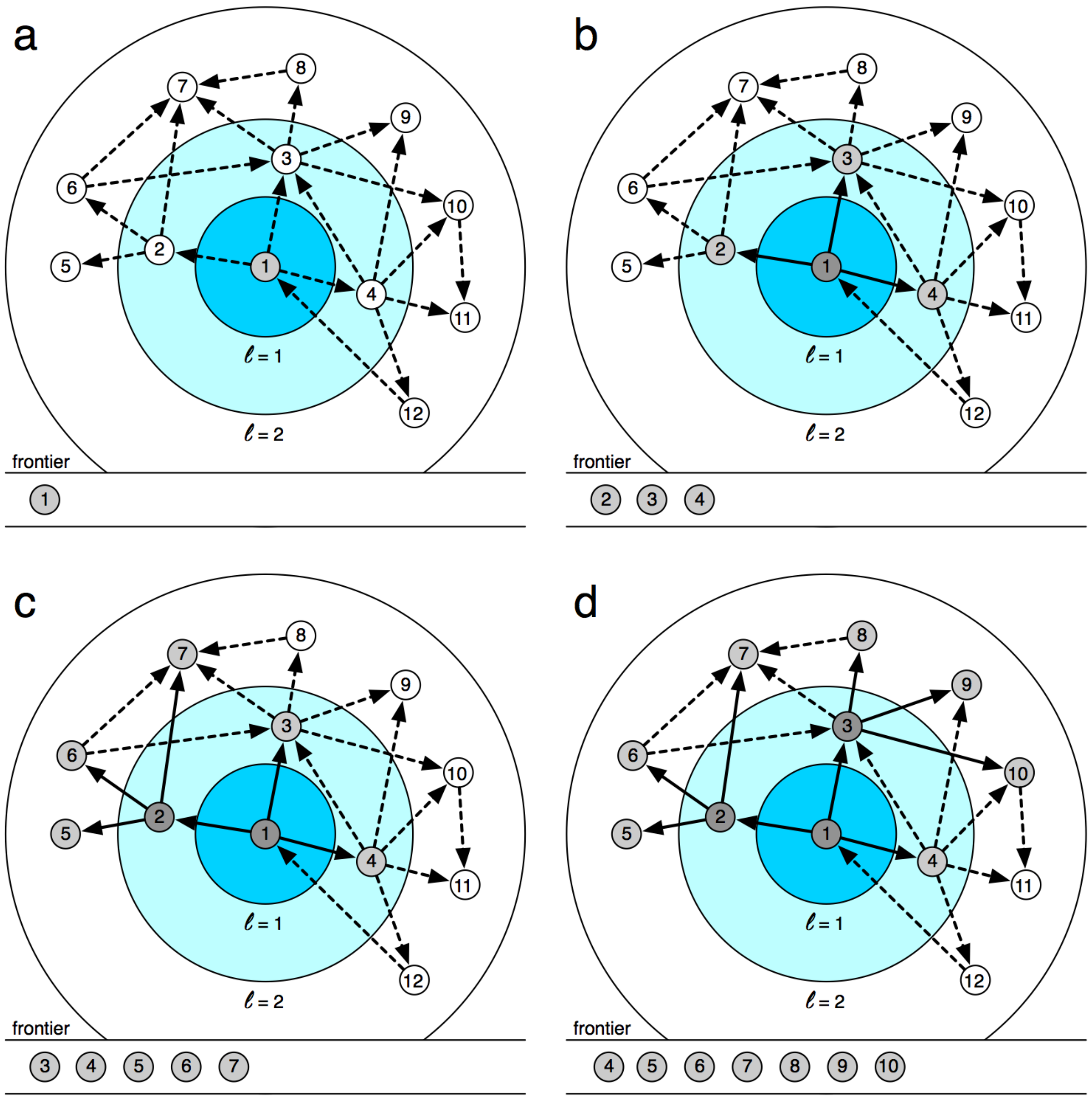
- To determine the shortest path between two nodes, network analysis tools like NetworkX systematically explore the network from a starting point (source node) to every other node
- This is essential in various applications, including routing and navigation systems, data organization, and optimizing network communications
- **Web Crawlers**
  - Search engines use automated programs known as web crawlers that navigate the web to find and index new pages
  - The process involves mapping the web as a network of pages (nodes) linked by hyperlinks (edges) and using shortest path algorithms to efficiently traverse this network

Breadth-First Search

The algorithm, or procedure for navigating through a network starting from a *source* node and finding the shortest path between the source and every other node in the network is called ***breadth-first search***

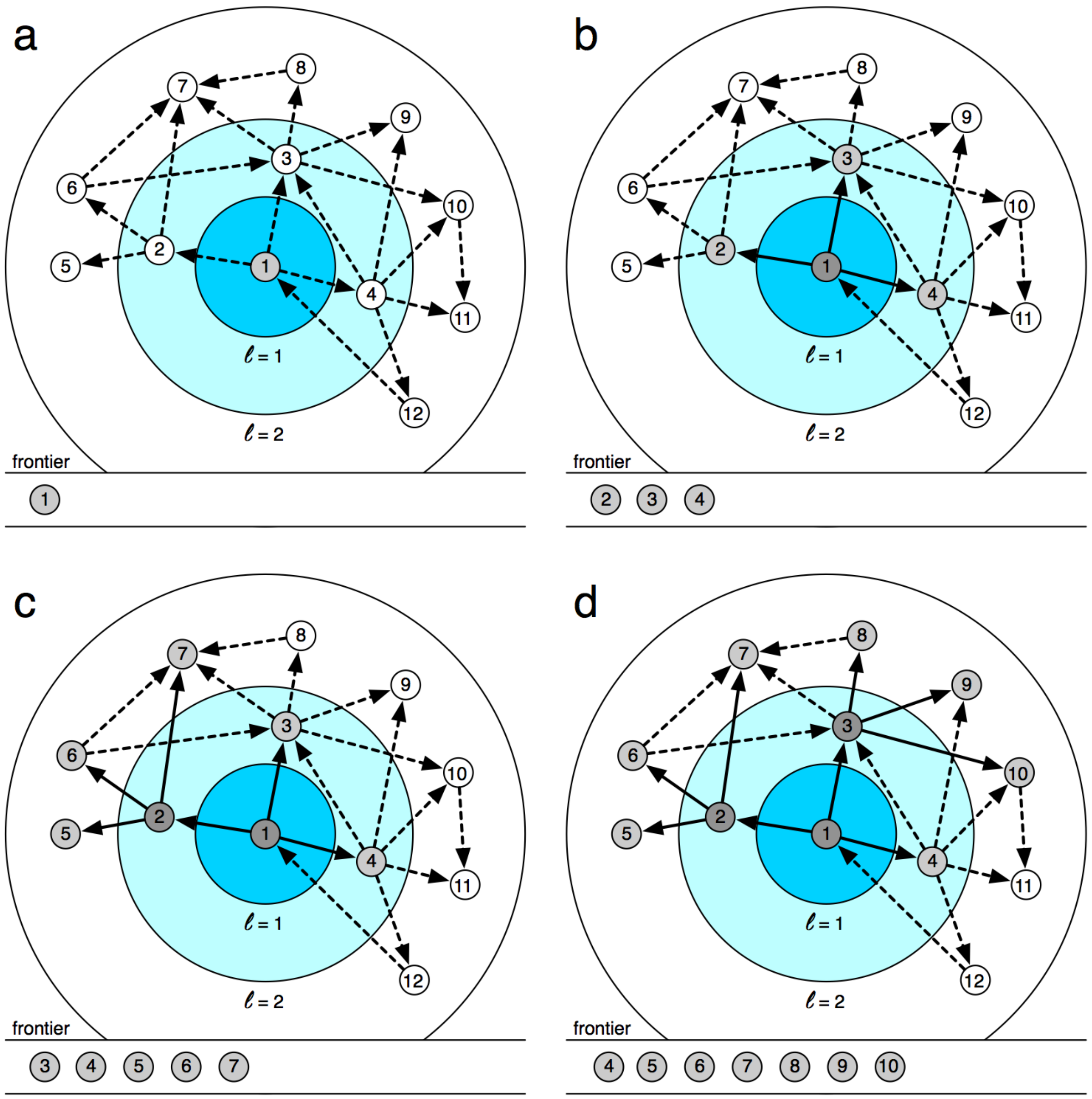
Breadth-First Search (BFS)

- Each node has an attribute storing its **distance** ℓ from the source, initially $\ell = -1$ except $\ell(\text{source}) = 0$
- A queue (FIFO) holds the **frontier**, initially contains the source
- A directed **shortest path tree**, initially all the nodes and no links
- Iterate until the frontier is empty:
  1. Remove next node i in frontier
  2. For each neighbor/successor j of i with $\ell(j) = -1$:
     1. Queue j into frontier
     2. $\ell(j) = \ell(i) + 1$
     3. Add link $(i \rightarrow j)$ to shortest-path tree



Breadth-First Search (BFS)

- To find the shortest path from the source to any target node, we have to follow the links in the shortest-path tree backward from the target node through predecessors in the upper layers, until we arrive at the source
- Recall that in a tree there is a single path to the root; each node has a single predecessor. Then we have to reverse the path to obtain the shortest path from the source to the target
  - In an undirected network this is the same as the path from the target to the source, but in a directed network they may be different



Social Distance

universität koblenz
weiter:denken

- How close or distant are two nodes in a network?
  - This question is fundamentally about the average path length between nodes
  - Average path length provides a measure of node closeness—offering insights into how efficiently information or influence can spread through a network
- The question has been explored extensively in social networks
  - Unlike grid-like structures (e.g., road or power networks), which often have long paths, social networks tend to show **surprisingly short average path lengths**, due to high **connectivity** and **clustering**
  - Let us start by considering coauthorship networks, in which nodes are scholars and links represent two people having coauthored one or more publications
  - Coauthorship data is both **accessible** and **reliable**, making these networks a valuable case study for examining **social collaboration dynamics**

Erdős numbers

Davis

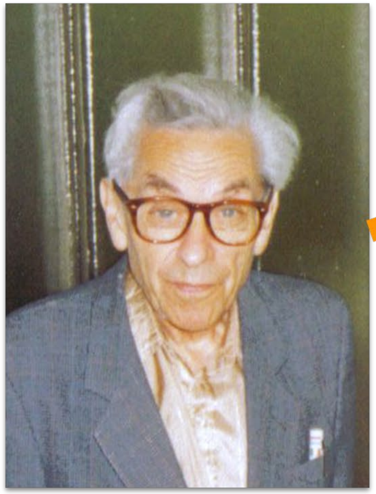
BY EMILIO FERRARA, ONUR VAROL, CLAYTON DAVIS, FILIPPO MENCZER, AND ALESSANDRO FLAMMINI

The Rise of Social Bots

4

Menczer

Fortunato



Paul Erdős
(Image by Kmhkmh - CC BY 3.0 commons.wikimedia.org/w/index.php?curid=38087162)

0

Journal of Graph Theory

Article | Full Access

Highly irregular graphs†

Yousef Alavi, Gary Chartrand, F. R. K. Chung, Paul Erdös, R. L. Graham, Ortrud R. Oellermann

1

Fan Chung (Image by Che Graham - CC BY 3.0 commons.wikimedia.org/w/index.php?curid=19475582)

The Workshop on Internet Topology (WIT) Report

Dmitri Krioukov CAIDA dima@caida.org	Fan Chung UCSD fan@math.ucsd.edu	kc claffy CAIDA kc@caida.org
Marina Fomenkov CAIDA marina@caida.org	Alessandro Vespignani Indiana University alexv@indiana.edu	Walter Willinger AT&T Research walter@research.att.com

2

Topical interests and the mitigation of search engine bias

S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani

PNAS August 22, 2006 103 (34) 12684-12689; https://doi.org/10.1073/pnas.0605525103

Communicated by Elinor Ostrom, Indiana University, Bloomington, IN, July 1, 2006 (received for review March 2, 2006)

3

Alessandro Vespignani - Image CC BY-SA 2.0 commons.wikimedia.org/w/index.php?curid=60299947

Six Degrees of Seperation

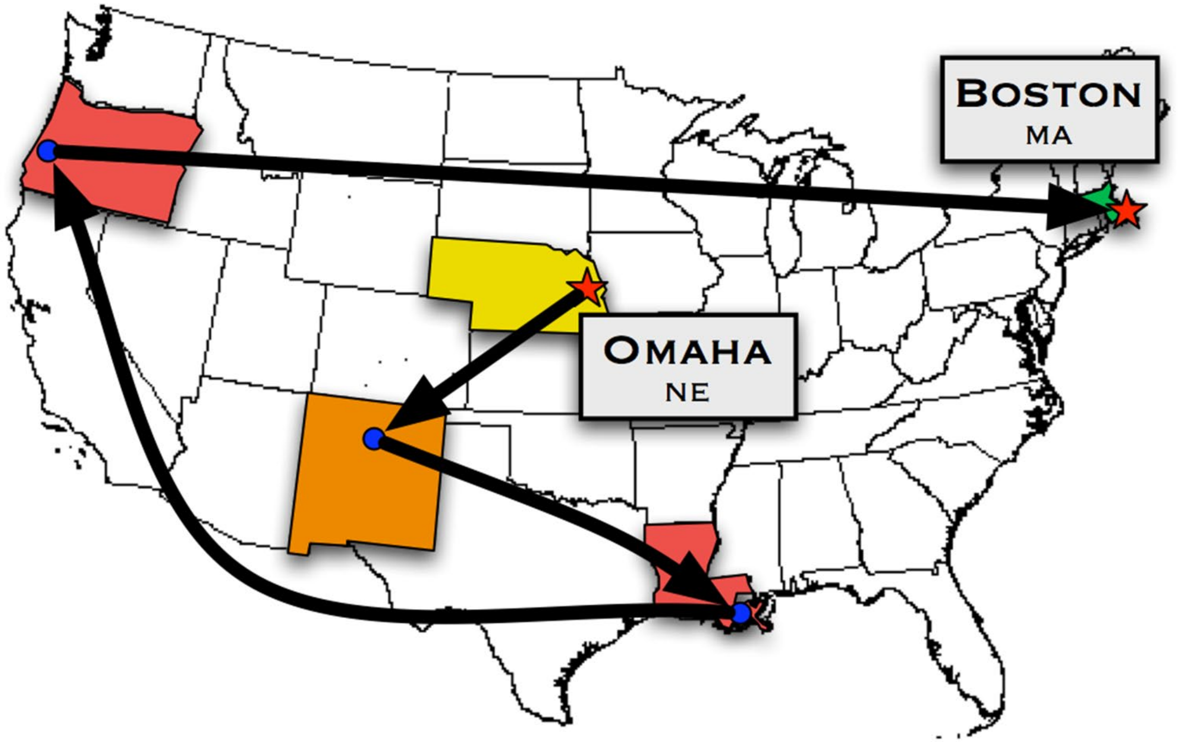
- What have we observed? Social networks tend to exhibit very short paths between nodes
- The concept of six degrees of separation suggests that any two people in a social network are connected through at most six intermediate steps
- The idea first appeared in the 1929 short story “Chains” by Hungarian author Frigyes Karinthy
- In 1967, psychologist Stanley Milgram provided empirical support through an influential experiment measuring social distance between randomly selected individuals in the United States
- The phrase “six degrees of separation” was later popularized by John Guare in his 1991 play (and subsequent film adaptation)

Concept of Small Worlds

- **Common Experience:** It’s a frequent occurrence to meet someone new only to discover a mutual acquaintance, illustrating the unexpectedly short social distances within large populations
- The "small world" phenomenon suggests that social networks are characterized by short path lengths between individuals, typically quantified by the concept of "degrees of separation"

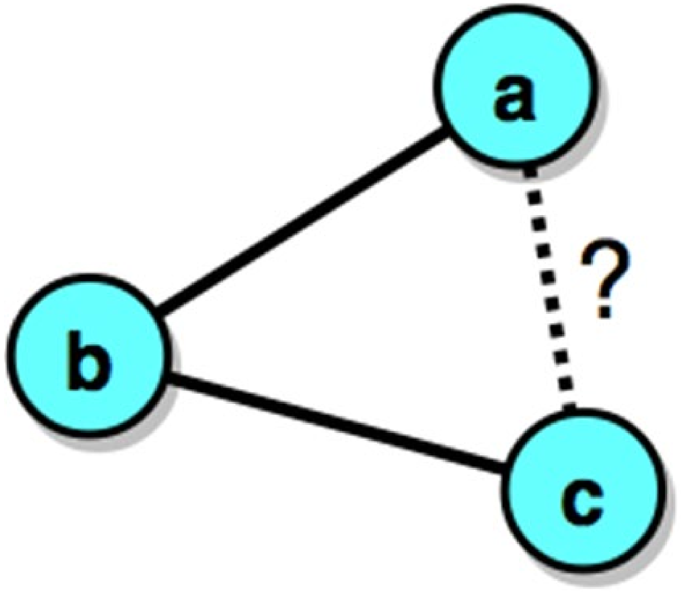
Milgram’s Experiment

- Instructions: send to personal acquaintance who is more likely to know target
- 160 letters to people in Omaha, NE and Wichita, KS
- 2 targets in Mass: the wife of a student in Sharon and a stockbroker in Boston
- 42 letters reached the target(only 26%)
- Average: 6.5 steps (range: 3-12 steps)
- Much lower than most people expected!
- “Small world” effect is still surprising



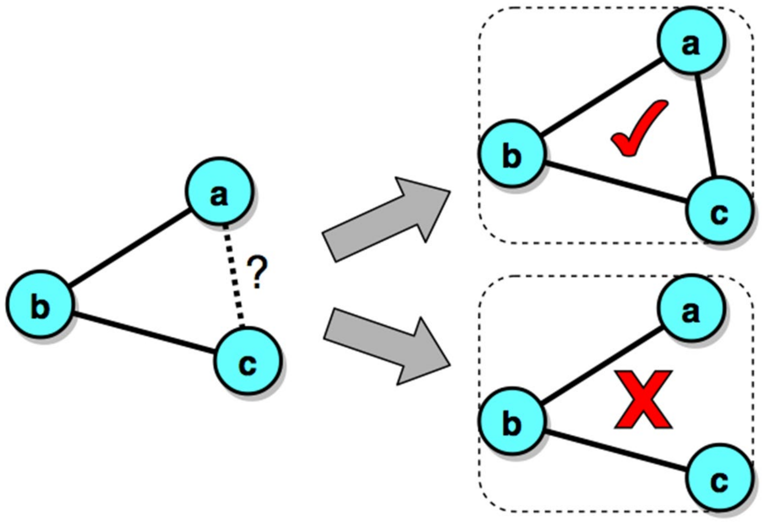
Friend of a Friend (1/3)

- Another feature of social (and some other) networks is the presence of **triangles**: if Alice (A) and Bob (C) are both friends with Charlie (B), they are also likely friends of each other
- In other words, many friends of my friends are also my friends



Friend of a Friend (2/3)

- A triangle is a triad (set of three nodes) where each pair of nodes is connected
- The connectivity among the neighbors of the nodes is crucial for understanding the local structure of a network
- This connectivity highlights how tightly knit or *clustered* the nodes are within the network



Clustering Coefficient

- We can measure the number of triangles that a node actually has relative to how many it could have
- The **clustering coefficient** of a node is the **fraction of pairs of the node's neighbors that are connected to each other:**

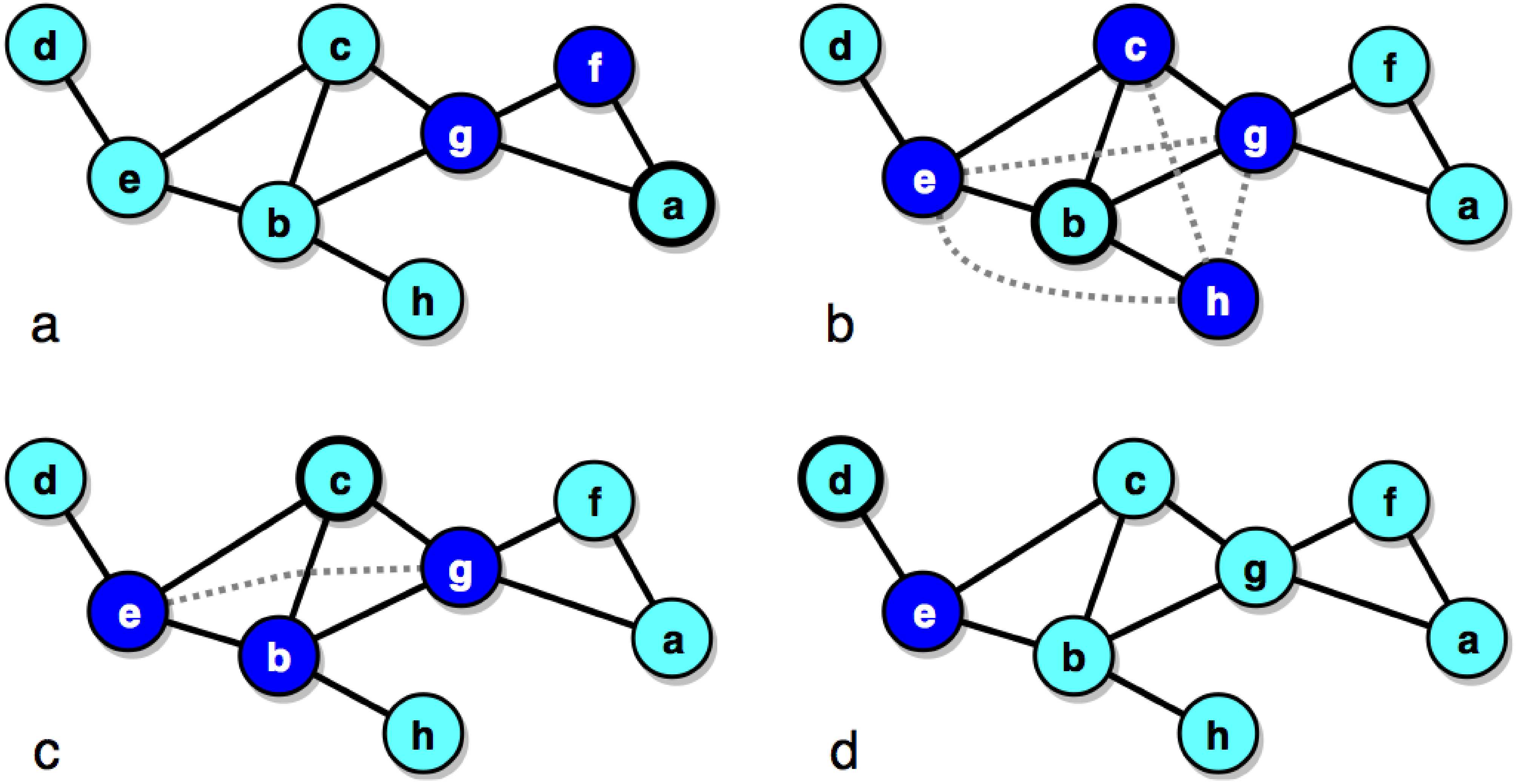
$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i - 1)}$$

where $\boldsymbol{\tau}(i)$ tau is the number of triangles involving *i.* Note that in this definition, the clustering coefficient is undefined if $k_i < 2$: a node must have at least degree 2 to have any triangles. However NetworkX assumes C=0 if k=0 or k=1.

Clustering Coefficient Exercises (2/2)

weiter:denken

$C(a) = \frac{1}{1}$



$C(b) = \frac{2}{6}$

$C(c) = \frac{2}{3}$

C(d) undefined

Network Clustering Coefficient (1/2)

- The clustering coefficient of the network is the average of the clustering coefficients of the nodes:

$$C = \frac{\sum_{i:k_i>1} C(i)}{N_{k>1}}$$

  - Again, we should exclude singletons and nodes with k=1, but NetworkX assumes those have C=0

```
nx.triangles(G)              # dict node -> no. triangles
nx.clustering(G, node)       # clustering coefficient of node
nx.clustering(G)             # dict node -> clustering coefficient
nx.average_clustering(G) # network's clustering coefficient
```

Network Theory and Dynamic Systems

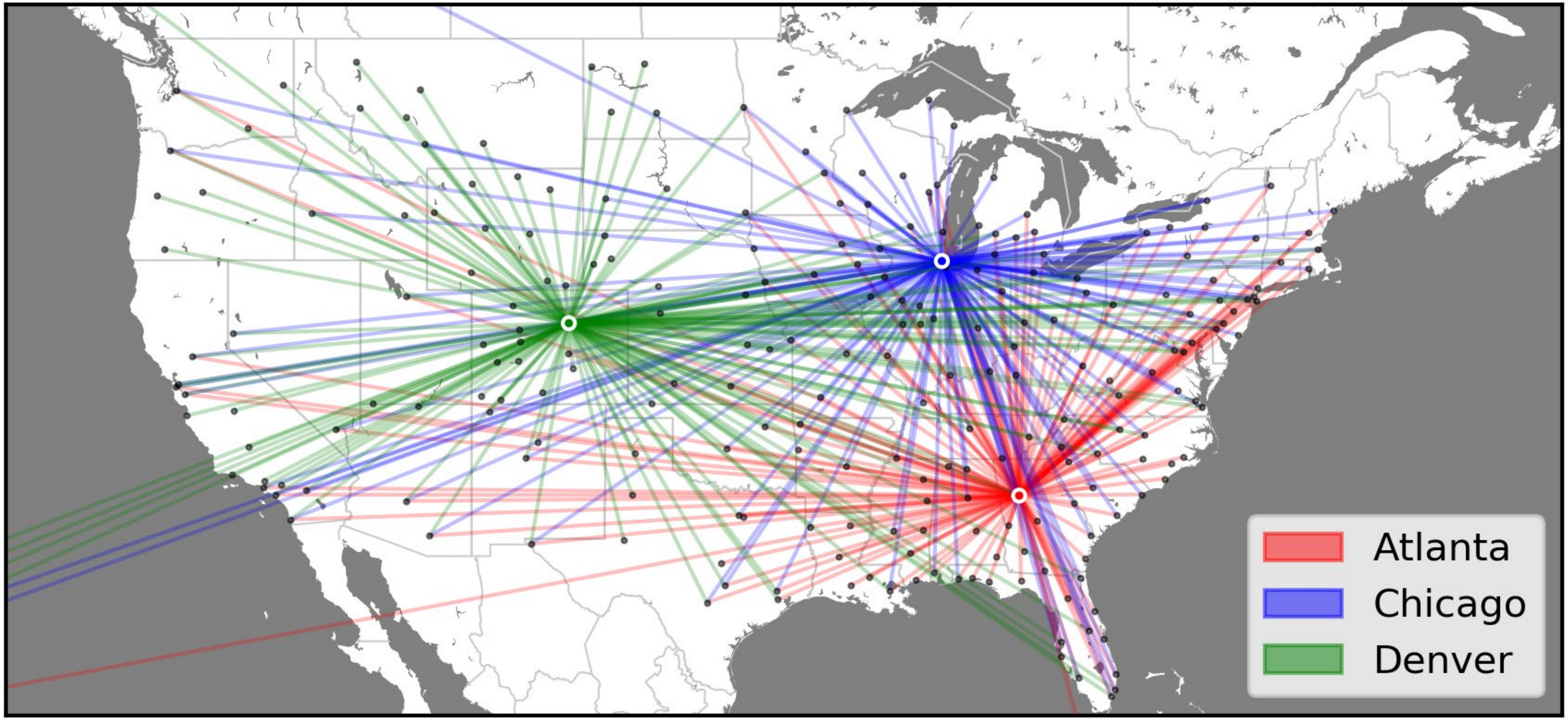
04. Hubs

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Real Networks are Heterogeneous (1/2)



Some nodes (and links) are much more important (**central**) than others!

