Universiteit Leiden

Social Network Analysis for Computer Scientists

Frank Takes

LIACS, Leiden University

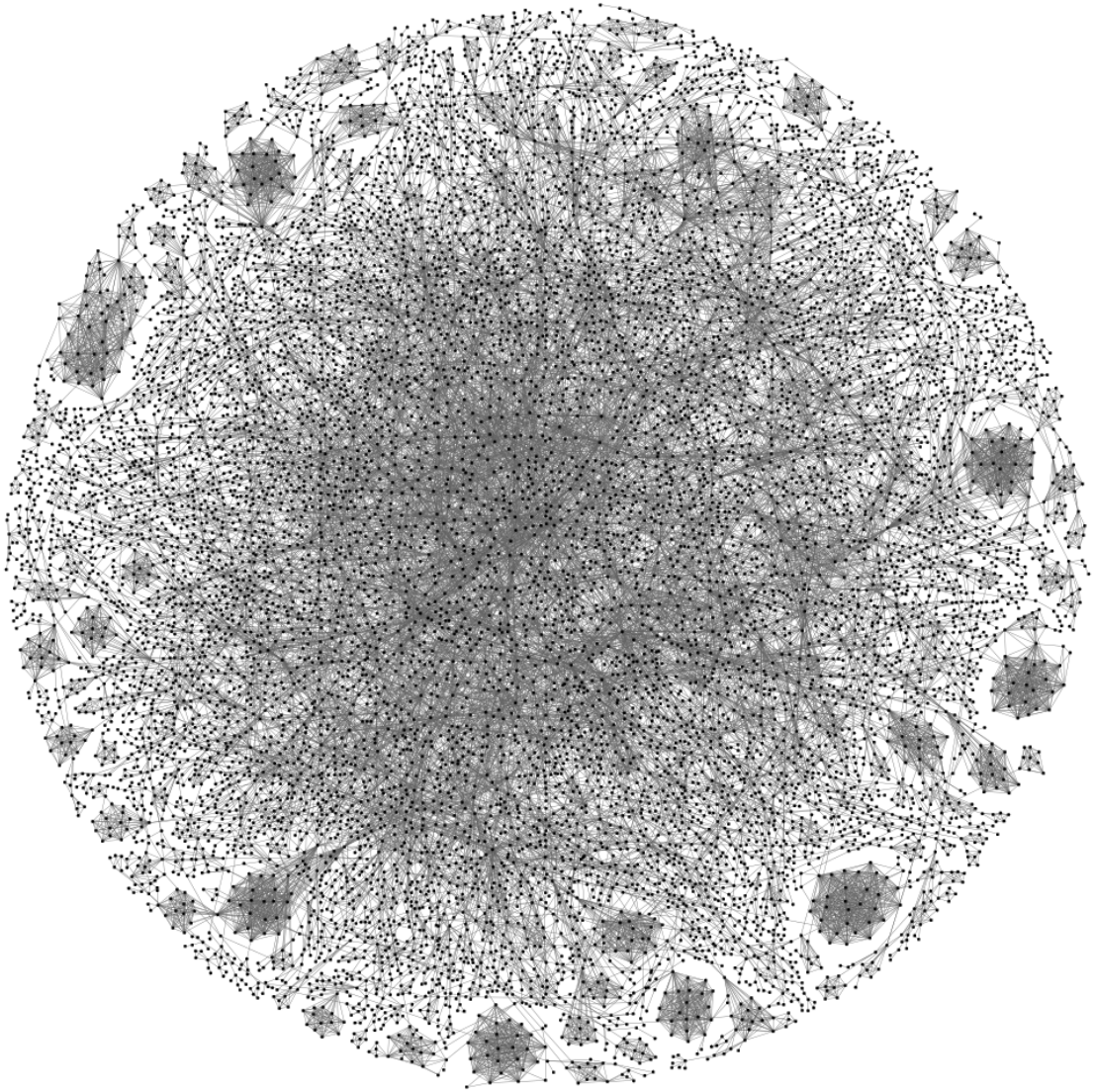
https://liacs.leidenuniv.nl/~takesfw/SNACS

Lecture 2 — Advanced network concepts and centrality

Universiteit Leiden

Recap

Networks



Notation

Concept	Symbol
Network (graph)	$G = (V, E)$
Nodes (objects, vertices, ...)	V
Links (ties, relationships, ...)	E
Directed — $E \subseteq V \times V$ — "links"	
Undirected — "edges"	
Number of nodes — $\lvert V \rvert$	n
Number of edges — $\lvert E \rvert$	m
Degree of node u	$deg(u)$
Distance from node u to v	$d(u, v)$

Real-world networks

Universiteit Leiden

1. Sparse networks — density
2. Fat-tailed power-law degree distribution — degree
3. Giant component — components
4. Low pairwise node-to-node distances — distance
5. Many triangles — clustering coefficient

- Many examples: communication networks, citation networks, collaboration networks (Erdös, Kevin Bacon), protein interaction networks, information networks (Wikipedia), webgraphs, financial networks (Bitcoin) ...