Heterogeneity

universität koblenz
weiter:denken

- Heterogeneity in networks refers to the variability in the properties and roles of elements within the network, such as nodes and links
  - Air transportation networks (Airports, Flights)
  - Social networks (Individuals, Relationships)
  - The Web (Websites, Hyperlinks)

- This variability reflects the diversity present in complex systems and affects how networks function and interact

The Role of Nodes in Networks

- A key source of heterogeneity is the degree of nodes, indicating the number of connections each node has
  - major airports like Atlanta
  - influential social figures like Barack Obama, and
  - dominant websites like Google
- These nodes act as hubs in their respective networks, influencing flow and accessibility significantly

Network Centrality and Its Impact

universität koblenz
weiter:denken

- The importance of a node or a link is estimated by computing its **centrality**
- There are different ways of measuring **centrality** such as **Degree**, **Betweenness**, and **Closeness** centrality
- **Hubs**: High-Degree Nodes
  - These nodes, or hubs, are crucial for understanding network dynamics
  - Facilitate quick dissemination of information or resources across a network
  - Enhance the resilience of networks to certain types of failures but may also pose risks of rapid spread (e.g., diseases in social networks or vulnerabilities in the web)
  - In general: hubs contribute to some striking properties that characterize a broad variety of networks, affecting everything from information flow to network stability

Centrality Measures

- **Centrality:** measure of **importance** of a node
- **Measures**
  1. Degree
  2. Closeness
  3. Betweenness

Degree

- **Degree of a node:** number of neighbors of the node

$$k_i = \text{number of neighbors of node } i$$

- High-degree nodes are called **hubs**
- **Average degree of the network:**

$$\langle k \rangle = \frac{\sum_i k_i}{N} = \frac{2L}{N}$$

```
G.degree(2) # returns the degree of node 2
G.degree()  # dict with the degree of all nodes of G
```

Closeness

- **Idea:** a node is the more central the *closer* it is to the other nodes, on average

$$g_i = \frac{1}{\sum_{j \neq i} \ell_{ij}}$$

where ℓ_{ij} is the distance between nodes i and j

```
nx.closeness_centrality(G, node) # closeness centrality
                                 # of node
```

Betweenness (1/3)

- **Idea**: a node is the more central the *more often it is crossed by paths*

$$b_i = \sum_{h \neq j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}$$

σ_{hj} = number of shortest paths from h to j

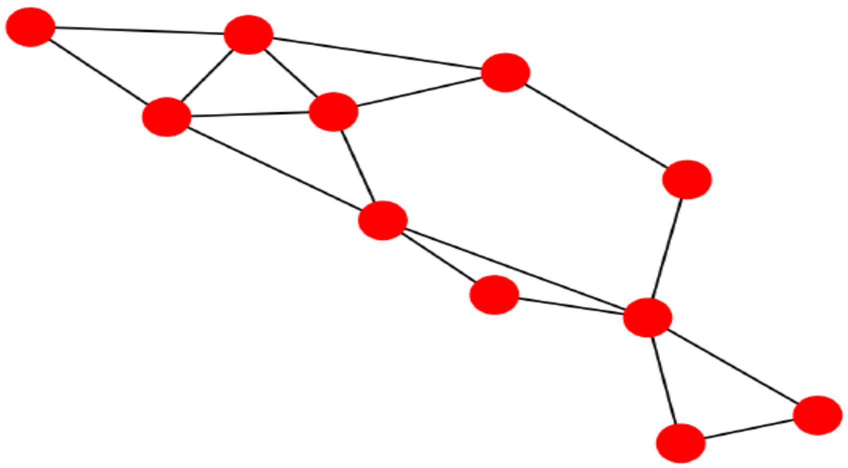
$\sigma_{hj}(i)$ = number of shortest paths from h to j running through i

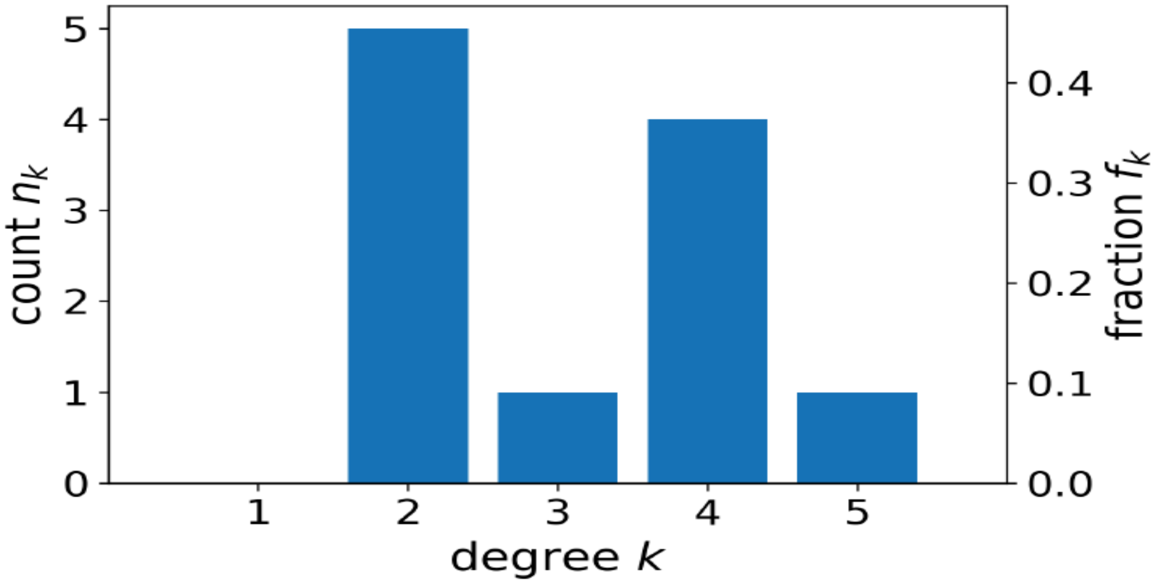
Centrality Distributions (1/3)

- For small networks, identifying the most important nodes or links is practical
- For large networks, this individual approach is impractical
- Solution: Adopt a statistical perspective
  - Rather than examining individual nodes and links, analyze groups (or classes) of nodes and links sharing similar properties

Centrality Distributions (2/3)

Histogram

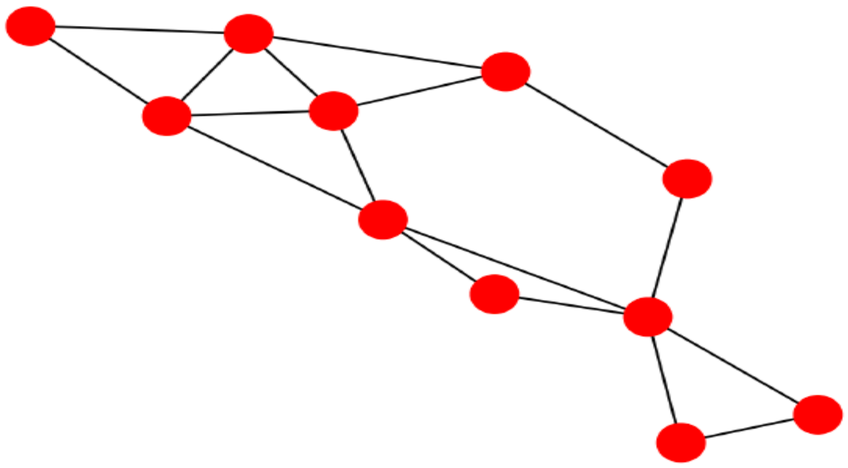




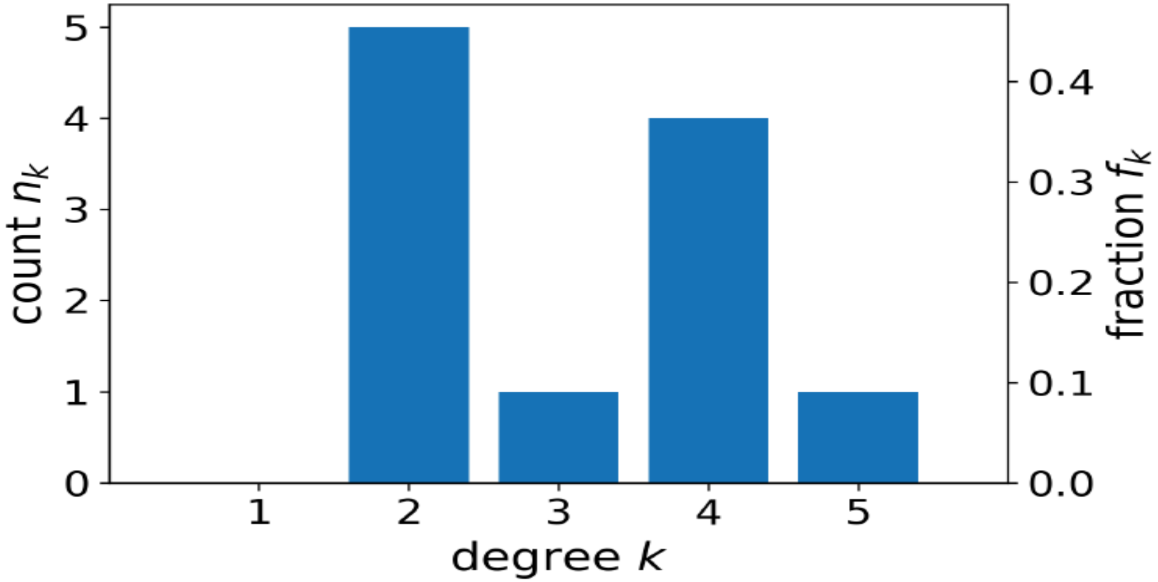
- n_k = **number of nodes with degree** k
- $f_k = \frac{n_k}{N}$ = **frequency of degree** k

Histogram representation of the degree distribution of a small network. First a list with the degree of each node is generated. The heights of the histogram bars are given by the counts of nodes with each degree k. The relative frequency of occurrence is defined as the fraction of all the nodes with degree k.

Centrality Distributions (3/3)



Histogram



- When $N \to \infty$, f_k becomes the **probability** p_k of having degree k
- p_k versus k is the **probability distribution** of node degree

Cumulative Distributions

- For non-integer variables (e.g., betweenness), we divide the range into intervals (bins) and count the values within each bin
- **Cumulative distribution *P(x)*:** The probability that the variable has values *larger* than a given value *x*
- **How to calculate it:** Sum frequencies of all intervals from *x* onward:

$$P(x) = \sum_{v \geq x} f_v$$

Degree Distributions (2/3)

universität koblenz
weiter:denken

To formally define the heterogeneity parameter κ (Greek letter "kappa") of a network's degree distribution, we need to introduce the *average squared degree* $\langle k^2 \rangle$, which is the average of the squares of the degrees:

$$\langle k^2 \rangle = \frac{k_1^2 + k_2^2 + \cdots + k_{N-1}^2 + k_N^2}{N} = \frac{\sum_i k_i^2}{N}. \tag{3.4}$$

The heterogeneity parameter can be defined as the ratio between the average squared degree and the square of the average degree of the network [Eq. (1.5)]:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle^2}. \tag{3.5}$$

For a normal or narrow distribution with a sharp peak at some value, say k_0, the distribution of the squared degrees is concentrated around k_0^2. Therefore $\langle k^2 \rangle \approx k_0^2$ and $\langle k \rangle \approx k_0$, yielding $\kappa \approx 1$. For a heavy-tailed distribution with the same average degree k_0, $\langle k^2 \rangle$ blows up because of the large degree of the hubs, so that $\kappa \gg 1$.

Degree Distributions (3/3)

- The **heterogeneity parameter κ** says how broad the distribution is:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle^2}$$

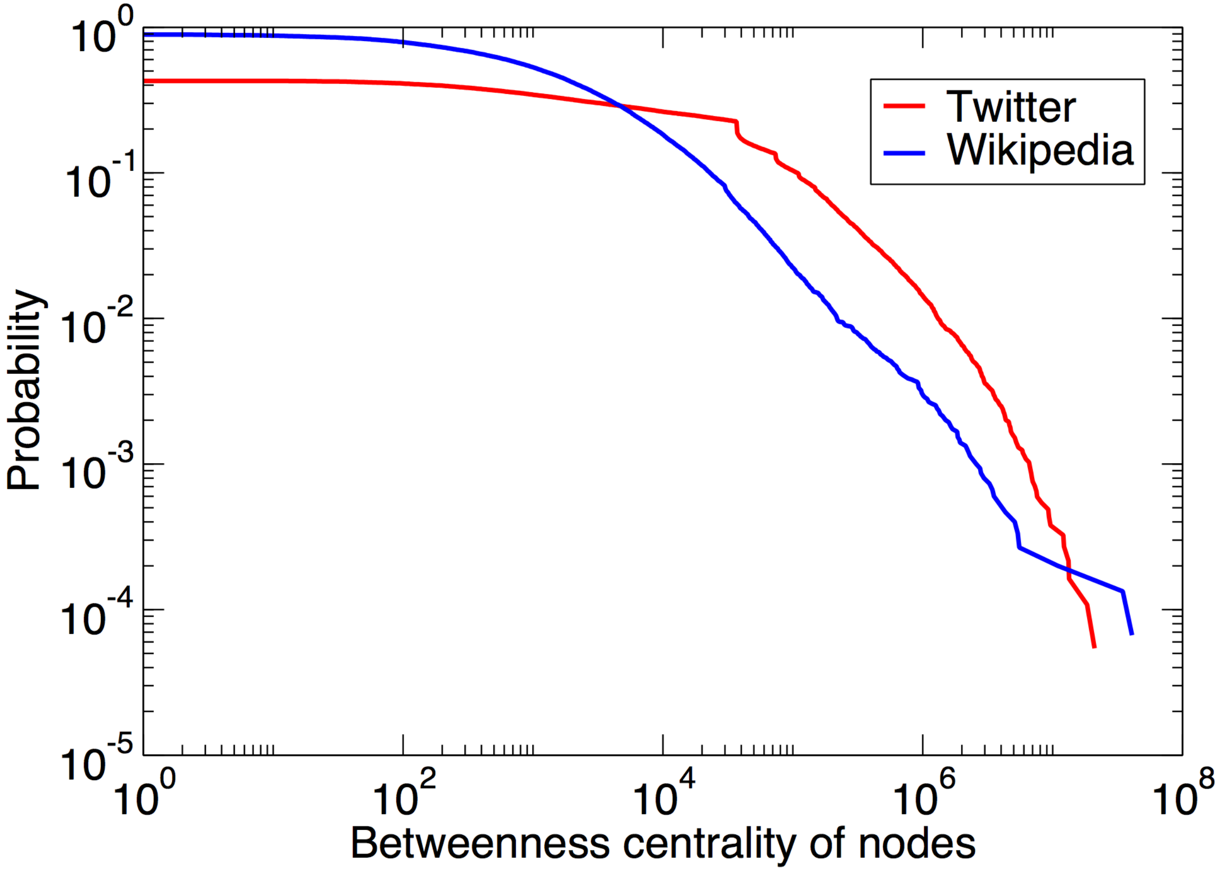
$$\langle k \rangle = \frac{\sum_i k_i}{N} = \frac{2L}{N}; \langle k^2 \rangle = \frac{\sum_i k_i^2}{N}$$

- If most degrees have the same value, say k_0 :

$$\langle k \rangle \approx k_0, \langle k^2 \rangle \approx k_0^2 \Longrightarrow \kappa \approx 1$$

- If the distribution is very heterogeneous: $\kappa \gg 1$

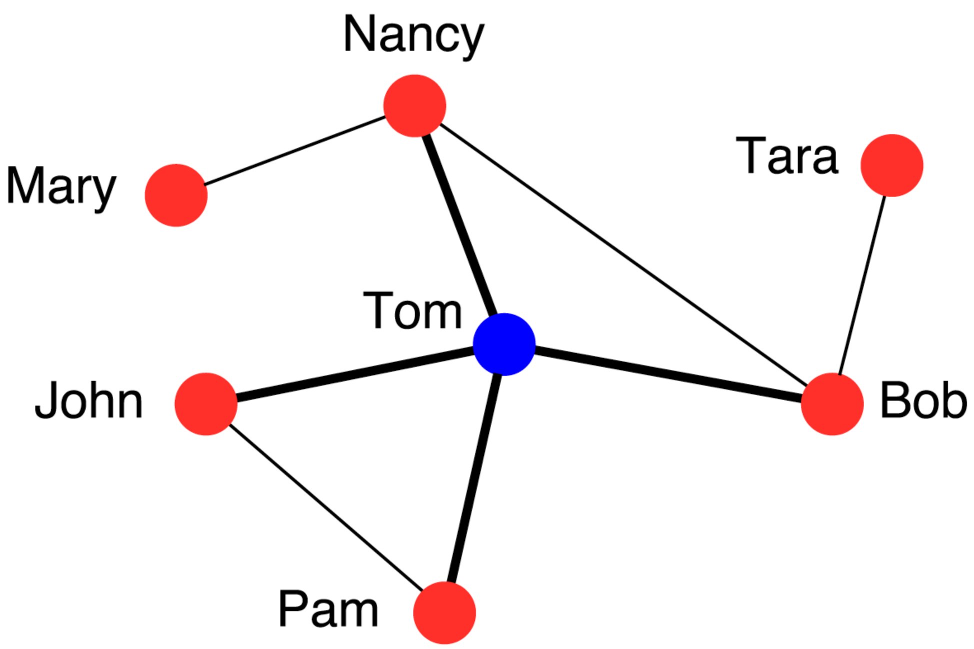
Betweenness Distributions



- Cumulative distribution of node betweenness centrality for Twitter and Wikipedia, shown on a log–log plot
- We considered both networks as undirected
- For Wikipedia we computed the betweenness only on its giant component, which includes over 98% of the nodes

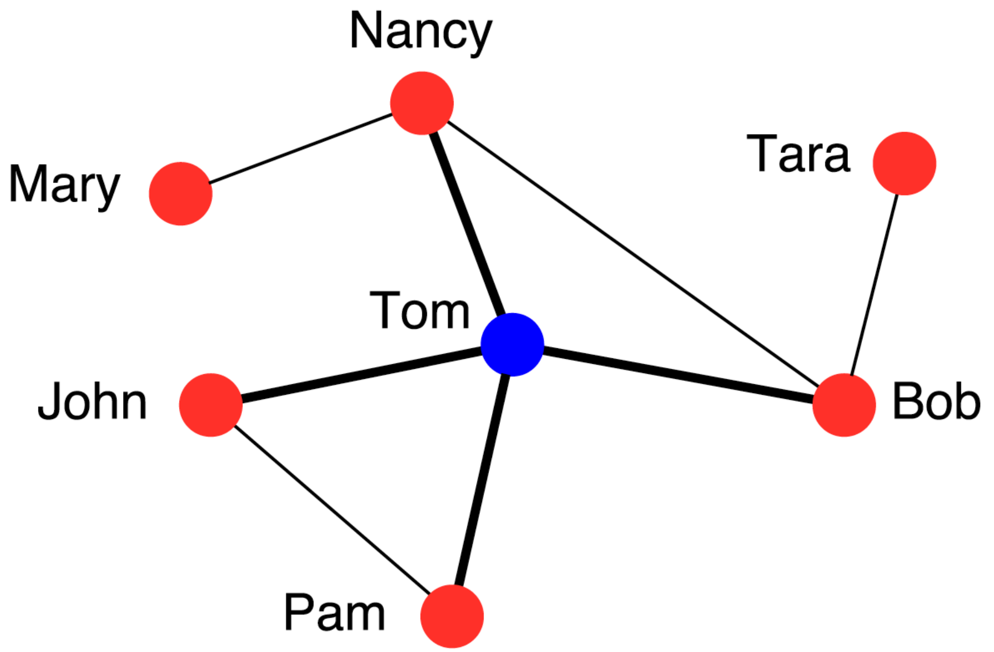
Heavy-tail distribution: the variable goes from small to large values

Friendship Paradox (1/4)



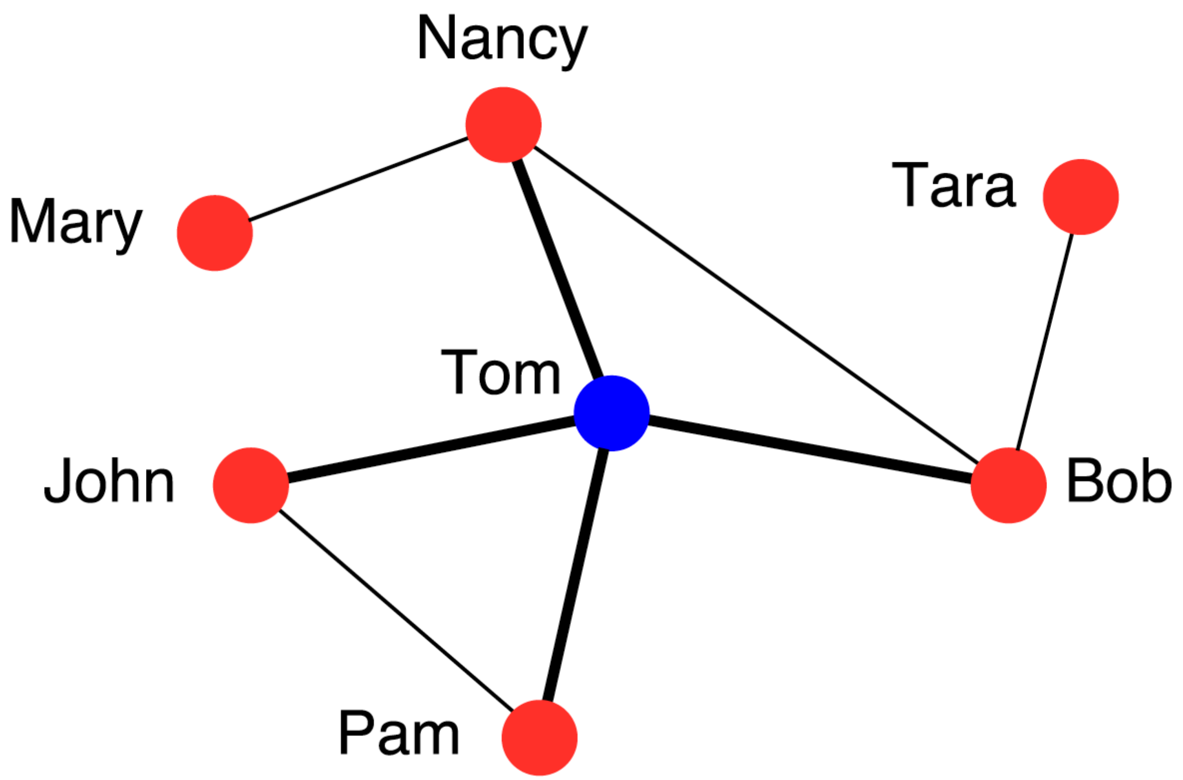
- If we pick a person (node) at random, everyone — including Tom — has equal probability of being selected
- If we pick a friendship (link) at random and then look at one of the people it connects, Tom is more likely to be chosen because he has more connections
- Nodes with more links are overrepresented when sampling through connections — this is the core of the **friendship paradox**

Friendship Paradox (2/4)



- When you follow a random link in the network, you are more likely to reach a highly connected node (a hub)
- This is because hubs are connected to many links, so they are more likely to be reached by random traversal
- In other words, the more friends someone has, the more likely you are to encounter him/her by following a random friendship

Friendship Paradox (3/4)



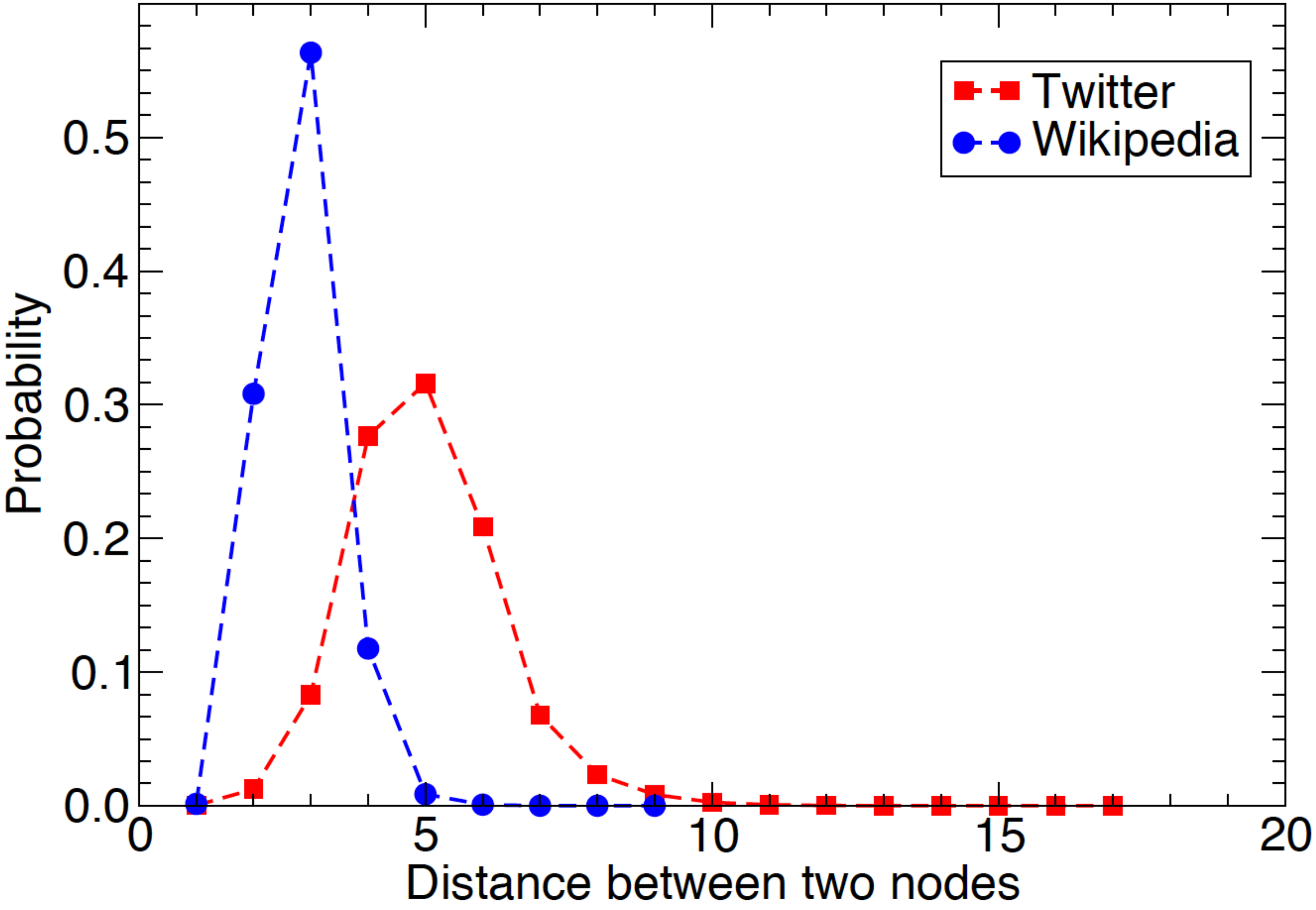
- Average degree of a node = 2.29
- Average degree of the neighbors of a node = 2.83 > 2.29
- Our friends have more friends than we do, on average: **friendship paradox**

Ultra-Small Worlds (1/2)

- In real networks, many shortest paths go through hubs
- **Example**: air travel
  - Small airports A and B might not be directly connected,
  - But you can still travel from A to B through a hub airport like C
- In networks with hubs, shortest paths become extremely short — this is known as the **ultra-small world** effect
  - Hubs dramatically reduce the average distance between nodes

Ultra-Small Worlds (2/2)

Shortest-path length distribution



- Ultra-small worlds
- The distributions of distance between nodes are peaked at very low values for both Twitter and Wikipedia
- This is due to the presence of hubs, which shrink the distance between most pairs of nodes, as shortest paths run through them
- Distances are computed by ignoring the direction of the links

Robustness (1/5)

- A system is **robust** if it continues to function even when some components fail
- **Question**: How do we measure the *robustness* of a network?
- **Answer**: We simulate failures by removing nodes or links and observe the structural impact
- **Key point:** *connectedness*
- If the Internet were not connected, it would be impossible to transmit signals (e.g., emails) between routers in different components

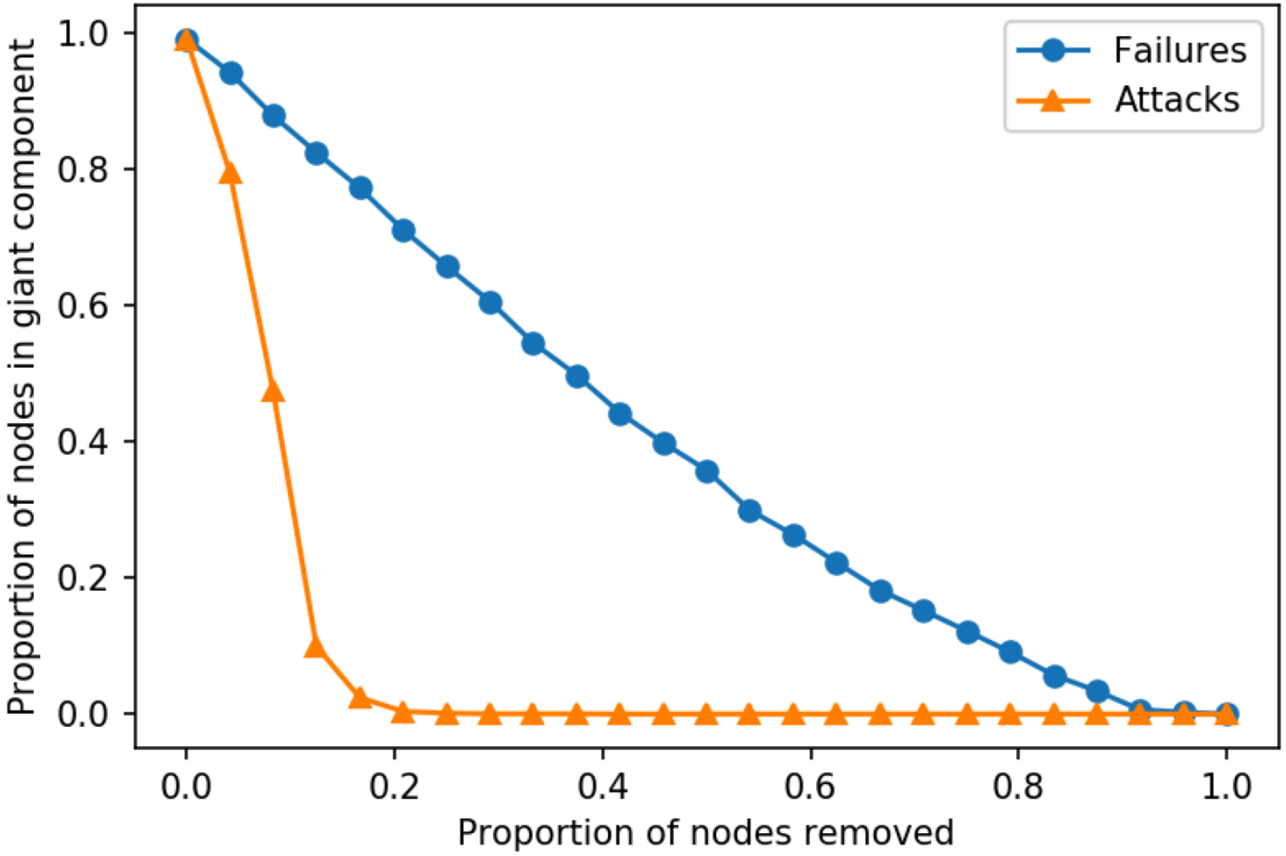
Robustness (2/5)

- **Robustness test:** Measure how network connectivity changes as we remove more nodes
- **Method:** Track the size S of the largest connected component as a function of the fraction of nodes removed
- We suppose that the network is initially connected: there is only one component and $S = 1$
- As more and more nodes (and their links) are removed, the network is progressively broken up into components and S goes down
- This helps us understand how resilient the network is to failures or attacks

Robustness (4/5)

- **Two types of node removal strategies:**
  1. **Random failures:** Nodes are removed randomly — each has the same chance of failure
  2. **Attacks:** hubs are deliberately targeted — the more connections a node has, the more likely it is to be targeted
- In the random case, we remove a fraction f of nodes, chosen at random
- In the attack case, we remove the top f-fraction of highest-degree nodes, starting from the most connected

Robustness (5/5)



Conclusion: Real networks are robust to random failures, but very vulnerable to targeted attacks on high-degree nodes

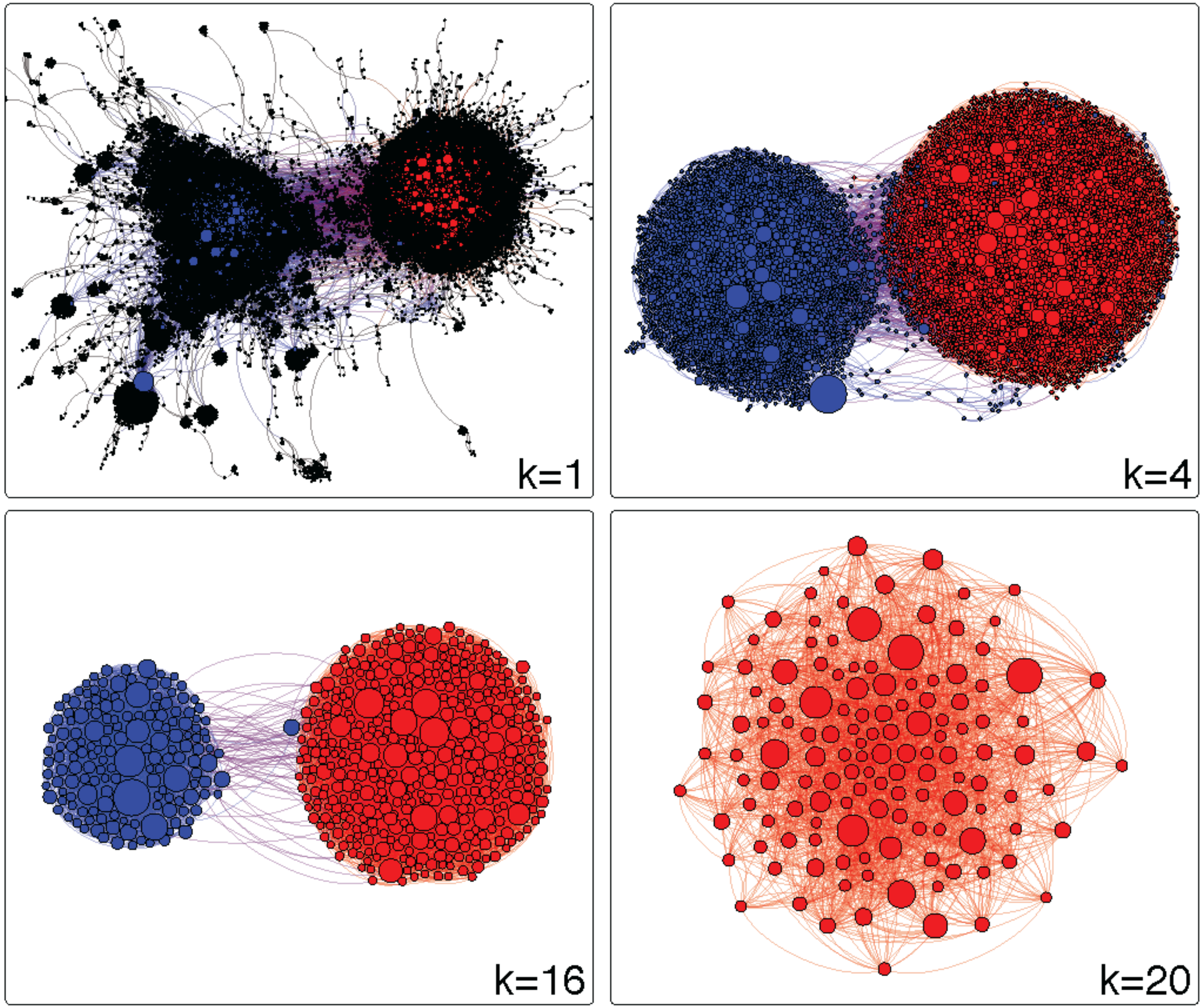
Core Decomposition (1/3)

- **Core:** A dense region in the network made of high-degree nodes
- **Core decomposition:** A method to find increasingly dense subgraphs by removing low-degree nodes step by step
- A **k-core** is what remains after removing all nodes with degree $\leq k-1$
- ***k*-core decomposition procedure:**

1. Start with k=0
2. Remove all nodes with degree k, repeatedly, until none remain
3. The set of removed nodes is the ***k*-th shell**, while the remaining ones form the **(*k*+1)-core**
4. If nodes remain, increase k and repeat

Core Decomposition (2/3)

- We apply k-core decomposition to a Twitter political retweet network
- Starting at k=1, the full network is included
- As k increases, nodes with fewer than k neighbors are removed
- The result:
  - The network becomes smaller and denser
  - Only highly connected nodes remain
- At k=20, the core consists only of red nodes (conservative accounts), each connected to at least 20 neighbors



Network Theory and Dynamic Systems

05. Strong and Weak Ties

SOSE 2025

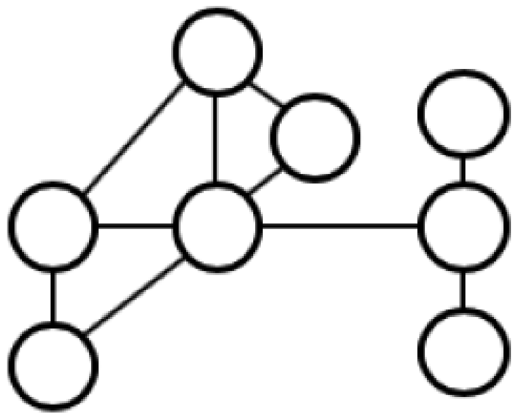
Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

"Strength of Weak Ties" – A Hypothesis from Sociology

Based on interviews by **Mark Granovetter**:

- People who recently changed jobs were asked how they found those new opportunities
- Many reported that they got help **not from close friends**, but from **acquaintances** (weak ties)
- This was surprising, since it's often expected that **close friends** would be more willing to help
- Granovetter explained this using **network structure** and **social dynamics**:
  - Weak ties connect us to **different social circles**, giving access to **new information and opportunities**
- *Example:* A close friend likely knows the same people you do, but an acquaintance might work in a different company or industry



Triadic Closure: Why a Friend of a Friend Often Becomes Your Friend

universität koblenz
weiter:denken

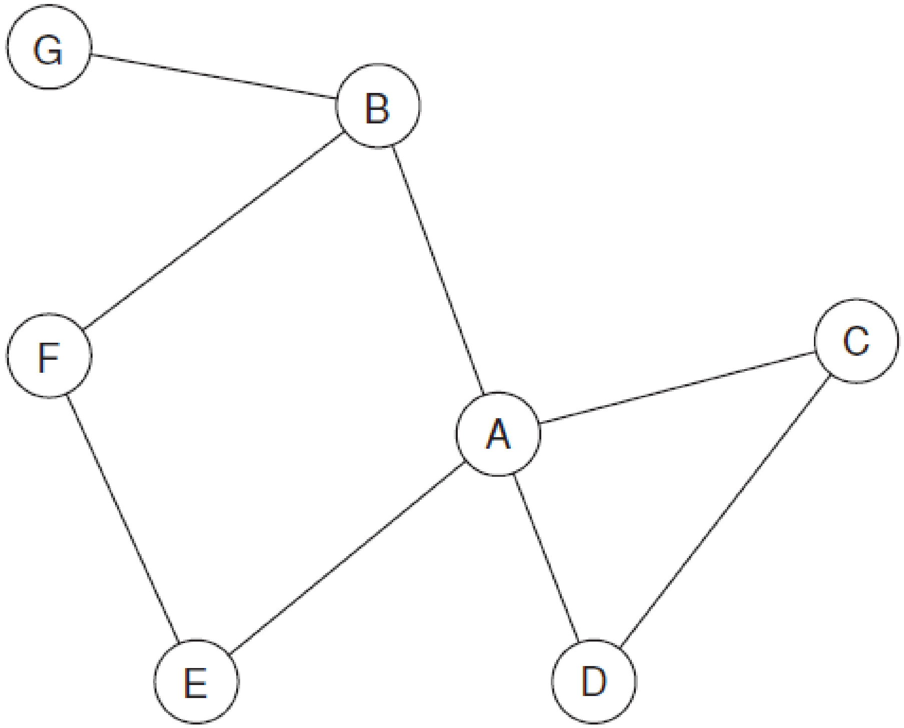
Core Idea:

- Social networks are not static — they **change and grow** over time
- When two people have a **mutual friend**, there's a **higher likelihood** that they will also become friends
- This pattern is called **triadic closure**, and it's a common principle in how social networks *develop*

Triadic Closure: An Example (1/2)

universität koblenz
weiter:denken

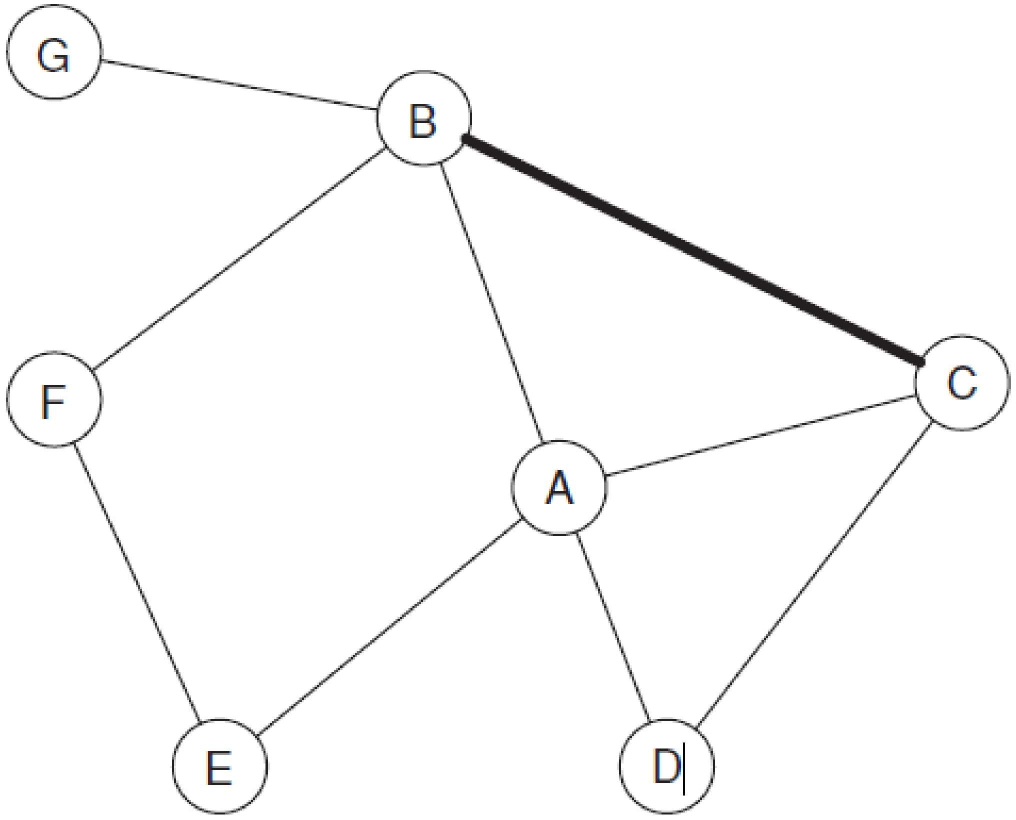
- Imagine a small network with Alice (A), Bob (B), and Charlie (C)
- Alice is friends with both Bob and Charlie
- According to triadic closure, there's a higher chance that Bob and Charlie will become friends too — because they both know Alice



(a)

Triadic Closure: An Example (2/2)

- When Bob and Charlie form a connection, the triangle between Alice, Bob, and Charlie is complete
- This triangle is a visual sign of the triadic closure principle — when mutual connections increase the chance of new links forming
- *This triangle structure is a common pattern in real social networks*



(b)

Triadic Closure: Clustering Coefficient (1/2)

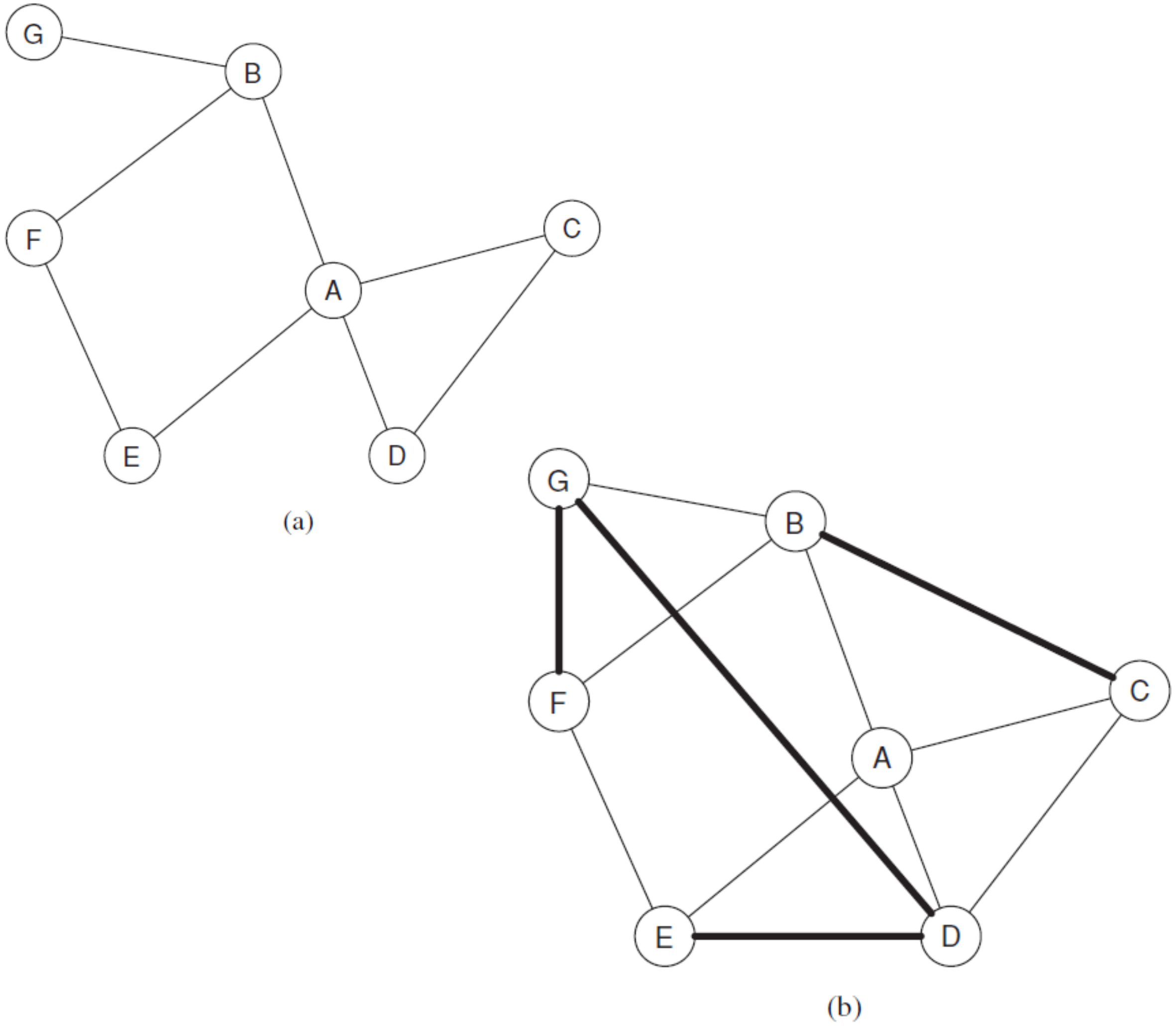
- Clustering Coefficient
  - A measure of triadic closure in a network
  - Indicates the likelihood that two friends of a node are also friends with each other
- How is it calculated?
  - The local clustering coefficient for a node v is calculated using (see also Lecture 03-Small Words)

$$C(v) = \frac{2 \times e(v)}{k(v) \times (k(v) - 1)}$$

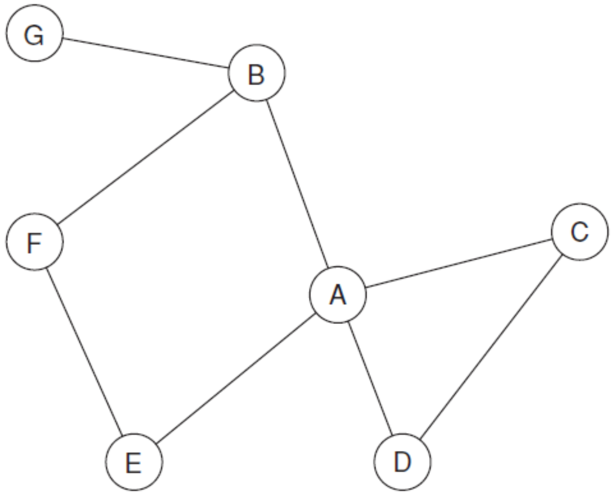
    - $e(v)$ is the number of edges between the neighbors of v (number of triangles involving v)
    - $k(v)$ is the degree of node v (i.e., the number of neighbours node v has)
- Range: 0 (no friends connected) to 1 (all friends connected)

Triadic Closure: Clustering Coefficient (2/2)

- Example
  - Node A's clustering coefficient is 2/12 =1/6 (only the C-D edge connects its friends)
  - The coefficient increases to 6/12 = 1/2 (with B-C, C-D, and D-E edges)
- Interpretation
  - Higher coefficient suggests stronger triadic closure in a node’s neighborhood



Why Triadic Closure Happens



(a)

1. More Chances to Interact

- A shared friend (e.g., Alice) gives Bob and Charlie opportunities to meet
- Spending time together through this connection can lead to new friendships

2. Builds Trust

- If Alice trusts both Bob and Charlie, this trust can transfer between them
- Knowing there's a mutual connection makes people more open and trusting

3. Social Pressure

- Alice may feel responsible for introducing Bob and Charlie
- It can be uncomfortable if two of a person’s friends don’t get along, creating pressure to maintain harmony

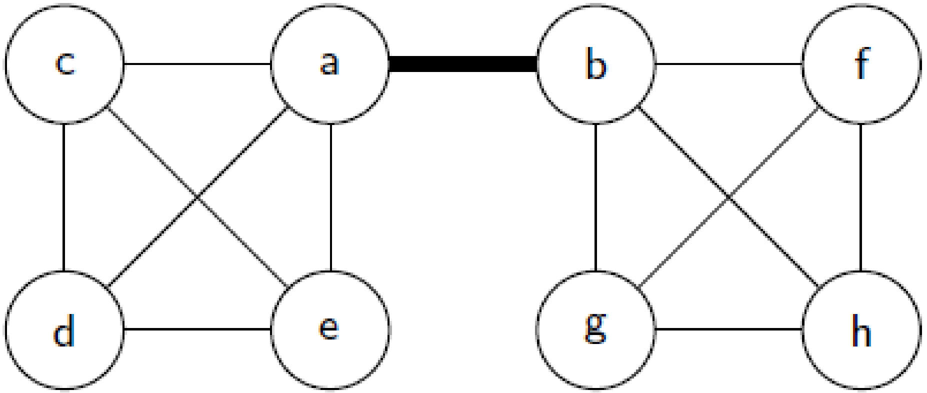
The Strength of Weak Ties

Based on key findings from Mark Granovetter’s research:

- Weak ties refer to links with acquaintances, not close friends
- Granovetter found that people often find new job opportunities through these weak ties

Acquaintances connect us to different social circles and new information

Bridges



In this graph, the edge *{a, b}* is a bridge

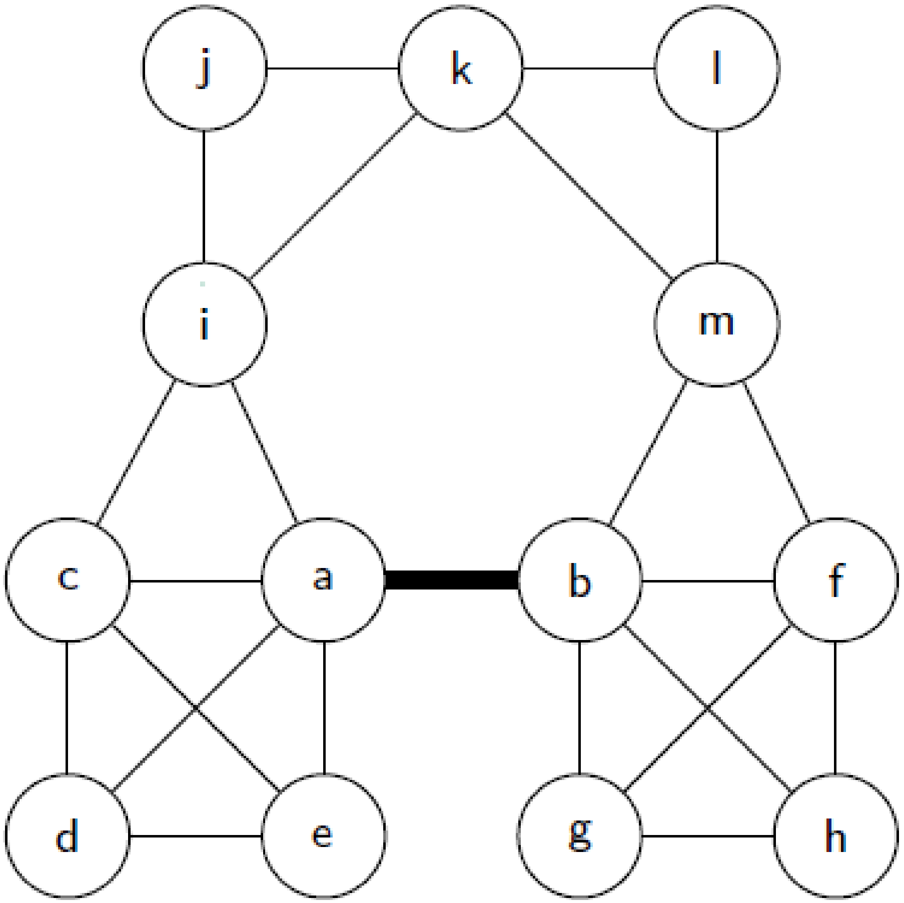
Bridge

- An edge in a graph is called a bridge if by dropping it, results the graph being split into two different components (disconnected parts)

Formally

- Let $G = (V, E)$ be an undirected grap
- An edge $\{a, b\} \in E$ is a *bridge* if a and b are in the same component in a G but different in $G' = (V, E \setminus \{\{a, b\}\}$

Local Bridges (1/2)