Universiteit Leiden

Advanced concepts

Advanced concepts

- Assortativity
- Reciprocity
- Power law exponent
- Planar graphs
- Complete graphs
- Subgraphs
- Trees
- Spanning trees
- Diameter
- Bridges
- Graph traversal

Assortativity

- **Assortativity**: extent to which "similar" nodes attract each other
  Value close to -1 if dissimilar nodes more often attract each other
  Value close to 1 if similar nodes more often attract each other

Assortativity

- **Assortativity**: extent to which "similar" nodes attract each other
  Value close to -1 if dissimilar nodes more often attract each other
  Value close to 1 if similar nodes more often attract each other
- Degree assortativity: nodes with a similar degree connect more frequently
- Attribute assortativity: nodes with similar attributes attract each other
- Influence on connectivity of network, spreading of information, etc.
- Social networks: **homophily**
- Complex networks: **mixing patterns**

Degree assortativity

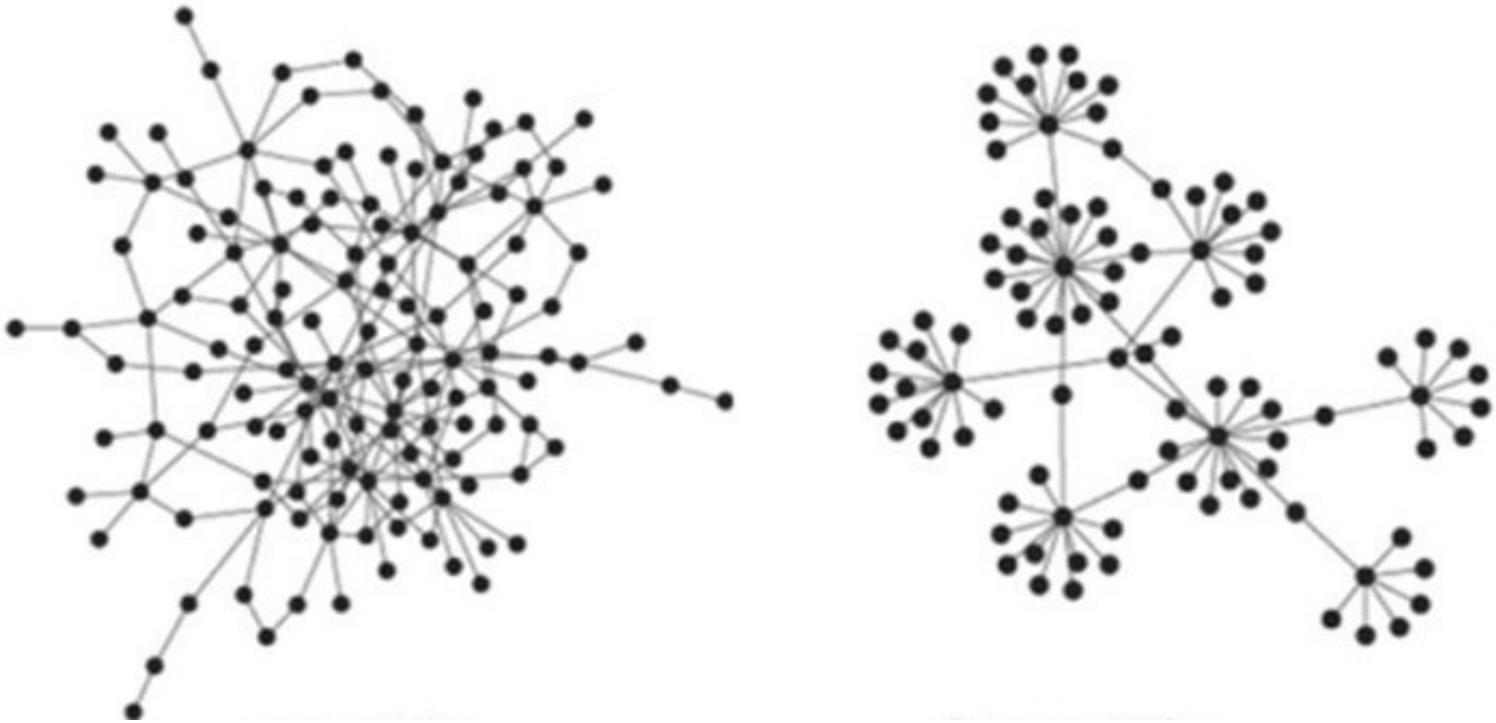


Figure: Degree **assortativity** (left) and degree **disassortativity** (right)

Image: Estrada et al., Clumpiness mixing in complex networks, J. Stat. Mech. Theor. Exp. P03008 (2008).