In this graph, the edge *{a, b}* is a local bridge

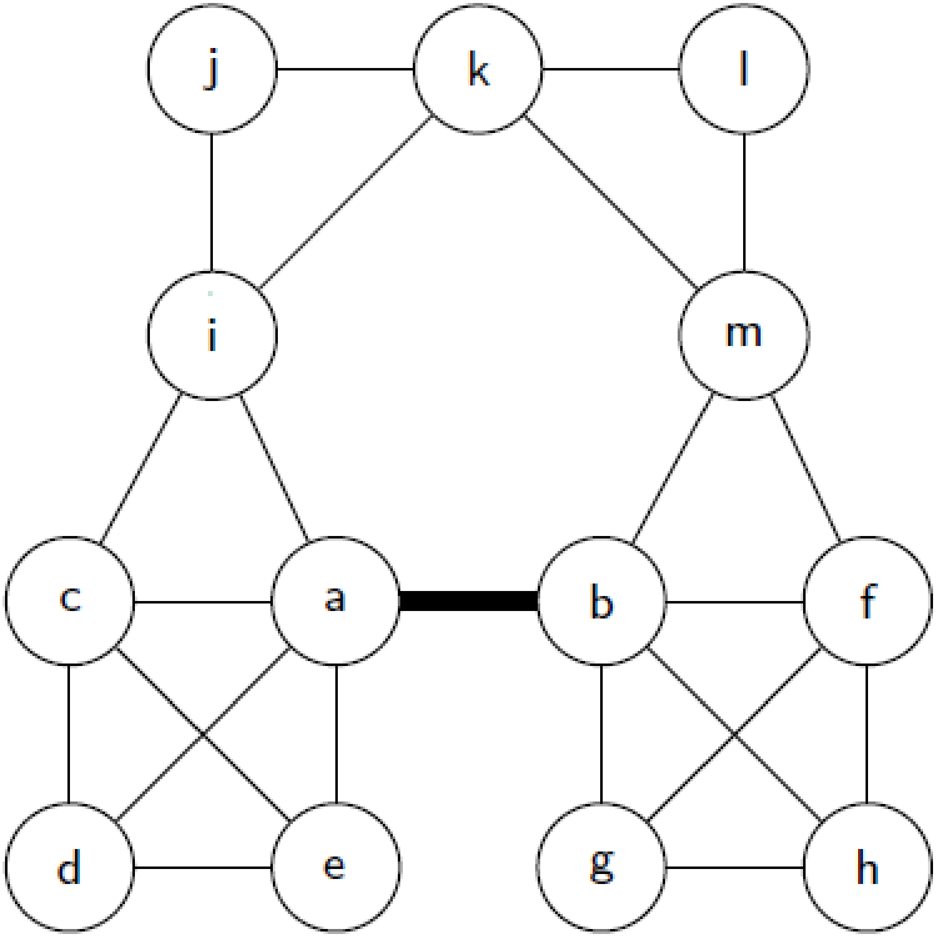
A local bridge is an edge that increases the distance between two nodes if removed but doesn't completely disconnect them

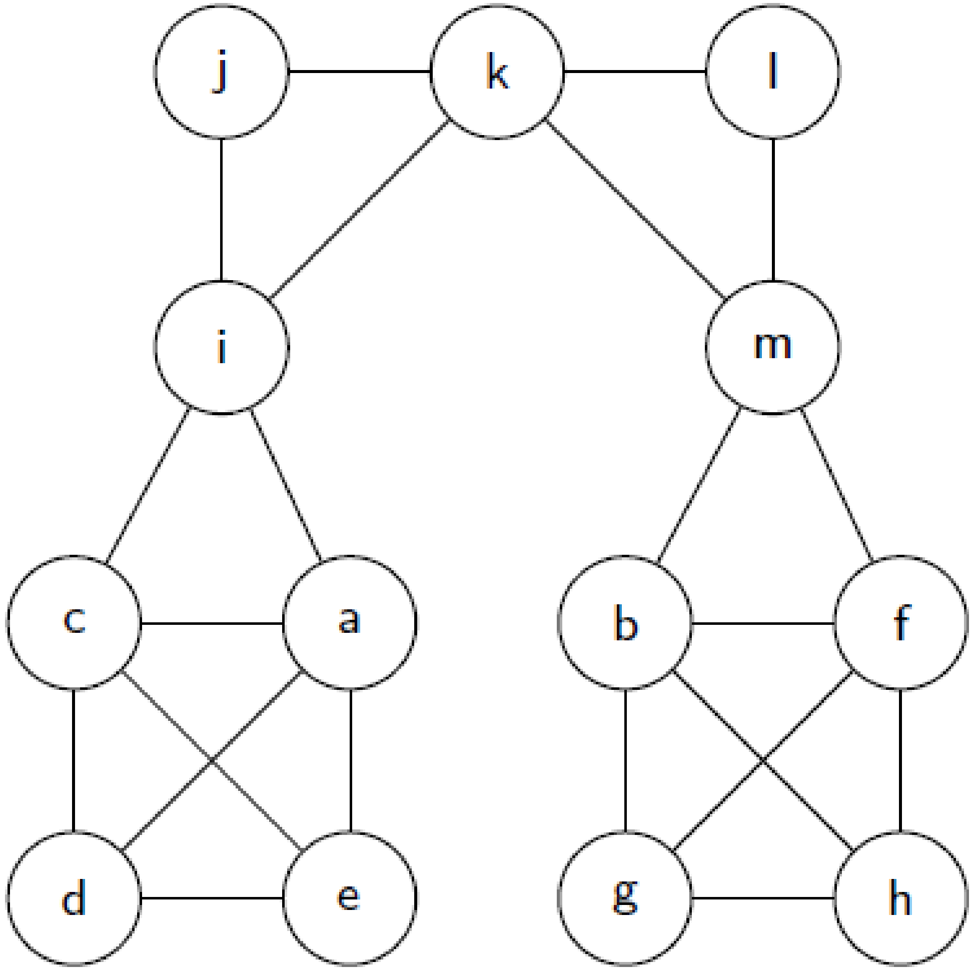
Formally

- Let $G = (V, E)$ be an undirected graph
- An edge $\{a, b\} \in E$ is a *local bridge* if there is no vertex $c \in V$ that is a neighbour of both a and b

Local Bridges (2/2)

universität koblenz
weiter:denken

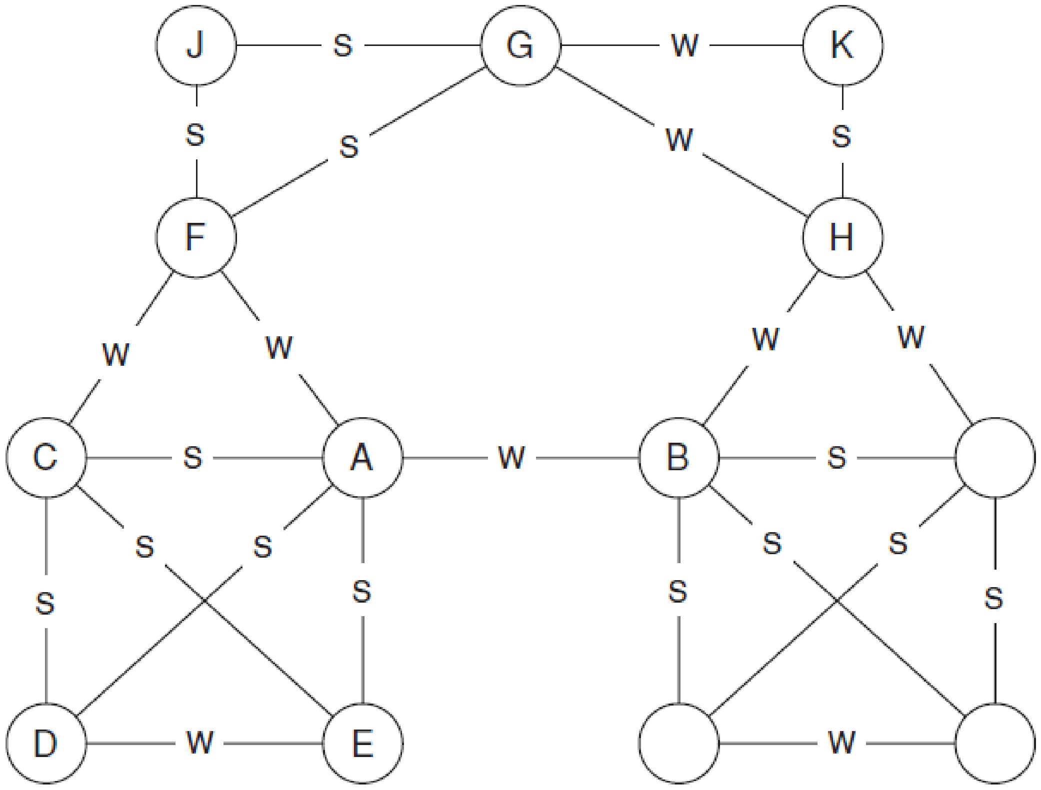




Here, *{a, b} is a local bridge with a span of 4*

Strength in Ties

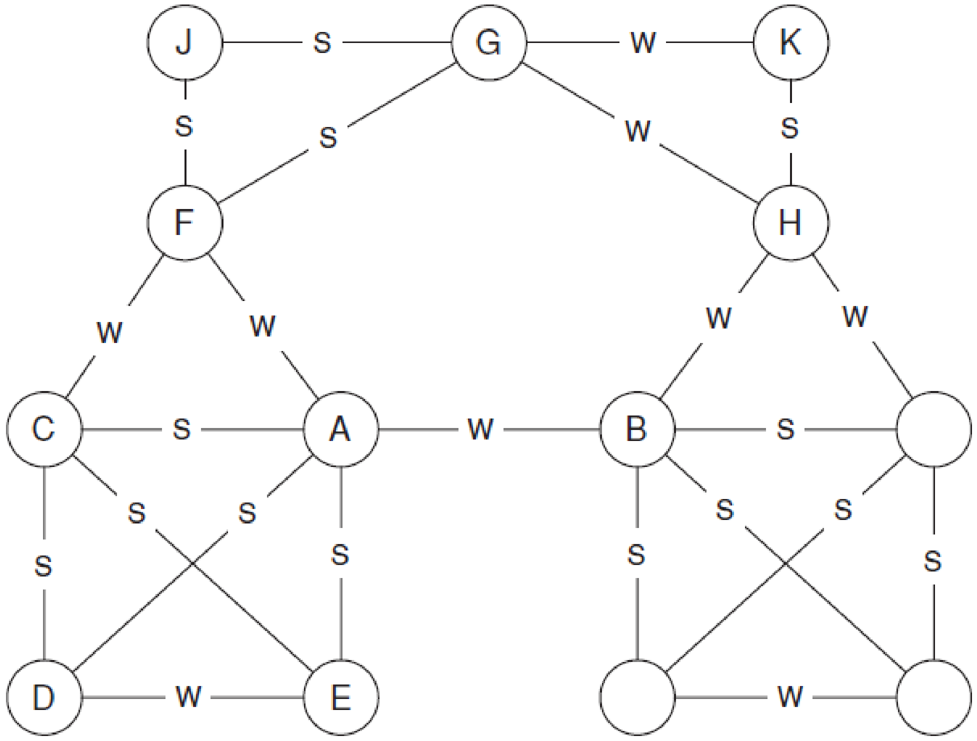
▪ In social networks, edges represent relationships between individuals
▪ These edges differ in strength:
  ○ Strong ties = close friendships
  ○ Weak ties = acquaintances
▪ Weak ties often connect different parts of the network, helping information spread widely



The Strong Triadic Closure Property (STCP)

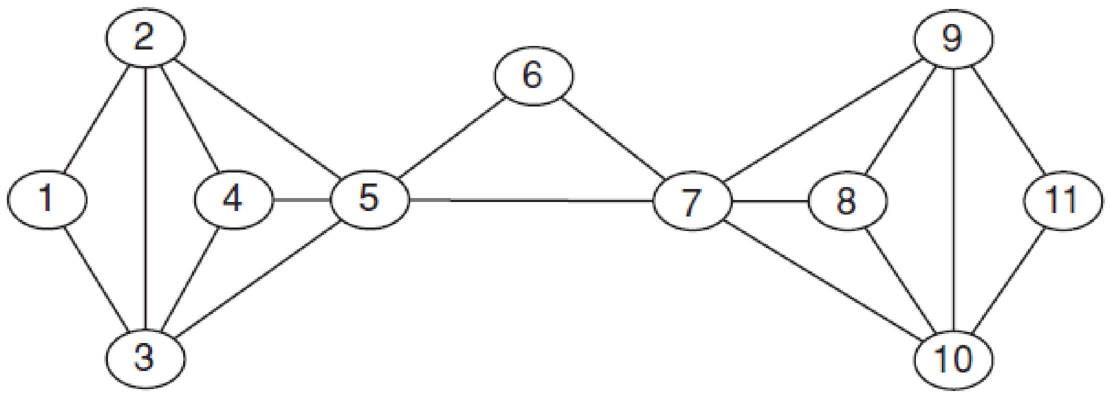
universität koblenz
weiter:denken

- Triadic closure means that if two people have a common friend, they are likely to become friends too
- The Strong Triadic Closure Property adds precision: it looks at whether the connections are strong or weak ties
- A node violates this property if it has strong ties to two people, who are not connected to each other at all (by any tie)
- Example: If Alice is very close to both Bob and Charlie, we expect Bob and Charlie to know each other — otherwise, the closure is broken



Girvan–Newman Method (1/3)

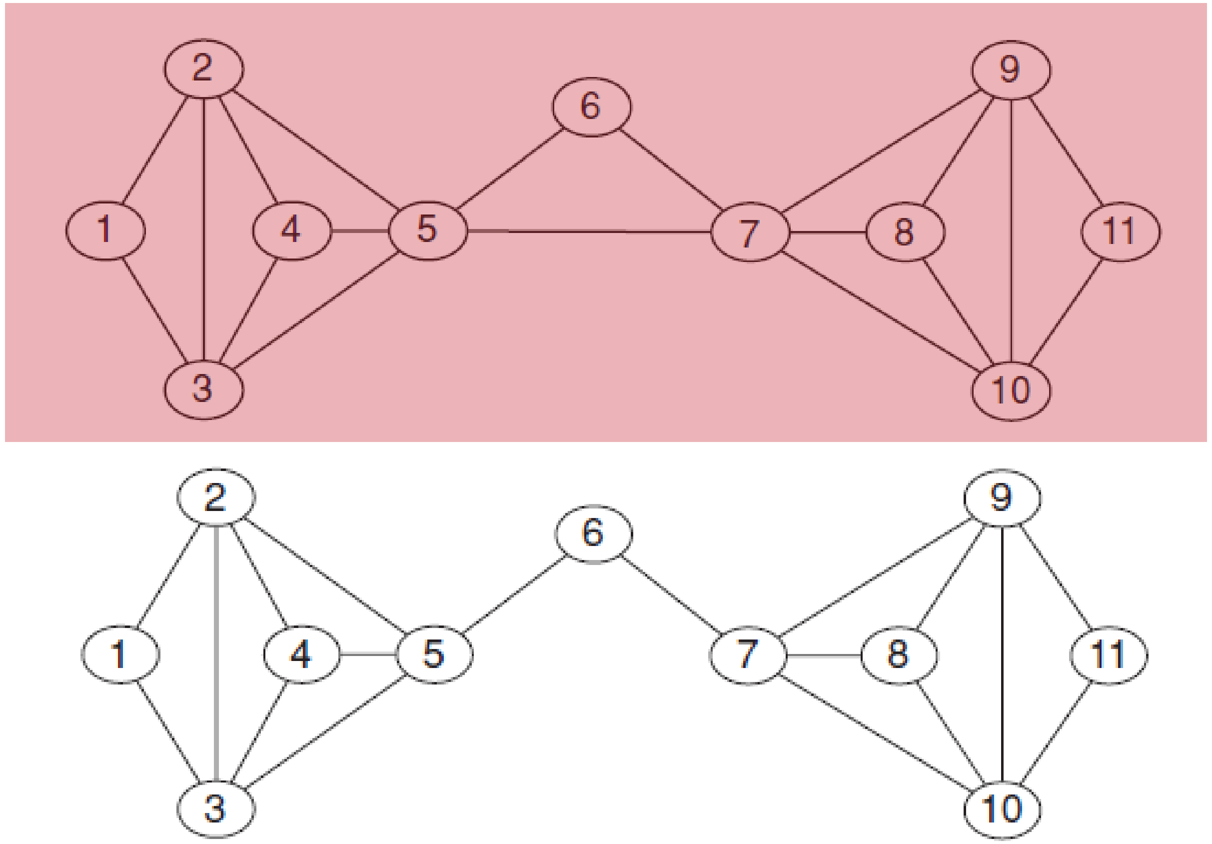
universität koblenz
weiter:denken



A network can display tightly-knit regions even when there are no bridges or local bridges along which to separate it.

- The Girvan–Newman Method is a pivotal algorithm in network analysis, particularly for uncovering community structures within complex networks
- The method involves iteratively removing edges of high betweenness centrality
- High-betweenness edges are those that facilitate the most traffic flow along shortest paths between nodes

Girvan–Newman Method (2/3)



A network can display tightly-knit regions even when there are no bridges or local bridges along which to separate it.

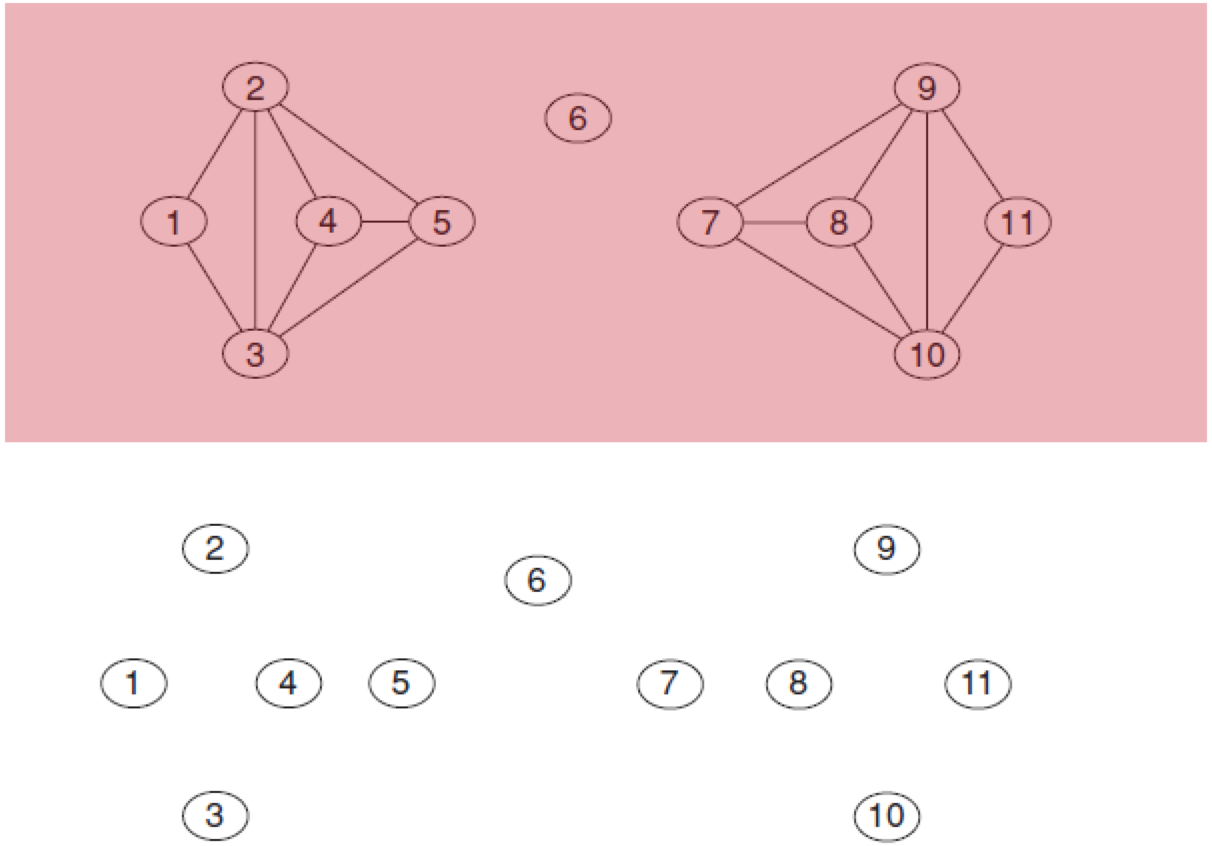
Purpose: Detect community structure by removing important edges.

How It Works:

1. Find edges with the highest betweenness centrality
   - These edges lie on many shortest paths between nodes
2. Remove these edges
   - This may split the network into separate parts (communities)
3. Recalculate betweenness centrality for the new graph
4. Repeat the process
   - Continue removing high-centrality edges until the graph breaks into meaningful clusters

Note: This method works even when there are no clear bridges, by identifying structural weak points using centrality

Girvan–Newman Method (3/3)



A network can display tightly-knit regions even when there are no bridges or local bridges along which to separate it.

- The algorithm gradually reveals the structure of the network by repeatedly removing central edges
- This process helps uncover tightly-knit regions (communities) that were not obvious before
- In the end, the network is split into smaller, well-connected groups, showing the underlying community structure

Network Theory and Dynamic Systems

06. Graph-Based View of the Web

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Web Graph vs. Host Graph

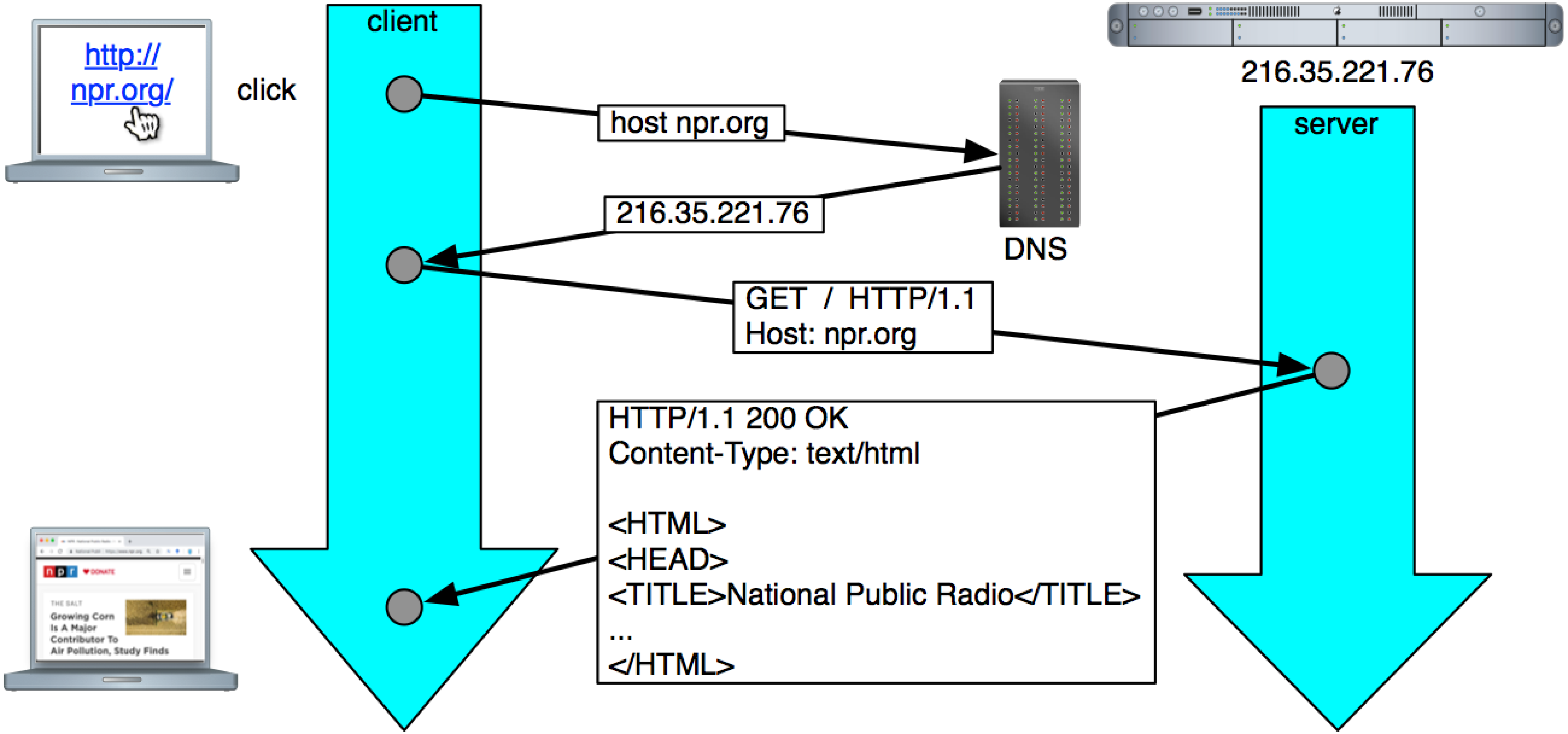
universität koblenz
weiter:denken

	Web Graph	Host Graph
Nodes	Individual **web pages** (e.g., https://example.com/page1)	**Websites** or **hosts** (e.g., example.com)
Edges	**Hyperlinks between pages**	**Aggregate links** between hosts: at least one hyperlink between any pages
Granularity	**Fine-grained** — shows exact page-to-page connectivity	**Coarser** — shows site-level or domain-level connectivity
Size	Much **larger**, due to many pages per site	**Smaller** — nodes represent hosts, not individual pages
Use case	Used for **detailed link analysis**, PageRank, crawling strategies	Used for **visualizing site-to-site relations**, host-level ranking
Example link	page1.html → page2.html	example.com → anotherdomain.org

Web Graph ≠ The Internet

	Internet	Web Graph
What it is	The global **network of computers and devices**	A **graph structure of webpages** connected by **hyperlinks**
Nodes	Physical **machines**: routers, servers, computers	**Web pages** (identified by URLs)
Links/Edges	**Physical/data connections** (e.g., Ethernet, fiber)	**Hyperlinks** (from one page to another)
Protocols	Includes TCP/IP, DNS, etc.	Uses **HTTP/HTTPS** protocols
Scope	Covers all digital communication infrastructure	Subset: Only about the **content** and **structure of the Web**
Examples	Email servers, DNS infrastructure, FTP servers	Google linking to Wikipedia, Wikipedia pages linking to each other

How the Web Works



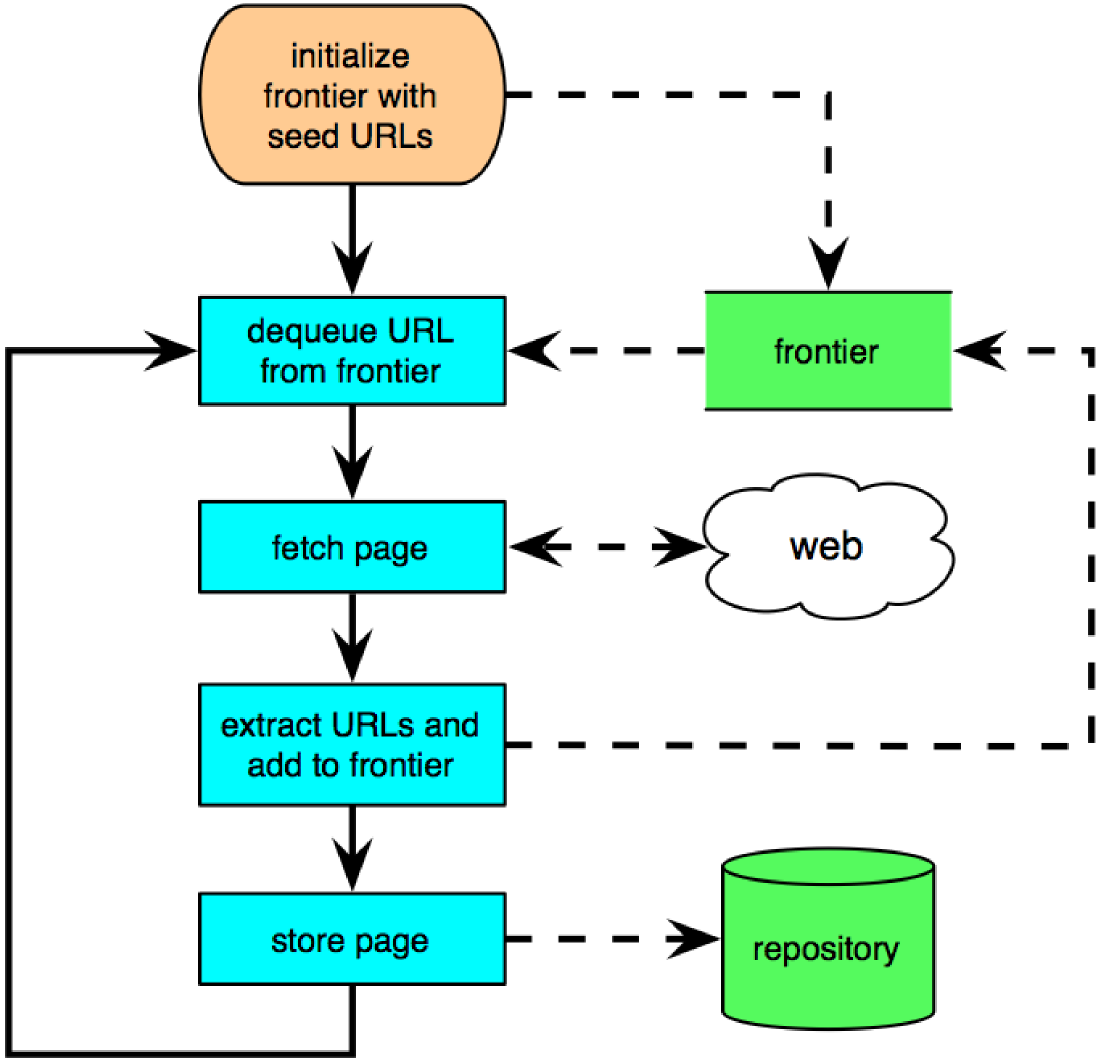
Web Crawlers (1/6)

- A **Web client** is any program that retrieves information from the Web by using the HTTP protocol to connect to a Web server
  - The most common example is the Web browser, used for interactive browsing by humans
- **Web crawlers** (also called *spiders* or *bots*) are automated programs that systematically download Web pages
  - Their main role is in search engines, where they gather and index content to make it searchable

Web Crawler (4/6)

universität koblenz
weiter:denken

- A crawler performs a breadth-first traversal of the Web graph, similar to the BFS (Breadth-First Search) algorithm
- It begins with a set of high-quality seed pages
- It maintains a frontier — a queue of unvisited URLs
- Crawling process:
  - Dequeue a URL from the frontier
  - Fetch the corresponding Web page
  - Extract all hyperlinks from the page
  - Enqueue the new (unvisited) links into the frontier

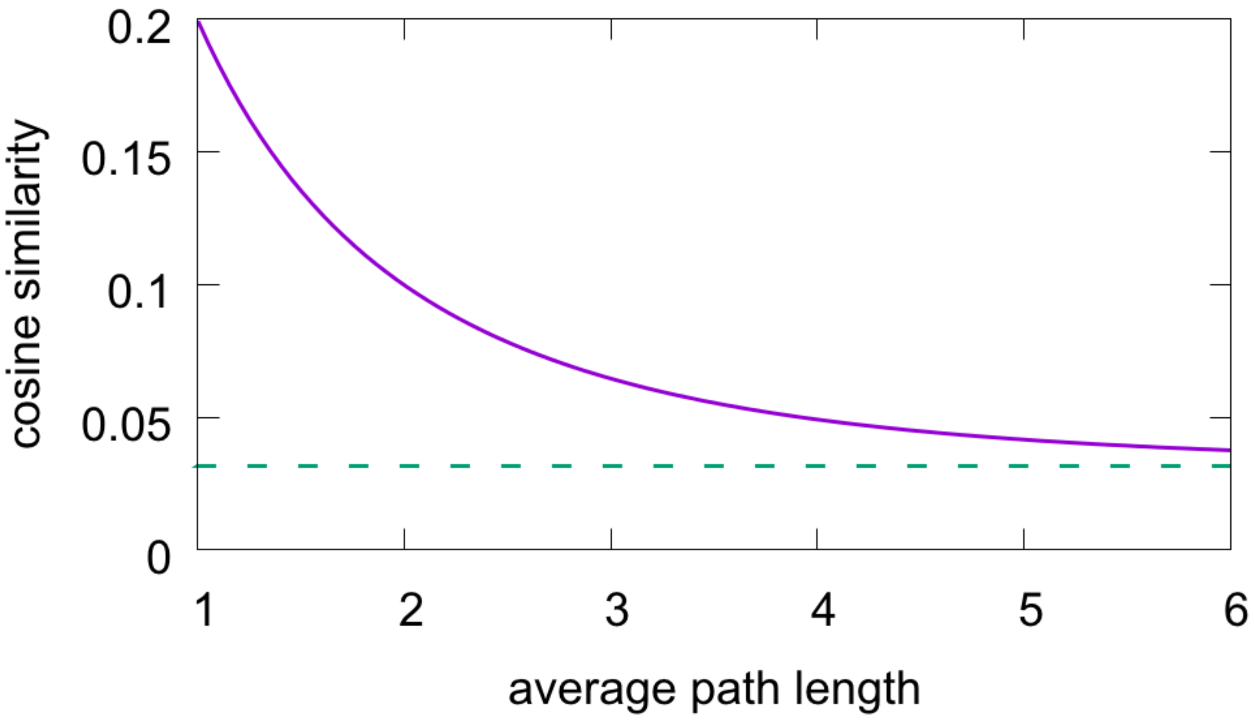


Cosine Similarity

universität koblenz
weiter:denken

- **Vector space model**: textual content of each page is represented as a vector
  - One dimension for each term in vocabulary (high-dimensional space!)
    - ✓ Remove or give lower weights to terms that are very common and therefore not meaningful
  - Or use deep learning to embed pages in lower-dimensional Euclidean space
- Measure similarity between two pages based on angle between vectors

$$\cos(\vec{d_1}, \vec{d_2}) = \frac{\vec{d_1}}{\parallel \vec{d_1} \parallel} \cdot \frac{\vec{d_2}}{\parallel \vec{d_2} \parallel} = \frac{\sum_t w_{d_1,t} w_{d_2,t}}{\sqrt{\sum_t w_{d_1,t}^2} \sqrt{\sum_t w_{d_2,t}^2}}$$



PageRank (1/4)

- **PageRank** is a centrality measure designed for directed networks, such as the Web graph
- It estimates the importance or prestige of nodes (e.g., Web pages), based on their link structure
- Especially useful when multiple pages are equally relevant by content — helps search engines rank them more effectively
- Introduced in 1998 by Sergey Brin and Larry Page as a core innovation behind Google's search engine
  - Pages gain importance by being linked to by other important pages
  - Captures the idea of recursive prestige
    - a page is important if it is linked to by other important pages

PageRank (3/4)

- **Random surfer model:** browse the Web at random
  - A random link is clicked from each page to get to the next
- Random walk model modified with random jumps (**teleportation**)
  - At each step, with probability α, stop browsing and start new session form a random page
- Recursive definition
- PageRank is conserved, neither created nor destroyed: $\sum_i R(i) = 1$

Power method to calculate PageRank:

- Initialize each node with

$$R_0 = 1/N$$

- At each iteration *t*, loop over nodes and update PageRank of each node *i* via this recursive equation:

$$R_t(i) = \boxed{\frac{\alpha}{N}} + \boxed{(1-\alpha) \sum_{j \in pred(i)} \frac{R_{t-1}(j)}{k_{out}(j)}}$$

probability to land on *i* by teleportation

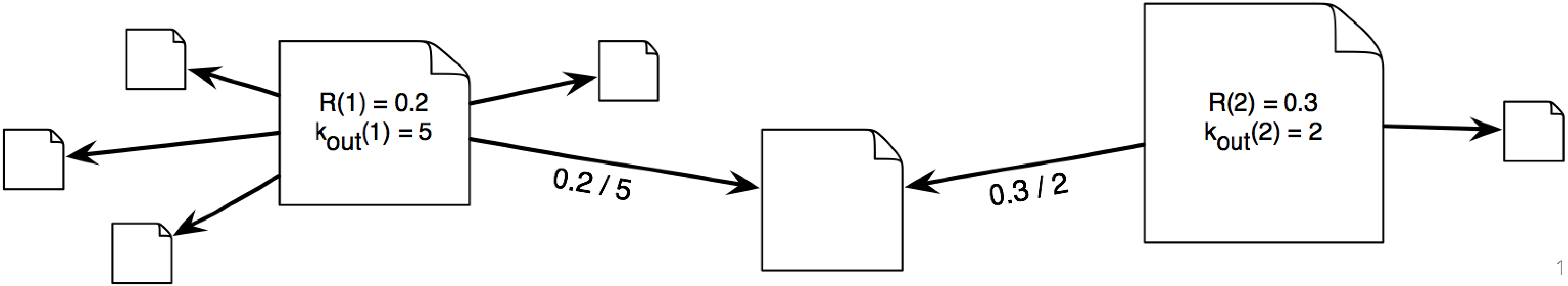
probability to land on *i* by random surfing

PageRank (4/4)

- α is the **teleportation** or **jump factor**, typically 0.15
  - PageRank converges quickly (in few iterations) if α > 0
- 1– α is the **damping factor**

Example with α = 0:

$$R_t(i) = \sum_{j \in pred(i)} \frac{R_{t-1}(j)}{k_{out}(j)}$$



R(3) = 0.2 / 5 + 0.3 / 2 = 0.19

Network Theory and Dynamic Systems

07. Network Models

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Network Models

- **Model**: set of instructions to build networks
- **Goal**: find models that generate networks with the same characteristics as real-world networks

Random Networks (2/2)

- Simple idea: placing links at random between pairs of nodes
- **Algorithm** (Gilbert random network model (parameters: nodes N and link probability p), equivalent to Erdős-Rényi model):

  1. Start with N nodes and zero links

  2. Go over all pairs of nodes; for each pair of nodes i and j, generate a random number r

     between 0 and 1

     1. If $r < p \Rightarrow i$ and j get connected

     2. If $r > p \Rightarrow i$ and j remain disconnected

- Erdős-Rényi model: the number of links of the network is fixed
- Gilbert random model: the number of links of the network is variable

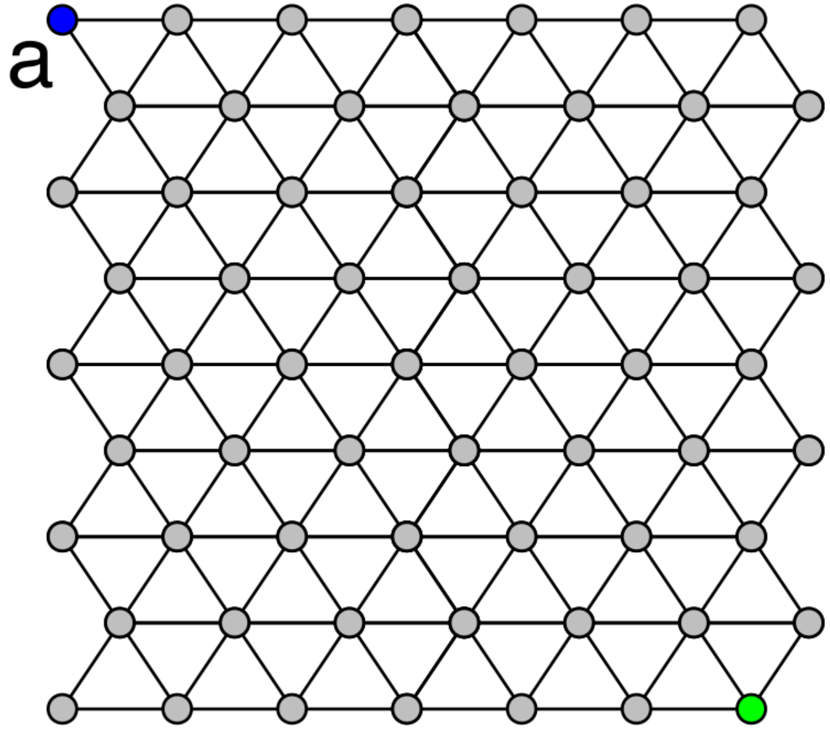
Random Networks: Summary

- Links are placed at random, independently of each other
  - Distances between pairs of nodes are short (small-world property): 😊
- The average clustering coefficient is much lower than on real networks of the same size and average degree: 😟
- The nodes have approximately the same degree, there are no hubs: 😟
- **Conclusion:** the random network is not a good model of many real-world networks

```
G = nx.gnm_random_graph(N,L) # Erdos-Renyi random graph
G = nx.gnp_random_graph(N,p) # Gilbert random graph
```

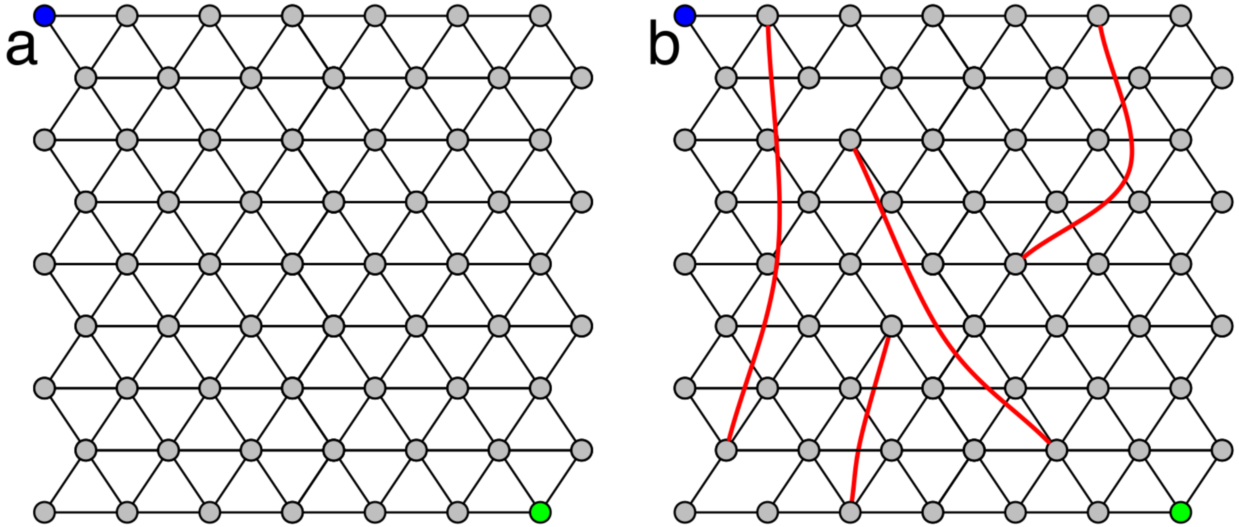
Small-World Networks (1/3)

- **Goal:** building networks with the small-world property and high clustering coefficient
- **Solution:** interpolating between a regular lattice (high clustering) and a random network (small-world property)
- **Clustering coefficient of lattice is high:**
  - The internal nodes have k = 6 neighbors, 6 pairs of which are connected
  - $C = 6/[(6*5)/2] = 6/15 = 2/5 = 0.4$
  - Most nodes are internal, so the average clustering coefficient of the network is close to 0.4!



Small-World Networks (3/3)

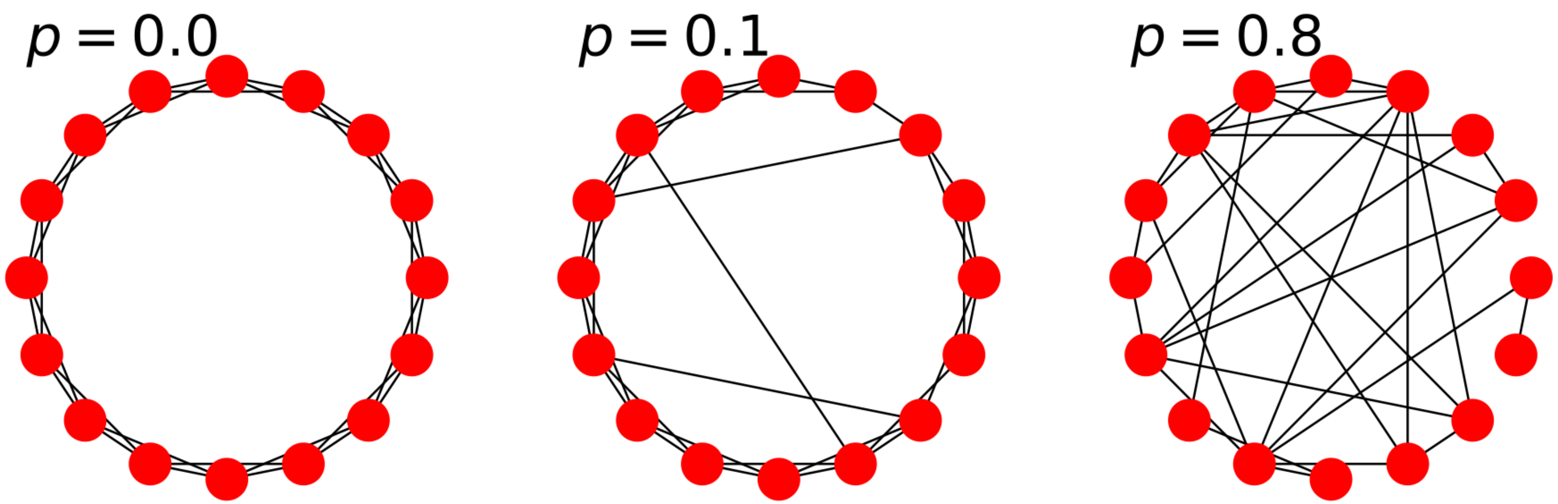
- **Large average shortest path length:** Going from a node to another can take a large number of steps, which grows rapidly with the size of the network/grid



- **Solution?** **Shortcuts!**

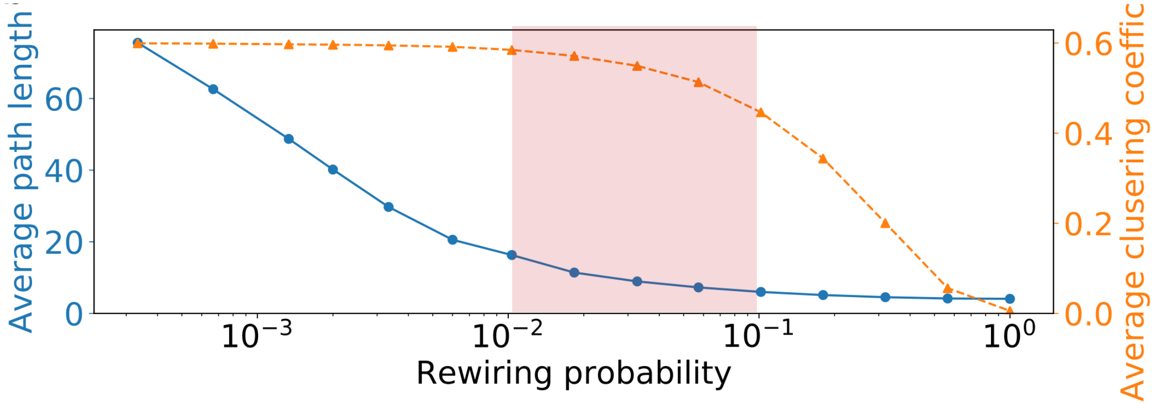
The Watts-Strogatz Model (1/2)

- N nodes form a regular ring lattice, with even degree k. With probability p, each link is rewired randomly



The Watts-Strogatz Model (2/2)

- The expected number of rewired links is $pL = pNk/2$
- If $p = 0$, no links are rewired: **no change**
- If p is small, few links are rewired: **the average clustering coefficient stays approximately the same because very few triangles are destroyed, but distances shrink considerably**
- If $p = 1$, all links are rewired: **the network becomes a random network**



Distances become short already for low values of p; the average clustering coefficient stays high up to large values of p. **There is a range of values of p where the average path length is short and the clustering coefficient is high!**

The Watts-Strogatz Model: Summary

- A regular lattice whose links are randomly rewired, with some probability *p*
- There is a range of values of the rewiring probability *p* for which distances between pairs of nodes are short (small-world property 😊) and the average clustering coefficient is high😊
- The nodes have approximately the same degree, there are no hubs 😟

```
# small-world model network
G = nx.watts_strogatz_graph(N,k,p)
```

Network Theory and Dynamic Systems

08. Network Models

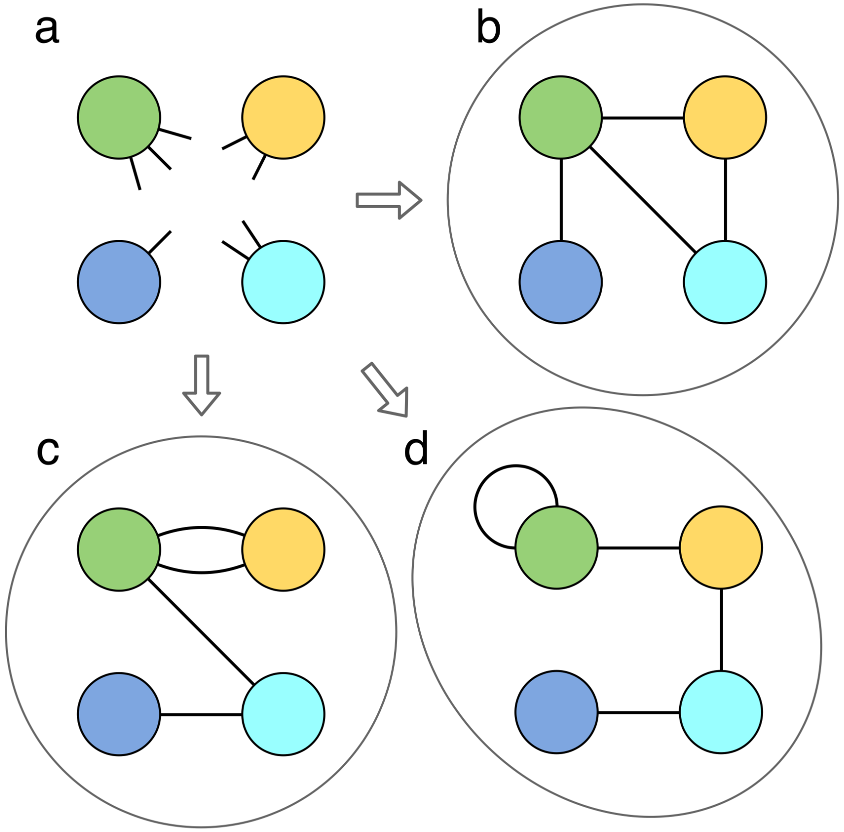
SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

The Configuration Model (1/2)

- **Problem**: is it possible to build networks with a predefined degree distribution?
- **Solution**: the configuration model
- **More specific focus:** build networks with a predefined degree sequence!
- **Degree sequence:** list of N numbers $(k_1, k_2, \dots, k_{N-1}, k_N)$, where k_i is the degree of node i
- **Warning:** many degree sequences can be extracted from the same distribution!
- **Principle:** assign a degree to each node (*e.g.*, from the desired distribution or a real network), place as many stubs on each node as the degree of the node, and attach pairs of stubs at random



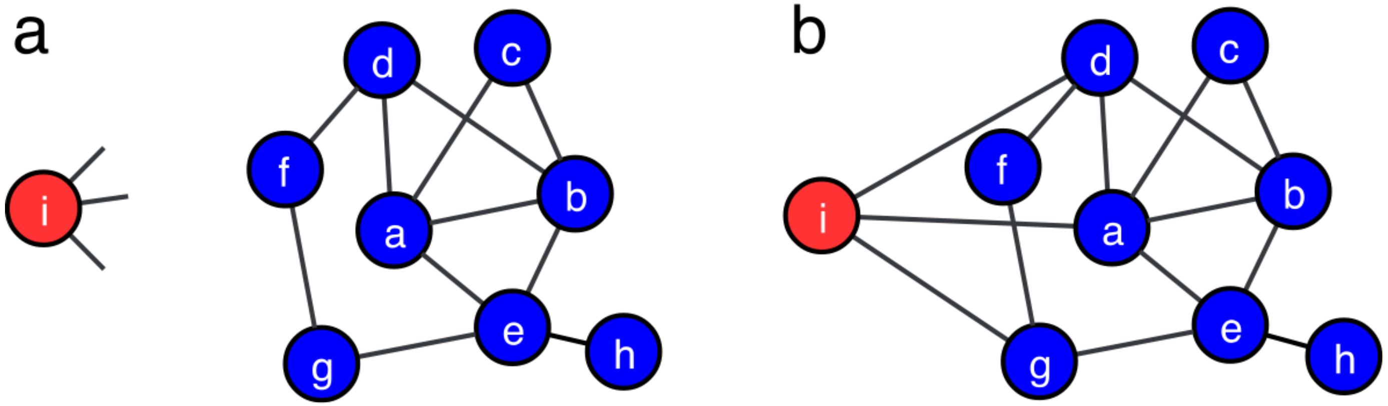
The Configuration Model (2/2)

- **Degree-preserving randomization:** generate randomized versions of a given network with the same degree sequence, using the configuration model
- **Why:** useful to see whether a specific property of the original network is determined by its degree distribution alone
  - If the property is maintained in the randomized configurations, then the degree distribution is the main driver
  - If the property is lost in the randomized configurations, other factors must be responsible for it

```
# network with degree sequence D
G = nx.configuration_model(D)
```

Network Growth (2/2)

- **General procedure**
  1. A new node comes with a given number of stubs, indicating the number of future neighbors of the node (degree)
  2. The stubs are attached to some of the old nodes, according to some rule



Preferential Attachment

- **Note:** Nodes prefer to link to the more connected nodes
- **Examples**
  - Our knowledge of the Web is biased towards popular pages, which are highly linked, so it is more likely that our website points to highly linked Web sites
  - Scientists are more familiar with highly cited papers (which are often the most important ones), so they will tend to cite them more often than poorly cited ones in their own papers
  - The more movies an actor makes, the more popular they get and the higher the chances of being cast in a new movie

The Barabási-Albert Model (1/2)

universität koblenz
weiter:denken

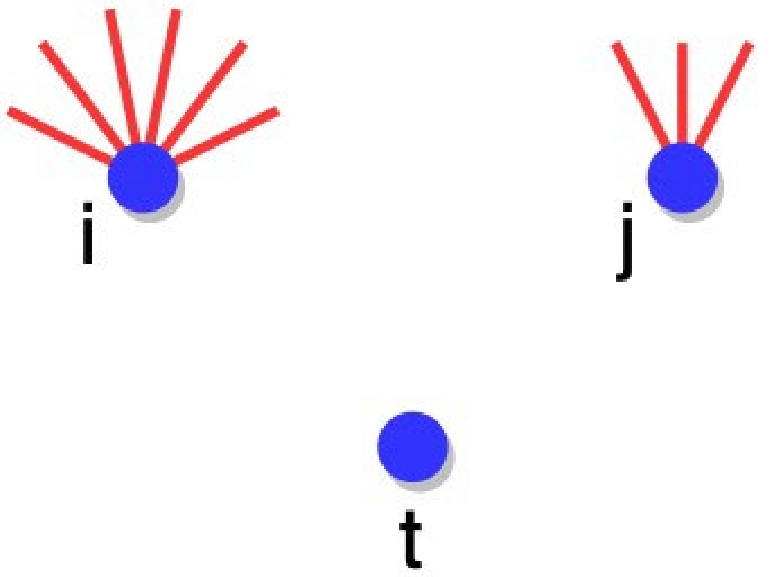
- **Procedure**
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node i is added to the system, and sets m links with some of the older nodes ($m \leq m_0$)
  - The probability that the new node i chooses an older node j as neighbor is **proportional to the degree k_j of j:**

$$\Pi(i \leftrightarrow j) = \frac{k_j}{\sum_l k_l}$$

- The procedure ends when the given number N of nodes is reached

The Barabási-Albert Model (2/2)

- **Example:** if t has to choose between node i, with degree 6, and node j, with degree 3, the probability of choosing i is twice the probability of choosing j



The Barabási-Albert Model (2/3)

universität koblenz
weiter:denken

- Hubs are the **oldest** nodes: they get the initial links and acquire an advantage over the other nodes, which increases via preferential attachment
- **Question:** if old nodes have an advantage over newer nodes anyway, do we need preferential attachment at all? Can we explain the existence of hubs just because of growth?
- **Alternative model:** each new node chooses its neighbors at random, not with probability proportional to their degree

Non-linear Preferential Attachment (1/4)

- Procedure
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node *i* is added to the system, and sets *m* links with some of the older nodes ($m \leq m_0$)
  - The probability that the new node *i* chooses an older node *j* as neighbor is **proportional to the power $\boldsymbol{\alpha}$ of the degree $\boldsymbol{k_j}$ of *j*:**

$$\Pi_\alpha(i \leftrightarrow j) = \frac{k_j^\alpha}{\sum_l k_l^\alpha}$$

- The procedure ends when the given number *N* of nodes is reached

Non-linear Preferential Attachment (3/4)

universität koblenz
weiter:denken

$$\Pi_\alpha(i \leftrightarrow j) = \frac{k_j^\alpha}{\sum_l k_l^\alpha}$$

- For $\boldsymbol{\alpha} = 1$ we recover the **linear preferential attachment (BA model)**
- **Question**: what happens when $\boldsymbol{\alpha} \neq 1$?
- **Answer**: it depends on whether $\boldsymbol{\alpha} > 1$ or $\boldsymbol{\alpha} < 1$

Extensions of the BA Model: Attractiveness Model (1/3)

- **Pitfall of preferential attachment:** What happens if a node has no neighbors (degree zero)? It will never get connections from other nodes!
- **No problem for standard initial condition:** the initial subgraph is complete (clique), so every node has nonzero degree
- **What if the network is directed and the linking probability is proportional to the in-degree? Bad,** as each new node has in-degree zero, so it will never be linked by future nodes!