Reciprocity

- **Reciprocity**: measure of the likelihood of nodes in a directed network to be mutually linked
- Let $m_{<->}$ be the number of links in the directed network for which there also exists a symmetric counterpart:

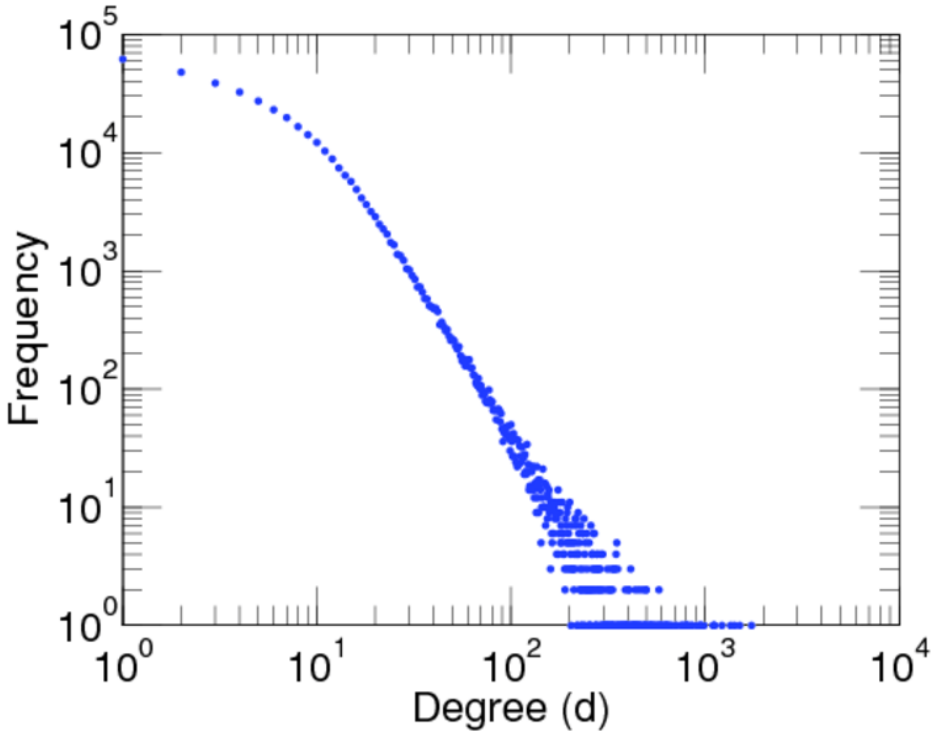
$$m_{<->} = |\{(u, v) \in E : (v, u) \in E\}|$$

- Reciprocity r is then the fraction of links that is symmetric:

$$r = \frac{m_{<->}}{m}$$

- Measures the extent to which relationships are mutual
- Useful to compare between networks

Power law degree distribution



Source: `http://konect.cc/networks/citeseer/`

Power law exponent in undirected networks

- The probability p_k of a node having degree k depends on the power law exponent γ:

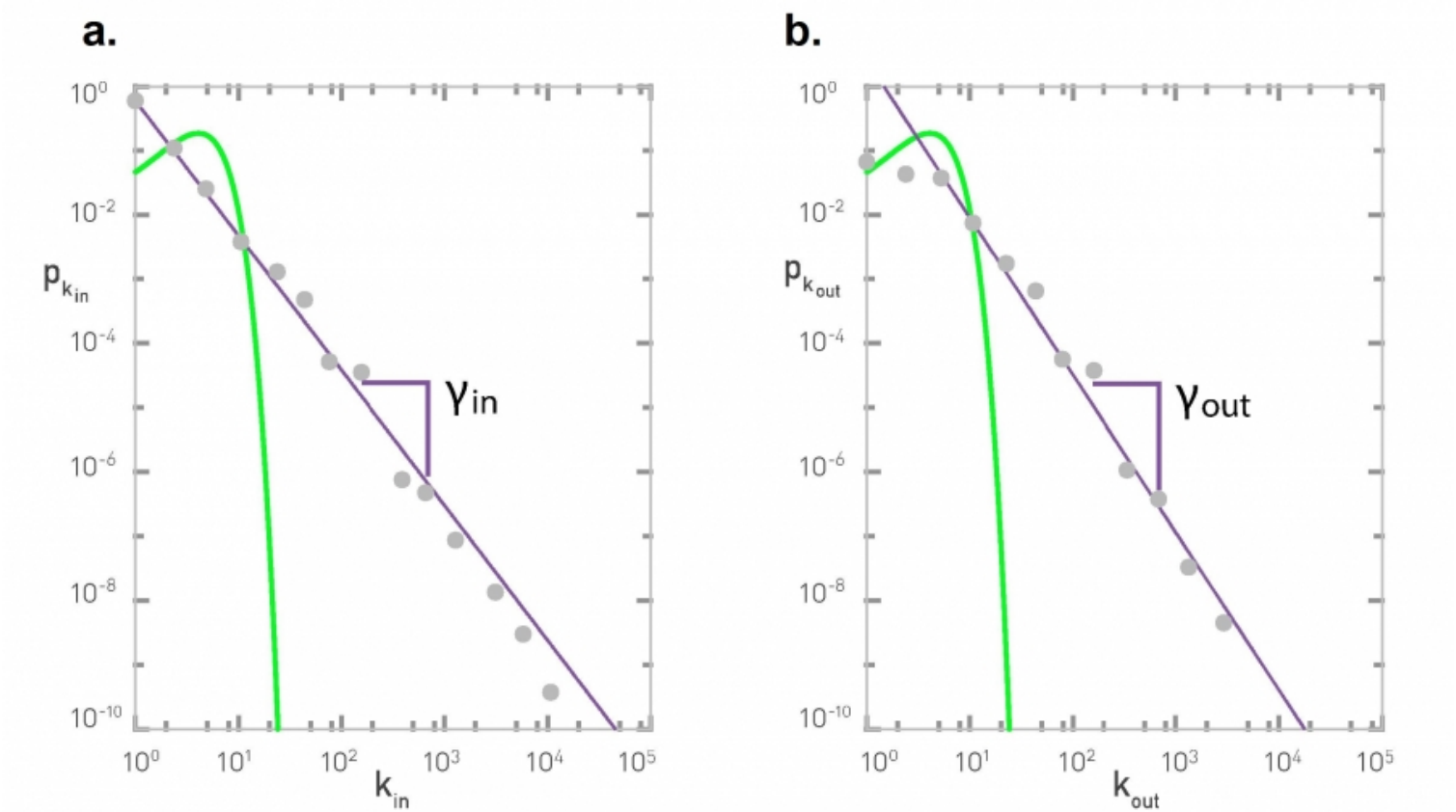
$$p_k \sim k^{-\gamma}$$

- This means that

$$\log p_k \sim -\gamma \log k$$

  And as such, the straight line in log-log scale plots is observed.
- In real-world networks, γ has a value of around 2 to 3
- Useful to compare between similar networks