Extensions of the BA Model: Attractiveness Model (2/3)

- **Procedure**
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node i is added to the system, and sets m links with some of the older nodes ($m \leq m_0$)
  - The probability that the new node i chooses an older node j as neighbor is proportional to the sum of the degree k_j of j and an attractiveness A:

$$\Pi(i \leftrightarrow j) = \frac{A + k_j}{\sum_l (A + k_l)}$$

Extensions of the BA Model: Fitness Model (1/4)

- **Pitfall of preferential attachment:** the hubs are the oldest nodes. **Unrealistic!**
- **Examples:**
  - In the Web, new pages can overrun old pages (*e.g.,* Google!)
  - In science, new papers can be more successful than (many) old papers
- **Reason:** each node has its own individual appeal!

Extensions of the BA Model: Fitness Model (2/4)

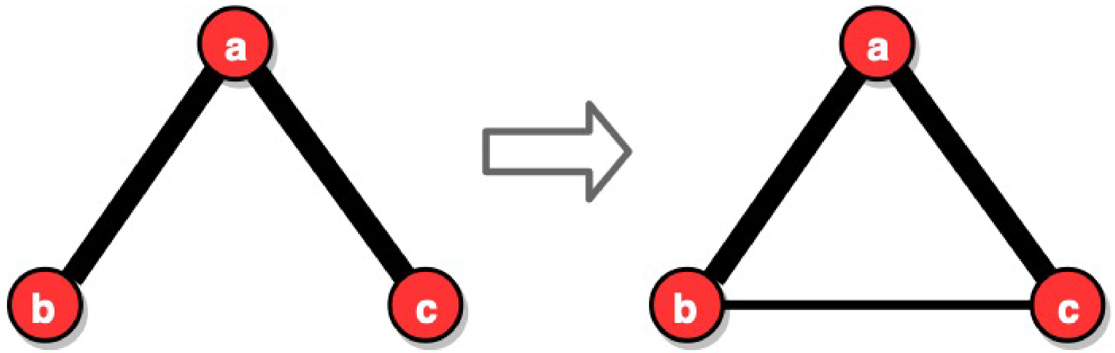
universität koblenz
weiter:denken

- **Procedure**
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node *i* is added to the system, and sets *m* links with some of the older nodes ($m \leq m_0$)
  - The probability that the new node *i* chooses an older node *j* as neighbor is **proportional to the product of the degree k_j of *j* with a fitness η_j**, indicating the intrinsic appeal of *j*:

$$\Pi(i \leftrightarrow j) = \frac{\eta_j k_j}{\sum_l \eta_l k_l}$$

Extensions of the BA Model: Random Walk Model (1/6)

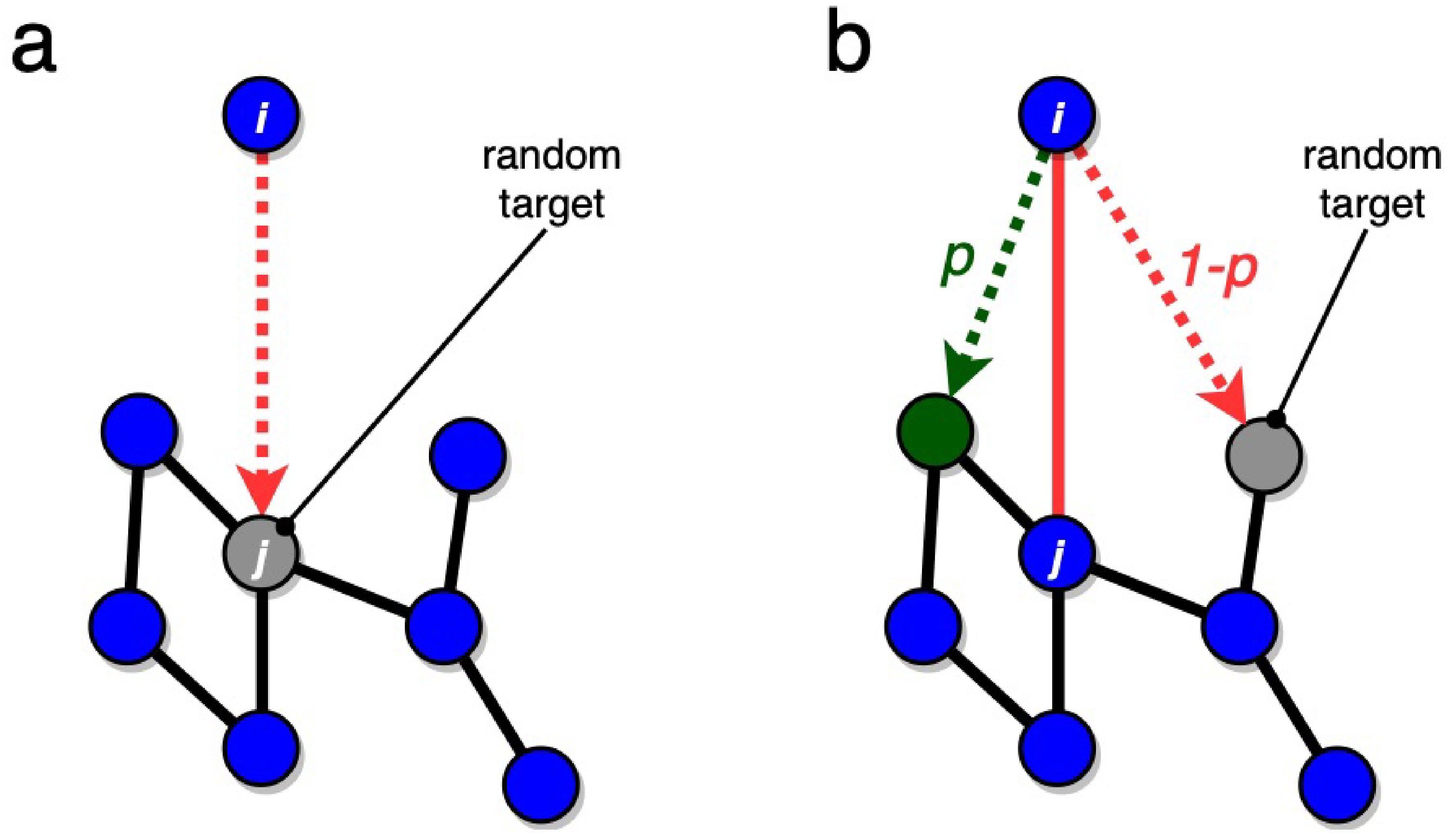
- **Pitfall of preferential attachment:** the BA model does not generate many triangles. **Why?**
- To close a triangle we need to set a link between two neighboring nodes, whereas in the BA model links are set based on degree, regardless of whether the future neighbors have common neighbors
- **Solution:** introduce a mechanism for triadic closure in the model!



Extensions of the BA Model: Random Walk Model (2/6)

- Procedure
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node i is added to the system, and sets m links with some of the older nodes ($1 < m \leq m_0$)
  - The first link targets a randomly chosen node j
  - From the second link onwards:
    - With probability p the link is set with a neighbor of j, chosen at random
    - With probability $1-p$ the link is set with a randomly chosen node

Extensions of the BA Model: Random Walk Model (3/6)



Extensions of the BA Model: Random Walk Model (4/6)

- Results
  - The degree distribution is heavy-tailed
  - The average clustering coefficient is much higher than in BA networks (the larger, the greater the probability p of triadic closure)
  - When the triadic closure probability p is sufficiently high that many triangles are formed ($p \sim 1$) the network has **community structure**, *i.e.*, it is made of cohesive groups of nodes, loosely connected to each other

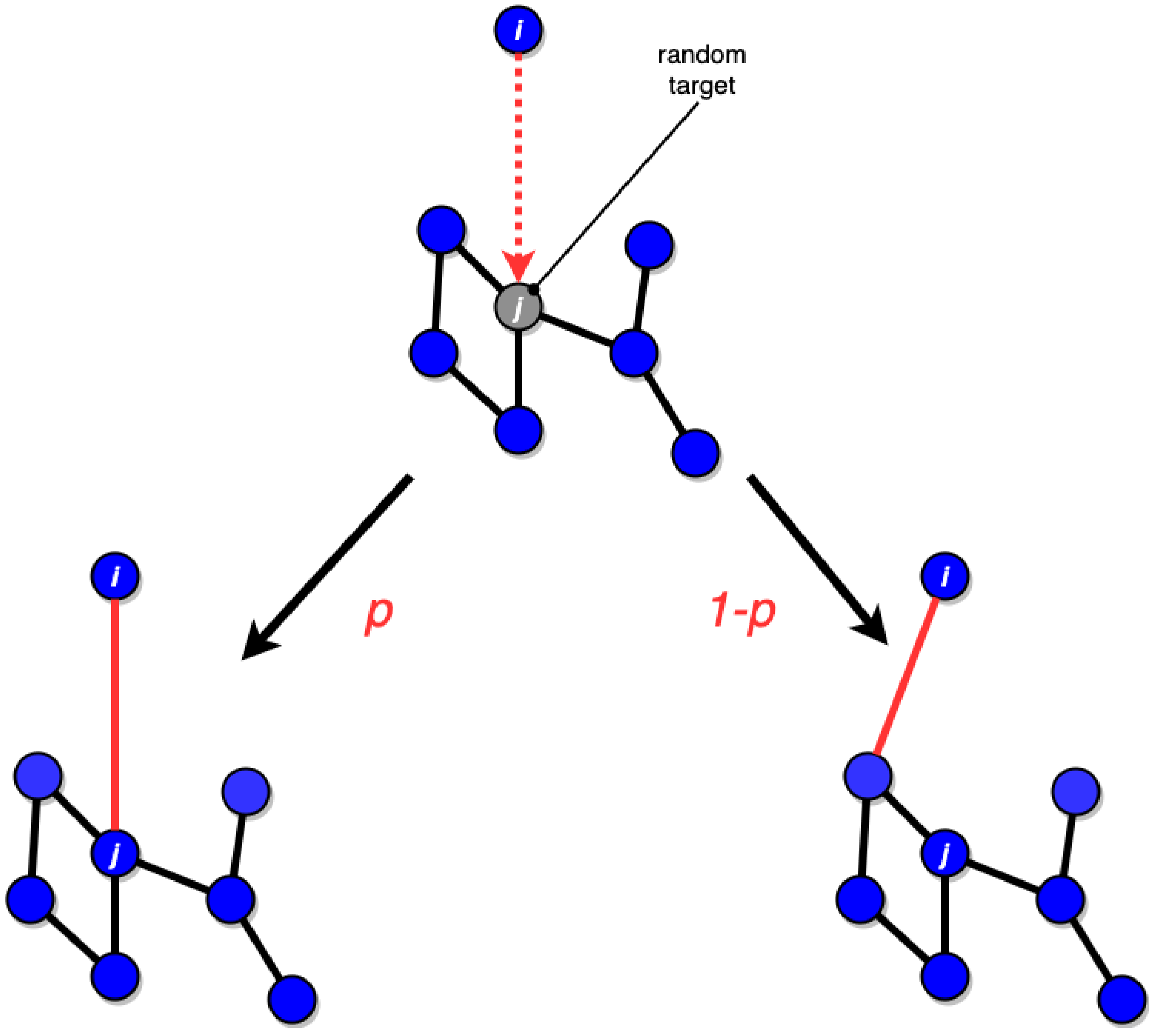
Extensions of the BA Model: Copy Model (1/3)

- **Motivation:** the authors of a new Webpage tend to “copy” the hyperlinks of other pages on the same topic
- **Steps**
  - **Growth:** at each time step a new node i is added to the network
  - **Target selection:** a node j is selected at random
  - **Random connection:** with probability p the new node is connected to j
  - **Link copying:** with probability $1-p$ the new node is connected to a neighbor of j, chosen at random

Extensions of the BA Model: Copy Model (2/3)

universität koblenz
weiter:denken

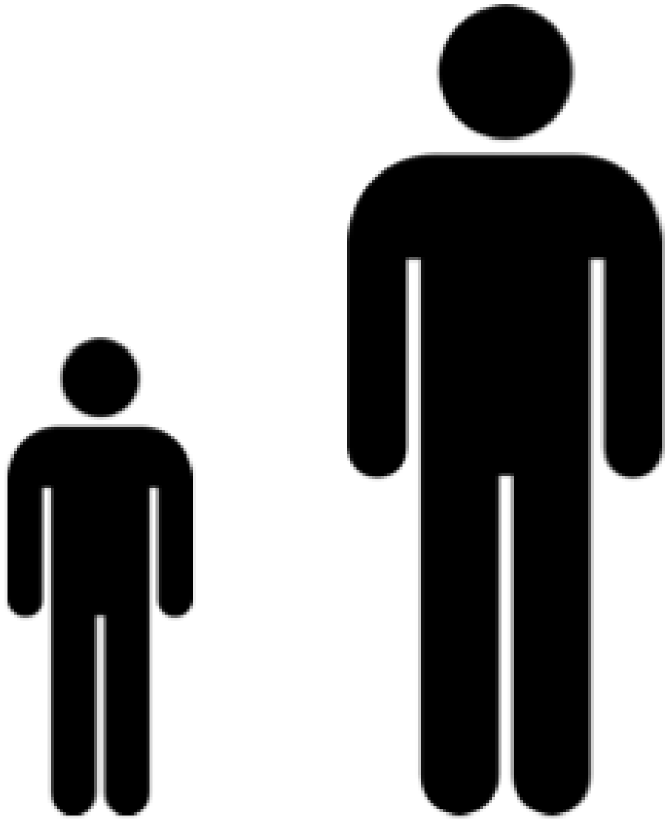
- Difference from random walk model: here we do not link to the target and to its neighbor (no triadic closure)!



Extensions of the BA Model: Rank Model (1/3)

universität koblenz
weiter:denken

- **Pitfall of preferential attachment:** BA model implies that nodes have a perception of how important other nodes are, *i.e.*, how large is their degree
- **Objection:** in the real world there is no such perception of the absolute value of things**, it is far easier to perceive the relative value!**
- **Solution:** ranking!



Extensions of the BA Model: Rank Model (2/3)

universität koblenz
weiter:denken

- **Procedure**
  - Nodes are ranked based on a property of interest (*e.g.*, age, degree). The rank of node *i* is R_i
  - Start with a group of m_0 nodes, usually fully connected (clique)
  - At each step a new node *i* is added to the system, and sets *m* links with some of the older nodes ($m \leq m_0$)
  - The probability that the new node *i* chooses an older node *j* as neighbor is **proportional to a power of the rank of *j*:**

$$\Pi(i \leftrightarrow j) = \frac{R_j^{-\alpha}}{\sum_l R_l^{-\alpha}}$$

Extensions of the BA Model: Rank Model (3/3)

universität koblenz
weiter:denken

$$\Pi(i \leftrightarrow j) = \frac{R_j^{-\alpha}}{\sum_l R_l^{-\alpha}}$$

- **Remark:** highly-ranked nodes (those with low values of *R*) have high probabilities of being linked, much higher than poorly-ranked nodes
- **Result:** the model generates networks with hubs, for any value of the exponent ***α*** and any property used to rank the nodes!

universität koblenz
weiter:denken

Network Theory and Dynamic Systems

09. Dynamics I

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

WeST
Web Science and Technologies

Dynamics on Networks (1/2)

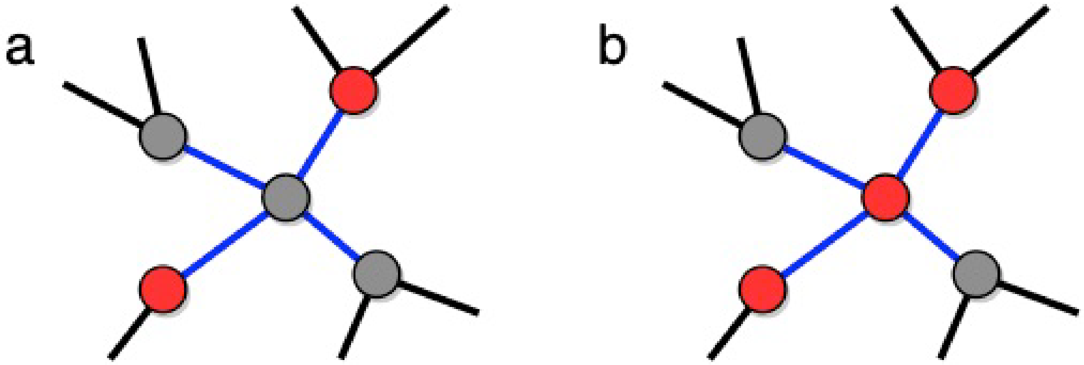
- **Misinformation diffusion** is a key example of how dynamic processes unfold on networks
  - Nodes (e.g., users or devices) interact with neighbors and **can adopt, modify, or pass on information**
- **What Changes Over Time?**
  - Node features: beliefs, knowledge, infection status, query targets, etc.
  - Influences often follow the structure of the network — **local neighbors impact a node's state**

Dynamics on Networks (2/2)

- **Common Examples:**
  - **Information Diffusion**: Like fake news or viral content spreading across social media
  - **Epidemic Spreading**: Diseases (e.g., COVID-19) propagating via human contact networks
  - **Opinion Dynamics**: How people's beliefs or preferences evolve (e.g., polarization, consensus)
  - **Network Search**: How information or targets are located through decentralized queries (e.g., peer-to-peer networks)

Information Diffusion (3/3)

- **Outcome**: The process results in **influence cascades** — a chain reaction where nodes activate one after another, triggered by their neighbors
- **What Do Cascades Look Like?**
  - Small cascades: Only a **few nodes** are influenced before the spread dies out
  - Global cascades: A **large part** of the network becomes activated — possibly reaching everyone
- **Example**
  - **a:** Initial state with red (active) influencers and inactive gray nodes
  - **b:** After applying the activation rule, more nodes turn red → showing how activation spreads



Threshold Models (1/4)

- Principle: A node becomes active **only if the total influence** from its active neighbors **exceeds a predefined threshold**
  - This models resistance or hesitation to adopt new behaviors, ideas, or technologies
- Linear threshold model: the influence on a node *i* is defined as a sum over its active neighbors, in which the contribution of each neighbor is given by the weight of the link joining it to the node

$$I(i) = \sum_{j:\,active} w_{ji}$$

  - w_{ji} = weight of the link from *j* to *i*

Threshold Models (2/4)

- Activation condition $I(i) \geq \theta_i$

  where θ_i is the threshold of node *i*, indicating its tendency to be influenced
- On unweighted networks $I(i) = n_i^{on} \geq \theta_i$

  where n_i^{on} is the number of active neighbors of node *i*
- If all nodes have the same threshold θ

$$I(i) = n_i^{on} \geq \theta$$

Threshold Models (3/4)

- **Model dynamics**
  - Start with some initially activated nodes — chosen randomly or based on a proportion.
  - Activation is permanent: once a node becomes active, it stays active.
  - Each inactive node checks whether the influence from active neighbors meets or exceeds its threshold

Fractional Threshold Model (1/7)

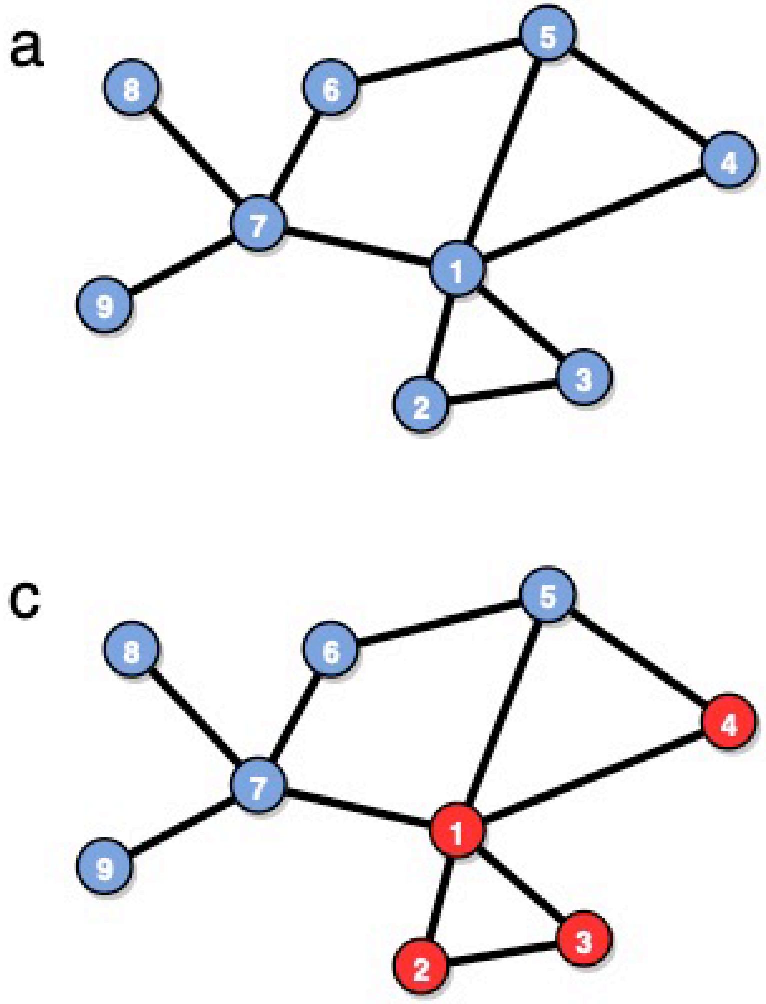
- **Principle:** Instead of counting how many neighbors are active, we focus on the fraction of active neighbors
- A node activates if this fraction exceeds its threshold θ_i
- **Example**
  - if θ=1/2, then at least 50% of the neighbors must be active for node i to activate
- **Activation condition**

$$\frac{n_i^{on}}{k_i} \geq \theta_i$$

- n_i^{on}: number of active neighbors of node *i*
- k_i: total number of neighbors (degree of node *i*)
- θ_i: activation threshold (fractional)

Fractional Threshold Model (2/7)

universität koblenz
weiter:denken



$\theta = 1/2$

Fractional Threshold Model (4/7)

universität koblenz
weiter:denken

- In **sparse networks**, whether a cascade spreads or not depends heavily on the **network structure**—not just thresholds
- Key driver: **Vulnerable Nodes**
  - These are nodes that can be activated by just one active neighbor
  - They are crucial to the onset and continuation of cascades
- **Condition for a node to be vulnerable**

$$k_i \leq \frac{1}{\theta_i}$$

- **Global Cascades**:
  - To trigger large-scale (global) cascades, the network must contain **enough vulnerable nodes**
  - Their presence creates "weak spots" where activation can propagate quickly

Independent Cascade Models (1/4)

- Principle of threshold models
  - Based on **peer pressure**: the **more neighbors** try to influence you, the **higher the chance** you'll adopt their behavior

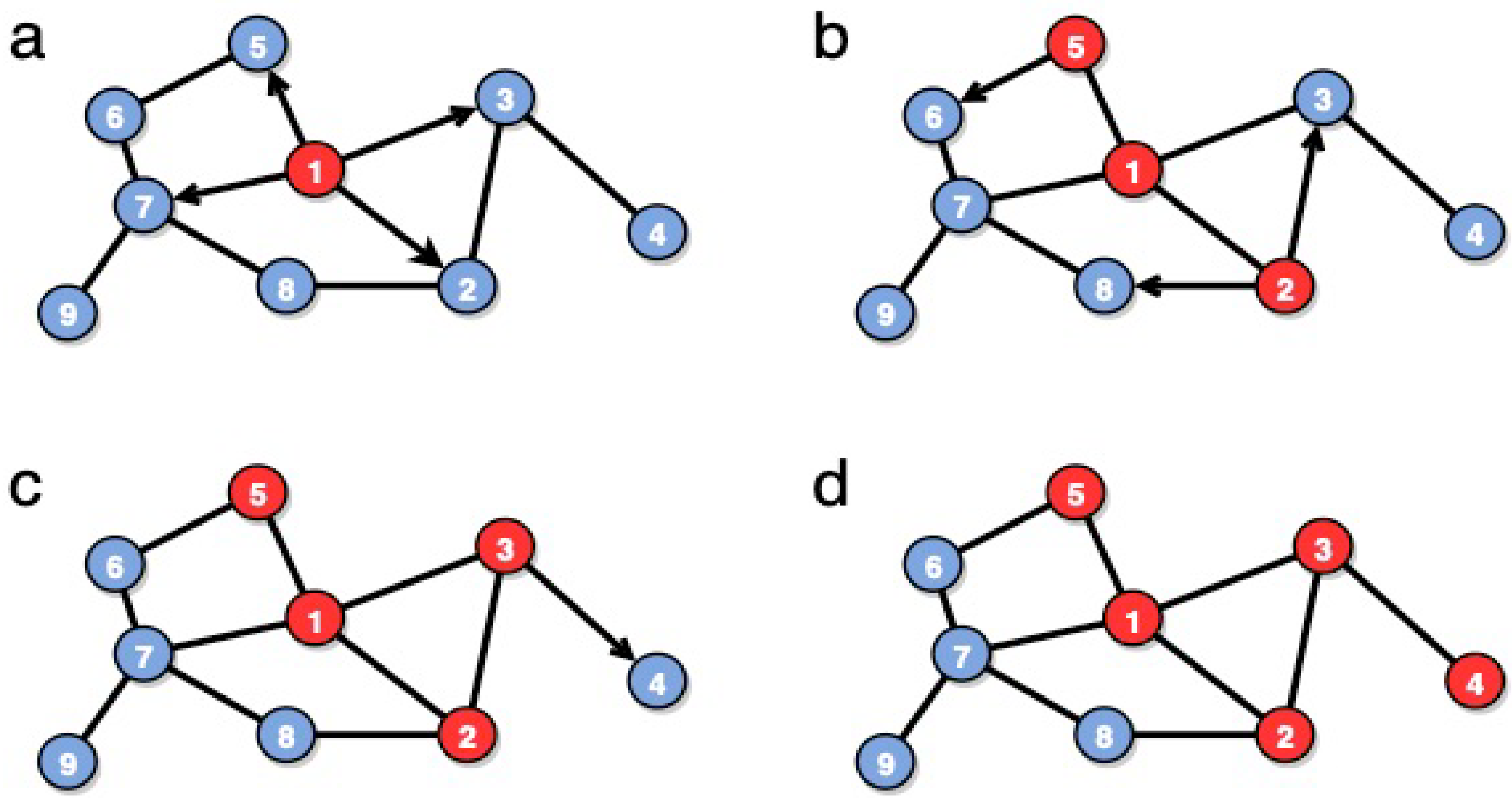
But

  - In reality, we may be persuaded by just **one** passionate friend
  - Influence happens in individual interactions, not just group pressure
- **Alternative principle**
  - Each contact has a separate, independent chance to influence you.
  - Influence spreads via pairwise (node-to-node) interactions
- **Independent cascade models** are based on node-node interactions!

Independent Cascade Models (2/4)

- **Model dynamics**
  - An active node i has a probability p_{ij} to convince its inactive neighbor j ($p_{ij} \neq p_{ji}$, in general)
  - All active nodes are considered in sequence
    - The inactive neighbor j of the active node i is activated with probability p_{ij}
    - All inactive neighbors of i have one chance to be persuaded by i
  - If a node j is activated, it has only one chance to activate its inactive neighbors

Independent Cascade Models (3/4)



Epidemic Models

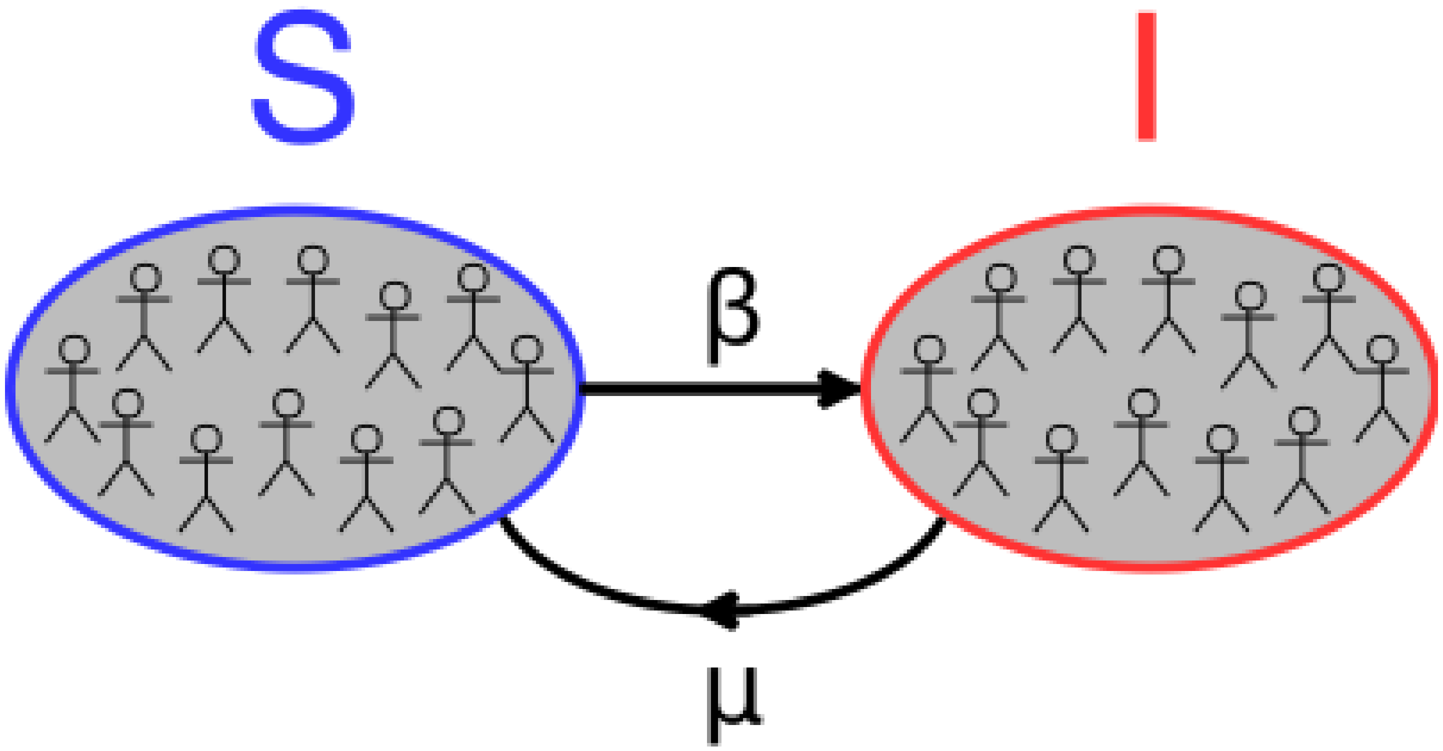
- Classic epidemic models divide the population into **compartments**, corresponding to different stages of the disease
- Key compartments
  - Susceptible (S): individuals who can contract the disease
  - Infected (I): individuals who have contracted the disease and can transmit it to susceptible individuals
  - Recovered (R): individuals who recovered from the disease and cannot be infected anymore

The SIS Model (1/3)

- Just **two compartments**: Susceptible (S) and Infected (I)
- **Dynamics**
  - A susceptible individual gets infected with a probability β (**infection rate**)
  - An infected individual recovers and becomes susceptible again with a probability μ (**recovery rate**)
  - The model applies to diseases that do not confer long-lasting immunity (e.g., common cold)

The SIS Model (2/3)

universität koblenz
weiter:denken

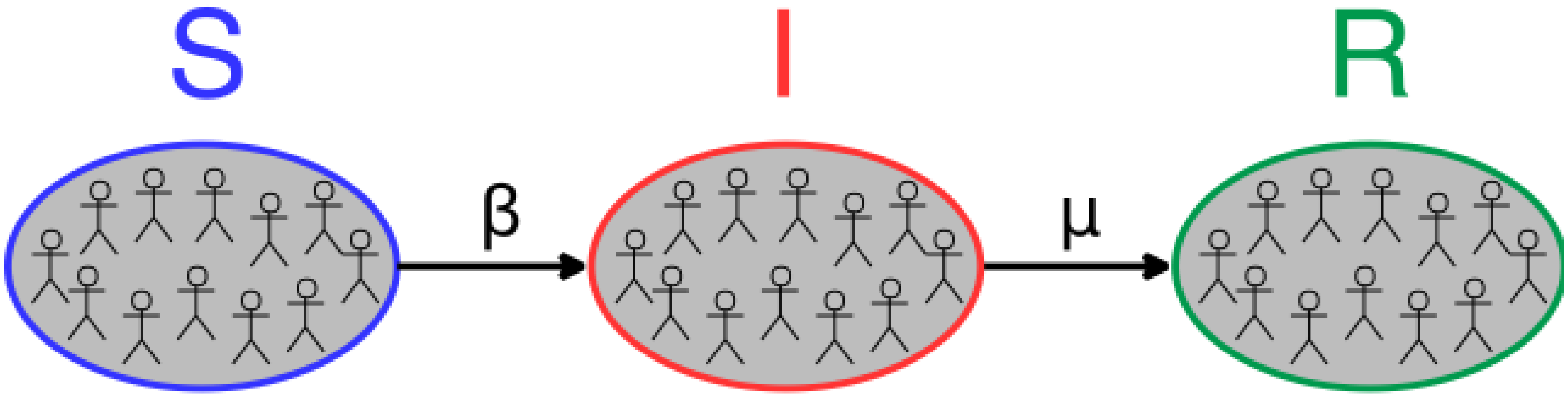


The SIS Model (3/3)

- **Simulation of SIS dynamics on networks**
  - Take a network (e.g., a random network or a real contact network)
  - A number (fraction) of the nodes are infected (e.g., at random), all others are susceptible
  - All nodes are visited in sequence
  - For each node i:
    - If i is susceptible, loop over its neighbors: for each infected neighbor, i becomes infected with probability β
    - If i is infected, it becomes susceptible with probability μ

The SIR Model

- **Difference from SIS model:** when infected individuals recover, they do not become susceptible again, but they are moved to the compartment R and play no further role in the dynamics
- The model applies to diseases that confer long-lasting immunity (e.g., measles, mumps, rubella, etc.)



SIS & SIR Models on Networks (1/3)

- **Start:** homogeneous contact network, with all nodes having degree approximately equal to $<k>$
- **Early stage:** few people are infected, so we can assume that every infected individual is in contact with mostly susceptible individuals
- Each infected individual can transmit the disease to about $<k>$ people at each iteration —> the expected number of people infected by a single person after one iteration is $\beta<k>$
- If there are I infected individuals, we expect to have $I_{sec} = \beta<k>I$ new infected people after one iteration and $I_{rec} = \mu I$ recovered people

SIS & SIR Models on Networks (2/3)

- Threshold condition for epidemic spreading: $I_{sec} > I_{rec}$

$$\beta\langle k\rangle I > \mu I \implies R_0 = \frac{\beta}{\mu}\langle k\rangle > 1$$

- $R_0 = \beta<k>/\mu$ is the **basic reproduction number**
- If $R_0 < 1$, the **initial outbreak dies out in a short time**, affecting only a few individuals
- If $R_0 > 1$, the **epidemic keeps spreading**

Rumor Spreading (1/5)

universität koblenz
weiter:denken

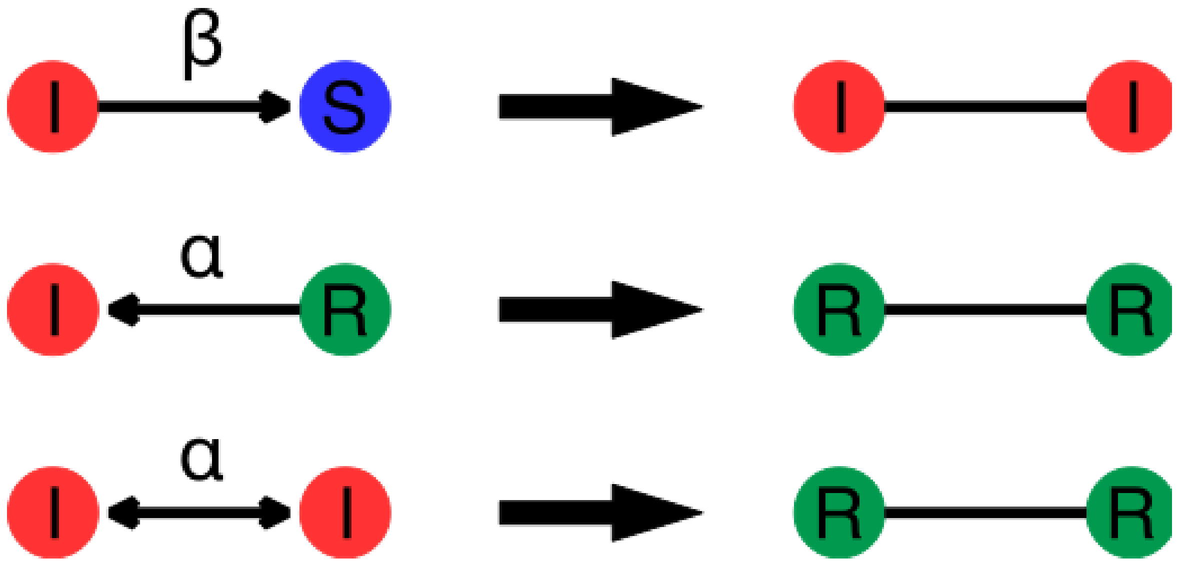
- Rumor spreading can be described as an epidemic spreading process
- **Simple model**: variant of SIR
- **Three compartments:** ignorant (S), spreaders (I) and stiflers (R). Stiflers are people who know the rumor but do not spread it
- **Basic idea:** people are engaged in the diffusion of the rumor as long as they find people who are unaware of it, otherwise they lose interest and stop spreading the rumor

Rumor Spreading (2/5)

- Model
  - When a spreader approaches an ignorant, the rumor is told and the ignorant becomes a spreader with a **transmission probability**
  - When a spreader meets a stifler, the spreader becomes a stifler with a **stop probability**
  - When two spreaders meet, they both turn to stiflers with the same stop probability

Rumor Spreading (3/5)

- Two parameters
  - **Transmission probability β**
  - **Stop probability α**
- Setup: network, real or computer-generated, all nodes are in state S (unaware of the rumor), except a few of them, which are in state I (aware of the rumor and willing to spread it)



ignorant (S), spreaders (I), stiflers (R)

Rumor Spreading (4/5)

- Dynamics
  - At each iteration all nodes are visited *synchronously* or *asynchronously* in random order. For each *i*:
    - If *i* is ignorant, loop over its neighbors: for each spreader neighbor, *i* becomes a spreader with probability β
    - If *i* is a spreader, loop over its neighbors
      - For each stifler neighbor, i becomes a stifler with probability α
      - For each spreader neighbor, i and the neighbor both become stiflers with probability α

Network Theory and Dynamic Systems

10. Dynamics II

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Opinion Dynamics (1/2)

- **Opinion dynamics:** The scientific study of how opinions form, evolve, and spread within society (social networks)
- Opinions change via interactions on social networks (friends, followers)
- Opinions can be:
  - **Discrete**: Categorical choices (e.g., "yes/no," specific candidate)
  - **Continuous**: A spectrum of beliefs (e.g., 0–1 rating, degree of agreement)

Opinion Dynamics (2/2)

Two Main Model Types

- **Discrete Opinion Dynamics**
  - Opinions are distinct, integer values
  - Example: A group decides where to eat; individuals adopt the most popular restaurant choice among their friends
- **Continuous Opinion Dynamics**
  - Opinions are real numbers (degrees of belief)
  - Example: People only discuss politics with those holding "similar enough" views, potentially leading to polarization or consensus
- **Why it is important:** Helps us understand social phenomena like *misinformation*, *political polarization*, and *collective decision-making*

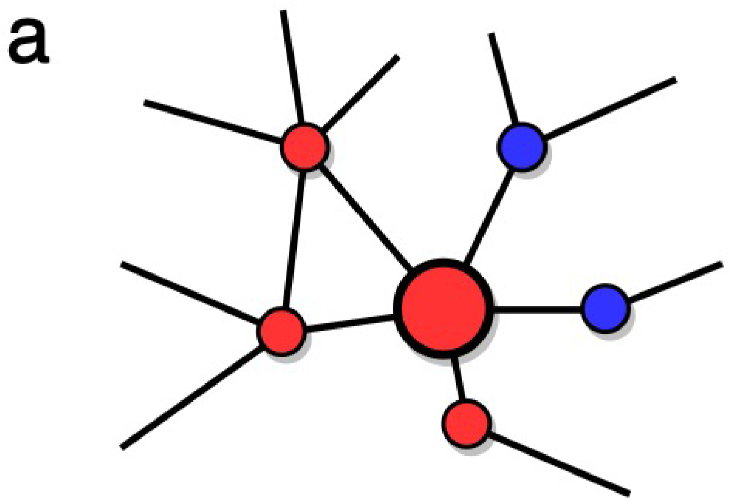
Discrete Opinion Dynamics Models

universität koblenz
weiter:denken

- Two models
  - The majority model
  - The voter model

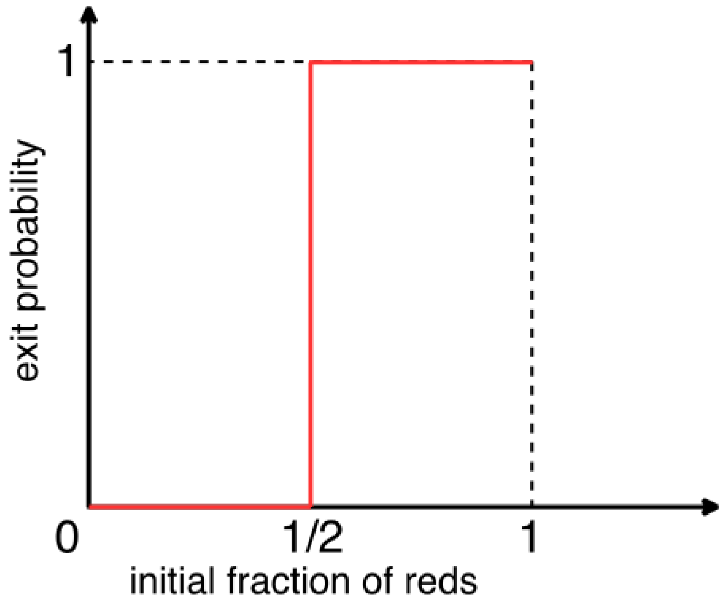
Majority Model (1/2)

- **Majority rule:** each node adopts the opinion of the majority of its neighbors
- If the number of neighbors is even (2, 4, etc.) and there is an equal number of them with either opinion, then we flip a coin to decide which opinion will be taken by the node
- Equivalent to fractional threshold model of information diffusion, with threshold 1/2



Majority Model (2/2)

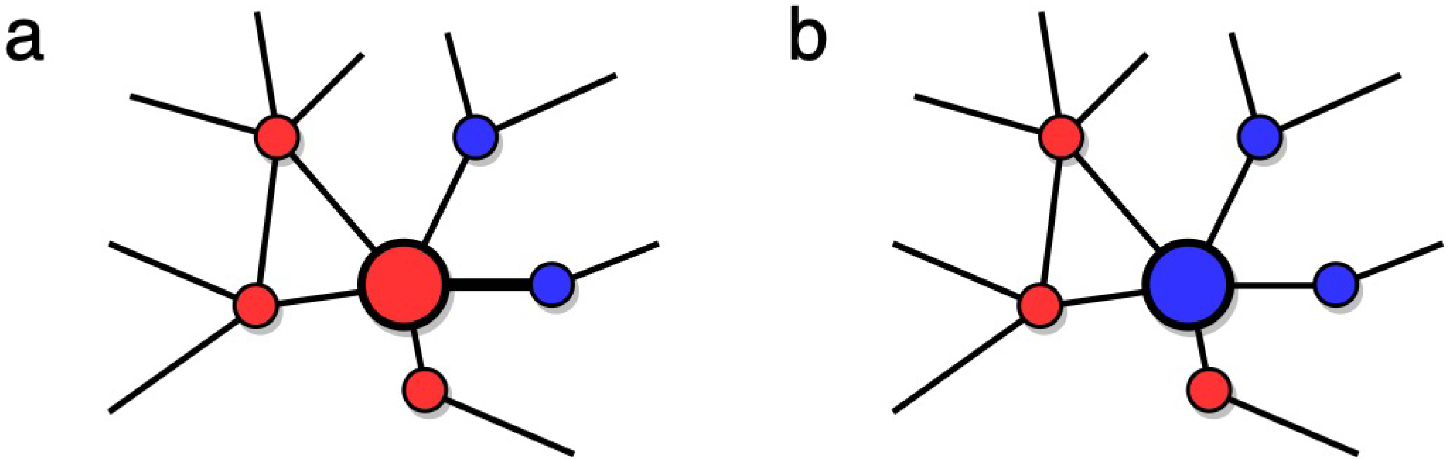
- **Stable States (Equilibria)**
  - **Consensus:** All nodes eventually adopt the same single opinion
  - **Coexistence** (Polarization): Different opinions persist stably in the network, with each node holding the majority opinion within its local neighborhood
- **Consensus in Majority Model**
  - On most complex networks (e.g., random networks, scale-free networks), the Majority Model typically does not reach full consensus, often resulting in a coexistence of opinions
  - However, consensus can be achieved on highly structured topologies such as **one-** and **two-dimensional grids**
- **Exit Probability Profile**
  - For the runs that do lead to consensus, the exit probability (likelihood of reaching consensus on a specific opinion) typically exhibits a step-like profile as a function of the initial fraction of that opinion (see figure)



Voter Model (1/2)

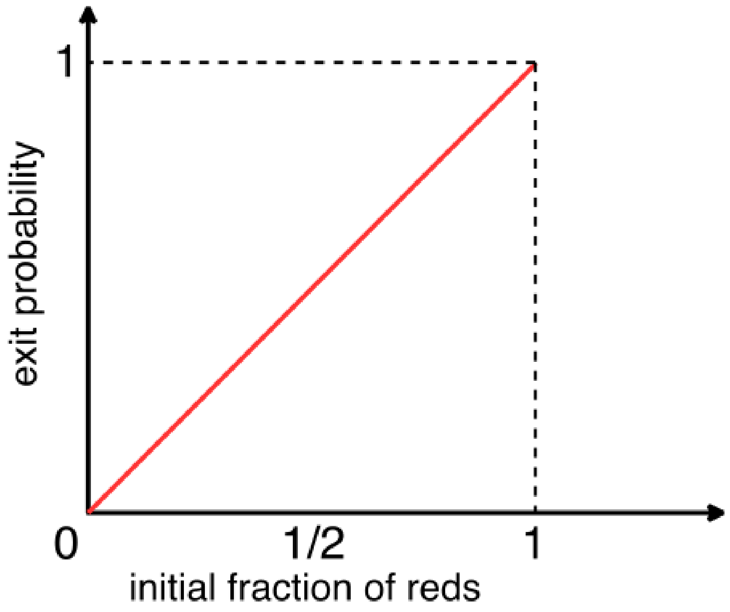
weiter:denken

- **Update Rule:** At each step, a randomly chosen node adopts the opinion of one of its randomly chosen neighbors
- **Consensus** (a state where all nodes share the same opinion) is the sole stable state of the Voter Model and is reached regardless of the underlying network topology
- **Convergence:** In any state where diverse opinions coexist, interactions between neighbors holding different opinions will always occur, facilitating the propagation of opinions until a single opinion dominates



Voter Model (2/2)

- **Exit Probability:** For the Voter Model, the probability of reaching consensus on a specific opinion (e.g., opinion '1') is precisely equal to the initial fraction of nodes holding that opinion
  - This is often referred to as a **diagonal function**
- **Consensus:** Even when starting from an initial configuration with a majority of one opinion (e.g., opinion '1'), the system can still converge to consensus on the minority opinion (e.g., opinion '0') due to the stochastic nature of the update process
- **Example:** If 30% of the nodes initially hold opinion '1', then we expect that in 30% of independent simulation runs, the entire network will ultimately reach consensus on opinion '1'



Continuous Opinions (1/2)

- Dynamics where opinions are represented by **real** numbers, allowing them to vary smoothly along a continuous spectrum of possible values
  - **Example**: Political alignment on a scale from very progressive (e.g., −1) to very conservative (e.g., +1)
- **Initial Configuration:** Opinions are typically assigned *randomly* from a uniform distribution within the predefined range of possible values
- **Stopping Criterion:** Simulations are usually terminated when the maximum change in any individual opinion between successive iterations falls below a small, predefined threshold
  - This signifies that the system has reached a stationary state (equilibrium)

Continuous Opinions (2/2)

- **Possible Stationary States**: Depending on the model and parameters, the system can converge to
  - **Consensus**: All opinions converge to a single, shared value
  - **Polarization**: Opinions cluster around two distinct, opposing values
  - **Fragmentation**: Opinions disperse and cluster around multiple distinct values, forming several factions

Bounded Confidence Model (1/4)

universität koblenz
weiter:denken

- **Core Principle**
  - Interactions between two individuals (nodes) result in opinion influence only if the absolute difference between their current opinions is less than or equal to a predefined value
    - This value is known as the confidence bound, or tolerance (often denoted as ϵ)
    - This implies that individuals only engage with, or are influenced by, those whose opinions are "close enough" to their own
- **Model Parameters**
  - The confidence bound ϵ (epsilon): Defines the maximum opinion difference for interaction
  - The convergence parameter μ (mu): Determines the extent to which opinions are adjusted during an interaction (i.e., how much closer they move)

Bounded Confidence Model (2/4)

- Dynamics
  - At iteration *t*, each node *i* has opinion $o_i(t)$, which is a real number between, say, zero and one
  - An iteration consists of a sweep over all nodes, synchronously or in random order
  - At iteration $t+1$, for each node *i* we pick one neighbor *j* at random. If

$$|o_i(t) - o_j(t)| < \epsilon$$

  - the opinions of *i* and *j* are changed as

$$o_i(t+1) = o_i(t) + \mu[o_j(t) - o_i(t)]$$
$$o_j(t+1) = o_j(t) + \mu[o_i(t) - o_j(t)]$$

Bounded Confidence Model (3/4)

$$o_i(t+1) = o_i(t) + \mu[o_j(t) - o_i(t)]$$

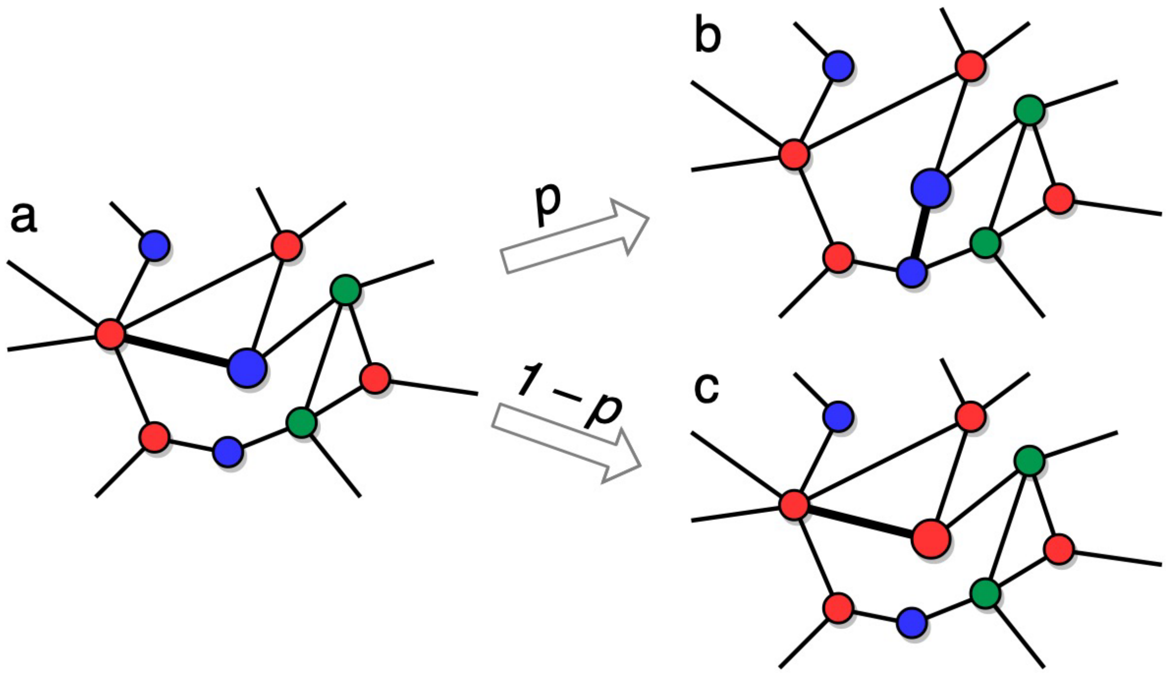
$$o_j(t+1) = o_j(t) + \mu[o_i(t) - o_j(t)]$$

- If μ = 1/2, the opinions converge to their average
- If μ = 1, they switch! The parameter μ usually varies between 0 and 1/2
- If we sum the opinion update equations side by side and divide by two, we see that the average opinion of *i* and *j* is the same before and after the update —> **the average opinion of the population is preserved by the dynamics**
- **Consequence**
  - If the initial opinions are taken at random from the range [0, 1], their average is 1/2 (with possible small deviations)
  - So, **if the system eventually reaches consensus, the opinions of all nodes will cluster around 1/2**

Coevolution of Networks and Dynamics (3/7)

universität koblenz
weiter:denken

- **Dynamics**
  - Each iteration is a sweep over the nodes, synchronously or in random order
  - For each node i select a random neighbor j with different opinion from i:
  - With the probability p, the link between i and j is rewired from i to a randomly selected non-neighbor holding the same opinion as i **(selection)**
  - With probability $1 - p$, i takes the opinion of j **(influence)**



Coevolution of Networks and Dynamics (4/7)

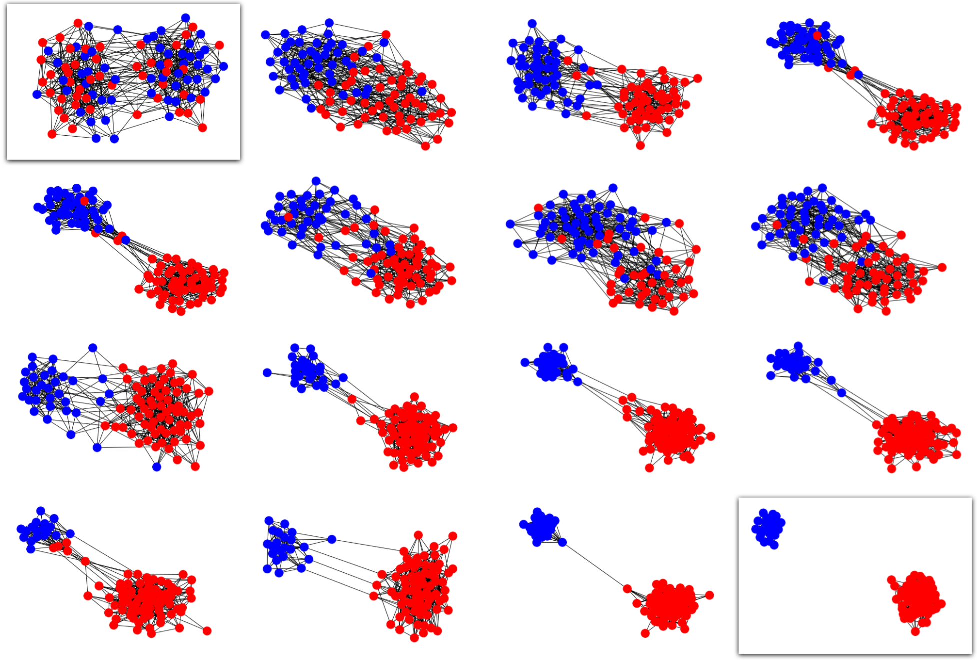
universität koblenz
weiter:denken

- Both **selection** (nodes forming ties based on similarity) and **social influence** (nodes becoming similar due to existing ties) work synergistically
  - They collectively tend to reduce the number of neighboring node pairs holding different opinions
  - This drives the network towards a state where all connected neighbors share the same opinion
- Network Fragmentation (Polarization): As a consequence of homophilous alignment, the network eventually divides into a set of separate, disconnected components (clusters)
  - Within each component, all members hold the same opinion. However, the specific opinion may differ across different components, leading to global polarization

Coevolution of Networks and Dynamics (5/7)

universität koblenz
weiter:denken

- **Stable Equilibrium:** This final fragmented state represents a stable state (equilibrium) for the coevolutionary system
  - At this point, no further changes occur in either individual opinions or the network's structure, and the dynamics cease



Network Search (1/2)

- The process of locating a specific resource or target node within a given network
- To achieve this, it's essential to devise efficient strategies for exploring the network until the desired node is successfully identified and reached
- Exploration Method: Typically, the search starts from a designated "node of origin" and proceeds by progressively visiting its immediate neighbors, then their neighbors, and so on, radiating outwards

Network Search (2/2)

- Exhaustive Approaches
  - Breadth-First Search (BFS): An example of an exhaustive algorithm that systematically explores all reachable nodes layer by layer
  - While effective in some scenarios, particularly for smaller networks or when abundant computational and storage resources are available (e.g., in web crawlers for search engine indexing), BFS can be computationally intensive for large-scale networks
  - **Better Strategy**: In many real-world network scenarios, a local search strategy is often a more efficient and practical approach

Local Search

- Breadth-First Search (BFS) for Search
  - BFS can be used as a general search mechanism to find a target node, even if it's not typically categorized as "local search" in the context of taking advantage of network structure
  - Starting from a source node, the process involves exhaustively visiting all nodes in the first layer (immediate neighbors), then checking if any are the target
  - If not, the query is propagated to their neighbors, and so on, until the target node is reached
- Limitations of BFS
  - Breadth-First Search is generally **not** an efficient strategy for optimizing local search in complex networks
  - This is because BFS does not leverage specific properties of network structure (e.g., small-world phenomenon, presence of hubs, community structure) that could guide the search more directly towards the target, thus requiring unnecessary exploration of a vast number of nodes

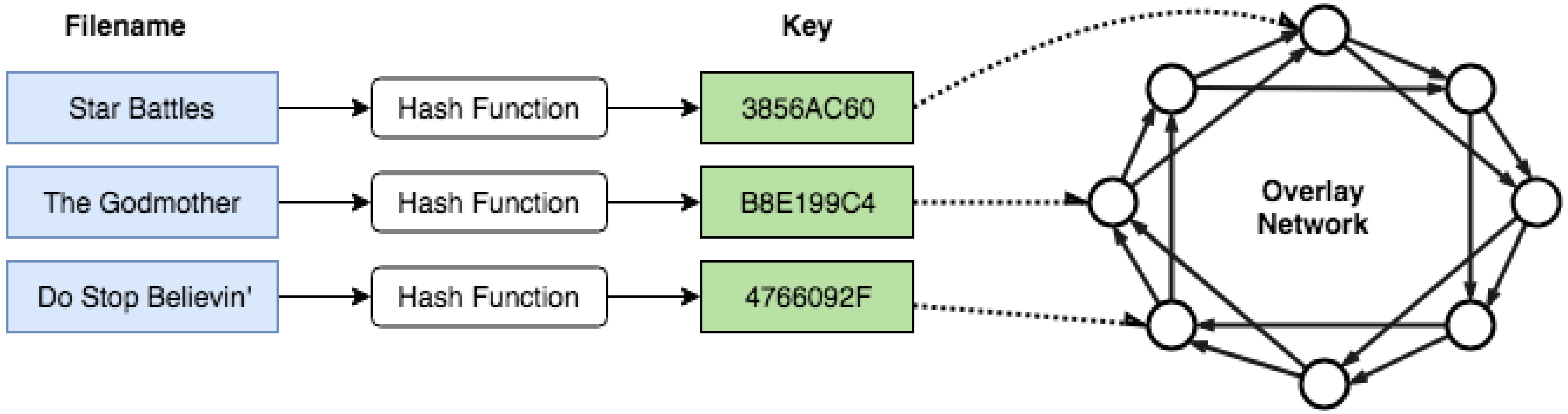
Peer-to-Peer (P2P) Networks (3/5)

P2P networks fundamentally rely on two interconnected components to facilitate decentralized resource sharing

- **Distributed Hash Table (DHT):** A decentralized system that maps unique file identifiers (keys, often cryptographic hashes of the file content) to the specific peer computers that store those files
  - This mechanism allows any peer to efficiently locate a file without requiring a central server
- **Overlay Network:** A virtual network built on top of the underlying physical network infrastructure. It connects the peer nodes that participate in the P2P system
  - This overlay dictates how queries and data requests propagate between peers, enabling the discovery and transfer of files

Peer-to-Peer (P2P) Networks (4/5)

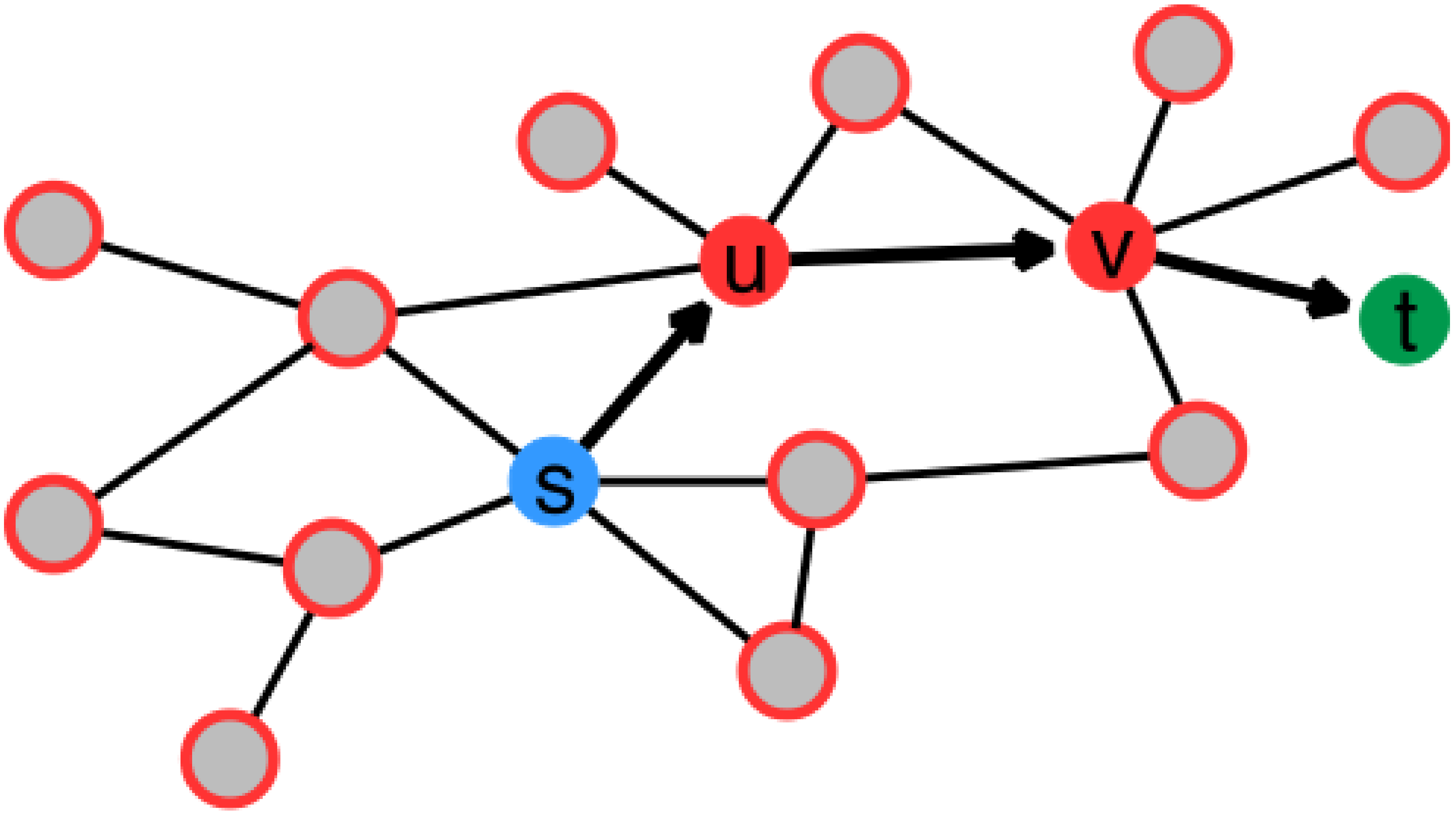
- **File Storage Mechanism**
  - When a file needs to be stored within the P2P network, a unique key is generated for that file
  - This key is produced by a hash function, which is a cryptographic algorithm designed to output a fixed-size, unique signature from any arbitrary input data
  - Crucially, each generated key is mapped to a specific node (peer computer) within the network's address space, enabling the file to be efficiently routed to, and retrieved from, that particular peer
- **File Retrieval Process**
  - During a file search operation, the unique key associated with the desired file is utilized to forward the query through the overlay network
  - This query propagation continues iteratively until it successfully reaches the specific node that possesses the file corresponding to that key



Local Search: Exploiting Network Structure (1/3)

- **Approach**: This strategy is designed to improve search efficiency by exploiting the presence of hubs (nodes with high connectivity) within the network
- **Dynamics of Search**
  - All information accessible to any given node is strictly local. Each node knows the degree of all its immediate neighbors, as well as the data (resources) stored within those neighbors
  - When a node is queried, starting from the initial source node, it forwards the request to its neighbor with the largest degree, unless the current node itself or any of its immediate neighbors is the target
  - This forwarding process is repeated iteratively until the message successfully reaches a neighbor of the target node, signifying the target's discovery
  - To avoid redundant queries and infinite loops, nodes that have already passed the request are marked as visited, ensuring that no node is queried more than once during a specific search procedure

Local Search: Exploiting Network Structure (2/3)



Local Search: Exploiting Network Structure (3/3)

universität koblenz
weiter:denken

- By preferentially exploring neighbors with a large degree (high connectivity), the search algorithm significantly increases the probability of encountering one of their neighbors that is also a major hub
  - During this initial "greedy" phase, the algorithm rapidly converges to a node with the highest degree in the local vicinity
  - Following this rapid transient phase, where visited nodes progressively exhibit larger degrees, the subsequent exploration effectively proceeds in an inverse order of the network's degree sequence—spreading downwards from the highly connected hub
  - This leads to a very rapid growth in the number of queried nodes (specifically, the neighbors of the hubs), resulting in the target being reached in a relatively small number of steps
- **Problem**: On average, the total number of nodes that must be queried by this local search strategy is comparable to that of a full Breadth-First Search
  - This is because, in principle, the target node can be located anywhere within the network

Web Searchability

- For the World Wide Web, the concept of geographic homophily is replaced by topical locality
- This means that web pages with similar content or topics are highly likely to have common neighbors or be directly linked
- Even for distantly related (dissimilar) pages, the decay in link probability is compatible with the condition for geographic searchability
- As a result, despite its vastness, **the Web is empirically observed to be searchable**!

Network Theory and Dynamic Systems

11. Information Cascades

SOSE 2025

Dr. -Ing. Stefania Zourlidou

Institute for Web Science and Technologies
Universität Koblenz

Following the Crowd (1/4)

- When people are connected by a network, it becomes possible for them to influence each other's behavior and decisions
- **Example 1**
  - Suppose that you are choosing a restaurant in an unfamiliar town, and based on your own research about restaurants, you intend to go to restaurant **A**
  - However, when you arrive you see that no one is eating in restaurant A, whereas restaurant B next door is nearly full
  - If you believe that other diners have tastes similar to yours, and that they too have some information about where to eat, it may be rational to join the crowd at **B** *rather than to follow your own information*

Following the Crowd (3/4)

- **Information cascade**, comes from the work of Banerjee
  - the concept was also developed in other work around the same time by Bikhchandani, Hirshleifer, and Welch
- An information cascade has the potential to occur when people make decisions sequentially, with later people watching the actions of earlier people and from these actions inferring something about what the earlier people know
- In the restaurant example, when the first diners to arrive chose restaurant B, they conveyed information to later diners *about what they knew*
- A *cascade* then develops when *people abandon their own information in favor of inferences based on earlier people's actions*
- Individuals in a cascade are imitating the behavior of others, but it is *not* mindless imitation
  - Rather, it is the result of drawing rational inferences from limited information

Following the Crowd (4/4)

universität koblenz
weiter:denken

Example 2 (Milgram, Bickman, and Berkowitz in the 1960s)

- The experimenters had groups of people, ranging in size from just one person to as many as fifteen people, stand on a street corner and stare up into the sky
- They then observed how many passersby stopped and also looked up at the sky
- They found that with only one person looking up, very few passersby stopped
- If five people were staring up into the sky, then more passersby stopped, but most still ignored them
- Finally, with fifteen people looking up, they found that 45% of passersby stopped and also stared up into the sky
- The experimenters interpreted this result as demonstrating a social force for conformity that grows stronger as the group conforming to the activity becomes larger
- But another possible explanation – essentially, a possible mechanism giving rise to the conformity observed in this kind of situation – is rooted in the idea of information cascades

A Simple Herding Experiment (1/8)

- A simple herding experiment created by Anderson and Holt to illustrate how the mathematical models for information cascade work

(a) There is a decision to be made – for example, whether to adopt a new technology, wear a new style of clothing, eat in a new restaurant, or support a particular political position

(b) People make decisions sequentially, and each person can observe the choices made by those who acted earlier

(c) Each person has some private information that helps guide their decision

(d) A person can’t directly observe the private information that other people *know*, but he or she can make inferences about this private information from what they *do*

A Simple Herding Experiment (2/8)

- Imagine the experiment taking place in a classroom, with a large group of students
- The experimenter puts an urn at the front of the room with three marbles hidden in it; she announces that there is a 50% chance that the urn contains two red marbles and one blue marble ("majority-red" urn), and a 50% chance the urn contains two blue marbles and one red marble ("majority-blue" urn)
- One by one, each student comes to the front of the room and draws a marble from the urn
- He looks at the color and then places it back in the urn **without** showing it to the rest of the class
- The student then guesses whether the urn is majority-red or majority-blue and publicly announces this guess to the class
  - The public announcement is the key part of the setup: the students who have not yet had their turn don't get to see which colors the earlier students draw, but they do get to hear the guesses that are being made
  - This parallels the previous example with the two restaurants: one by one, each diner needs to guess which is the better restaurant, and while they don't get to see the reviews read by the earlier diners, they do get to see which restaurant these earlier diners chose

A Simple Herding Experiment (3/8)

Let's now consider what we should expect to happen when this experiment is performed

- We will assume that all the students reason *correctly* about what to do when it is their turn to guess, using everything they have heard so far

- Things are fairly straightforward for the first two students; they become interesting once we reach the third student
- The First Student
  - The first student should follow a simple decision rule for making a guess: if he sees a red marble, it is better to guess that the urn is majority-red; and if he sees a blue marble, it is better to guess that the urn is majority-blue
  - This means the first student's guess conveys perfect information about what he has seen

A Simple Herding Experiment (4/8)

universität koblenz
weiter:denken

- The Second Student
  - If the second student sees the same color that the first student announced, then her choice is simple: she should guess this color as well. However, suppose she sees the opposite color – say that she sees red while the first guess was blue. Since the first guess was exactly what the first student saw, the second student can essentially reason as though she got to draw twice from the urn, seeing blue once and red once. In this case, she is indifferent about which guess to make; we will assume in this case that she breaks the tie by guessing the color she saw. Thus, whichever color the second student draws, her guess too conveys perfect information about what she has seen
- The Third Student
  - Things start to get interesting here. If the first two students have guessed opposite colors, then the third student should just guess the color he sees, since it will effectively break the tie between the first two guesses
  - But suppose the first two guesses have been the same – say they've both been blue – and the third student draws red. Since we've decided that the first two guesses convey perfect information, the third student can reason in this case as though he saw three draws from the urn: two blue, and one red. Given this information, he should guess that the urn is majority-blue, ignoring his own private information

A Simple Herding Experiment (5/8)

- The Third Student (cont.)
  - More generally, the point is that when the first two guesses are the same, the third student should guess this color as well, regardless of which color he draws from the urn. And the rest of class will only hear his guess; they don’t get to see which color he’s drawn
  - In this case, an *information cascade has begun*
  - The third student makes the same guess as the first two, *regardless of which color he draws from the urn*, and hence regardless of his own private information

A Simple Herding Experiment (6/8)

- The Fourth Student and Onward
  - Let's consider just the "interesting" case of the third student discussed above, in which the first two guesses were the same – suppose they were both blue. In this case, we've argued that the third student will also announce a guess of blue, regardless of what he actually saw
  - Now consider the situation faced by the fourth student, getting ready to make a guess having heard three guesses of "blue" in a row. She knows that the first two guesses conveyed perfect information about what they saw. She also knows that, given this, the third student was going to guess "blue" no matter what he saw – so his guess conveys no information
  - As a result, the fourth student is in exactly the same situation – from the point of view of making a decision – as the third student. Whichever color she draws, it will be outweighed by the two draws of blue by the first two students, and so she should guess "blue" regardless of what she sees
  - This will continue with all the subsequent students: if the first two guesses are "blue," then everyone in order will guess "blue" as well (a completely symmetric thing happens if the first two guesses are "red"). An information cascade has taken hold: no one is under the illusion that every single person is drawing a blue marble, but once the first two guesses turn out "blue," the future announced guesses become worthless and so everyone's best strategy is to rely on the limited genuine information they have available

Bayes's Rule: A Model of Decision Making under Uncertainty (1/5)

universität koblenz
weiter:denken

What is the probability this is the better restaurant, given the reviews I've read and the crowds I see in each one?

Or

What is the probability this urn is majority-red, given the marble I just drew and the guesses I've heard?

Bayes's Rule: A Model of Decision Making under Uncertainty (2/5)

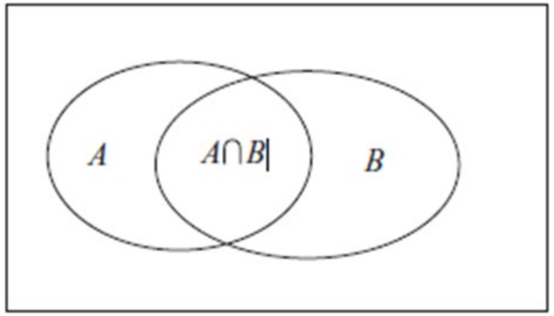
universität koblenz
weiter:denken

- **Conditional Probability and Bayes' Rule:** We will compute the probabilities of various *events*, and use these probabilities to reason about decision making
  - In the context of the previous experiment, an event could be "the urn is majority-blue," or "the first student draws a blue marble"
  - Given any event A, we will denote its probability of occurring by Pr[A]
  - Whether an event occurs or not is the result of certain random outcomes
  - We imagine a large sample space in which each point in the sample space consists of a particular realization for each of these random outcomes

Bayes's Rule: A Model of Decision Making under Uncertainty (4/5)

- **Example**: A may be the event that the urn in the previously mentioned experiment is majority-blue, and B may be the event that the ball you've drawn is blue
- We will refer to this quantity as the *conditional probability of A given B* and denote it by Pr [A | B]
- We can think of this as the fraction of the area of region B occupied by A ∩ B, and so we define

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$



- Similarly, the conditional probability of B given A is

$$\Pr[B \mid A] = \frac{\Pr[B \cap A]}{\Pr[A]} = \frac{\Pr[A \cap B]}{\Pr[A]}$$

Bayes's Rule: A Model of Decision Making under Uncertainty (5/5)

- where the second equality follows simply because A ∩ B and B ∩ A are the same set. Rewriting previous Equations , we have

$$\Pr[A \mid B] \times \Pr[B] = \Pr[A \cap B] = \Pr[B \mid A] \times \Pr[A]$$

- and therefore, dividing through by Pr [B]

$$\Pr[A \mid B] = \frac{\Pr[A] \times \Pr[B \mid A]}{\Pr[B]}$$

- This Equation is called *Bayes' rule*
- When we want to make explicit that we're interested in the effect of event B on the probability of an event A, we refer to Pr [A] as the *prior probability* of A, since it reflects our understanding of the probability of A without knowing anything about whether B has occurred
- We refer to Pr [A | B] as the *posterior probability* of A given B, since it reflects our new understanding of the probability of A now that we know B has occurred

Bayes's Rule in the Herding Experiment (1/5)

- Each student's decision is intrinsically based on determining a conditional probability
  - he is trying to estimate the conditional probability that the urn is majority-blue or majority-red, given what she has seen and heard
- To maximize her chance of winning the monetary reward for guessing correctly, she should guess majority-blue if

$$\Pr\left[\textit{majority-blue} \mid \textit{what she has seen and heard}\right] > \frac{1}{2}$$

- and guess majority-red otherwise
- If the two conditional probabilities are both exactly 0.5, then it doesn't matter what she guesses
- Also we know the prior probabilities of majority-blue and majority-red are each 1/ 2

$$\Pr\left[\textit{majority-blue}\right] = \Pr\left[\textit{majority-red}\right] = \frac{1}{2}$$

Bayes's Rule in the Herding Experiment (2/5)

universität koblenz
weiter:denken

- Also, based on the composition of the two kinds of urns

$$\Pr[blue \mid majority\text{-}blue] = \Pr[red \mid majority\text{-}red] = \frac{2}{3}$$

- Suppose that the first student draws a blue marble. He therefore wants to determine Pr [majority-blue | blue] and, just as in the examples from, he can use Bayes' rule to calculate

$$\Pr[majority\text{-}blue \mid blue] = \frac{\Pr[majority\text{-}blue] \times \Pr[blue \mid majority\text{-}blue]}{\Pr[blue]}$$

- The numerator is $\frac{1}{2} \times \frac{2}{3} = \frac{1}{3}$. For the denominator, we reason just as we did before, by noting that there are two possible ways to get a blue marble – if the urn is majority-blue or if it is majority-red

$$\begin{aligned}\Pr[blue] &= \Pr[majority\text{-}blue] \times \Pr[blue \mid majority\text{-}blue] \\ &\quad + \Pr[majority\text{-}red] \times \Pr[blue \mid majority\text{-}red] \\ &= \frac{1}{2} \times \frac{2}{3} + \frac{1}{2} \times \frac{1}{3} = \frac{1}{2}.\end{aligned}$$

Bayes's Rule in the Herding Experiment (3/5)

- The answer Pr [blue] = 1 / 2 makes sense, given that the roles of blue and red in this experiment are completely symmetric. Dividing numerator by denominator, we get

$$\Pr[\textit{majority-blue} \mid \textit{blue}] = \frac{1/3}{1/2} = \frac{2}{3}.$$

- Since this conditional probability is greater than 1/2 , we get the intuitive result that the first student should guess majority-blue when he sees a blue marble. Note that in addition to providing the basis for the guess, Bayes' rule provides a probability, namely 2/3 , that the guess will be correct
- The calculation is very similar for the second student, and we skip it so as to move on to the calculation for the third student, where a cascade begins to form
- Let's suppose, that the first two students have announced guesses of blue, and the third student draws a red marble. As we discussed there, the first two guesses convey genuine information, so the third student knows that there have been three draws from the urn, consisting of the sequence of colors blue, blue. and red. What he wants to know is

$$\Pr[\textit{majority-blue} \mid \textit{blue}, \textit{blue}, \textit{red}]$$

Bayes's Rule in the Herding Experiment (4/5)

- So as to make a guess about the urn. Using Bayes' rule we get

$$\begin{aligned}\Pr[\textit{majority-blue} \mid \textit{blue, blue, red}] \\ = \frac{\Pr[\textit{majority-blue}] \times \Pr[\textit{blue, blue, red} \mid \textit{majority-blue}]}{\Pr[\textit{blue, blue, red}]}\end{aligned}$$

- Since the draws from the urn are independent, the probability Pr[blue, blue,red| majority-blue] is determined by multiplying the probabilities of the three respective draws together:

$$\Pr[\textit{blue, blue, red} \mid \textit{majority-blue}] = \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} = \frac{4}{27}$$

- To determine Pr [blue, blue, red], as usual we consider the two different ways this sequence could have happened – if the urn is majority-blue or if it is majority-red

$$\begin{aligned}\Pr[\textit{blue, blue, red}] &= \Pr[\textit{majority-blue}] \times \Pr[\textit{blue, blue, red} \mid \textit{majority-blue}] \\ &\quad + \Pr[\textit{majority-red}] \times \Pr[\textit{blue, blue, red} \mid \textit{majority-red}] \\ &= \frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} + \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} = \frac{6}{54} = \frac{1}{9}.\end{aligned}$$

Bayes's Rule in the Herding Experiment (5/5)

- Plugging all this back into Equation, we get

$$\Pr\left[\textit{majority-blue} \mid \textit{blue}, \textit{blue}, \textit{red}\right] = \frac{\frac{4}{27} \times \frac{1}{2}}{\frac{1}{9}} = \frac{2}{3}$$

- Therefore, the third student should guess majority-blue (from which he will have a 2/3 chance of being correct)
- This outcome confirms our intuitive observation that the student should ignore what he sees (red) in favor of the two guesses he's already heard (both blue)
- Finally, once these three draws from the urn have taken place, all future students will have the same information as the third student, and so they will all perform the same calculation, resulting in an information cascade of blue guesses

A Simple, General Cascade Model (1/12)

Let's return to the motivation for the herding experiment

- The experiment served as a stylized metaphor for any situation in which people make decisions sequentially, basing these decisions on a combination of their own private information and observations of what earlier people have done
- We now formulate a model that covers such situations in general
  - We will see that Bayes' rule predicts in this general model that cascades will form, with probability tending to 1 as the number of people goes to infinity