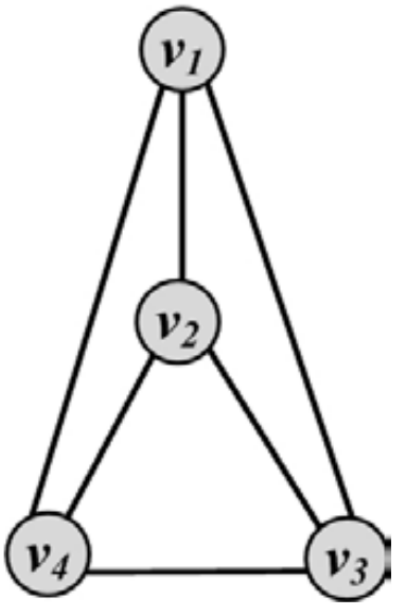
Power law exponent in directed networks



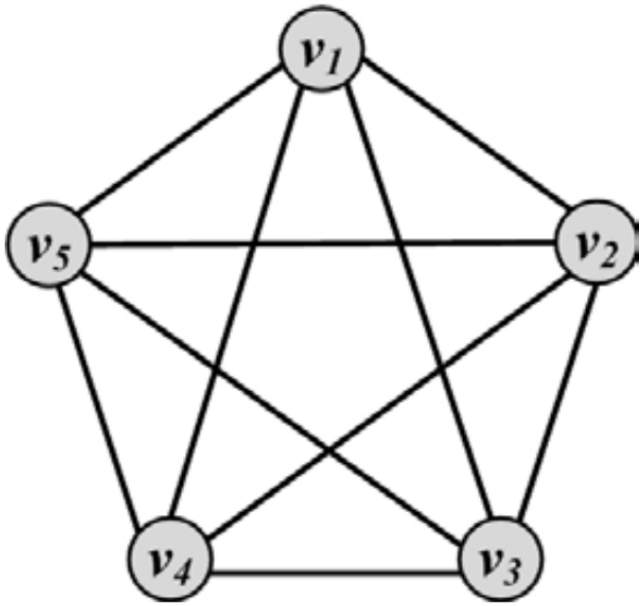
Source: A. Barabasi, Network Science, 2016.

Planar graphs

- **Planar graphs** can be visualized such that no two edges cross each other



(a) Planar Graph



(b) Non-planar Graph

Image: Zafarani et al., Social Media Mining, 2014.

Complete graphs

- In **complete graphs**, all pairs of nodes are connected
- The number of edges m is equal to $\frac{1}{2} \cdot n \cdot (n-1)$

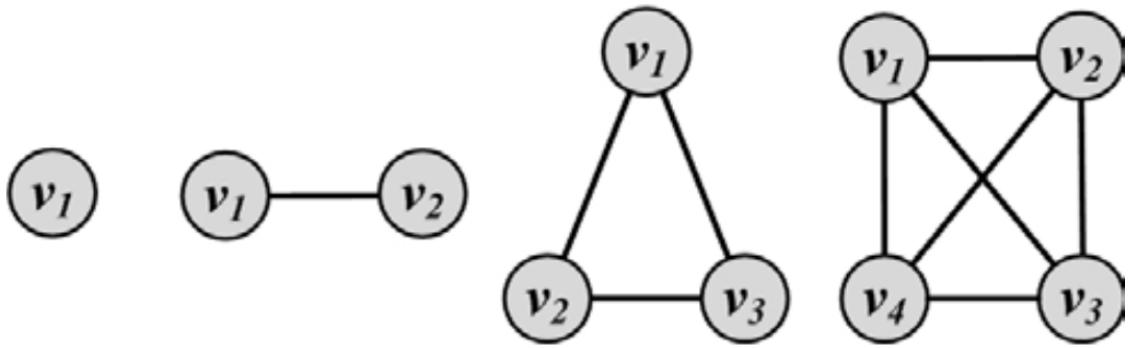


Figure: Complete graphs of size 1, 2, 3 and 4

Image: Zafarani et al., Social Media Mining, 2014.

Trees

- A **tree** is a graph without cycles
- A set of disconnected trees is called a **forest**
- A tree with n nodes has $m = n - 1$ edges

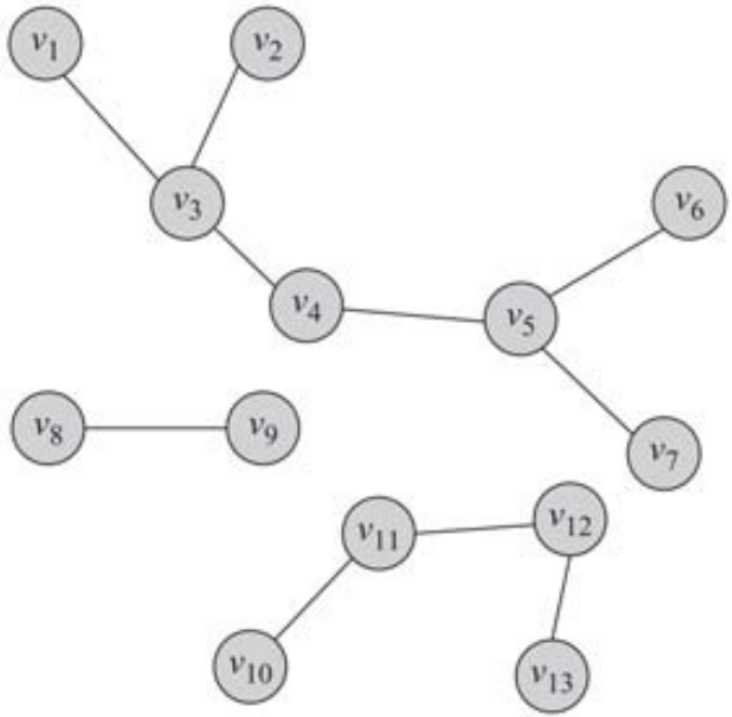


Image: Zafarani et al., Social Media Mining, 2014.

Trees

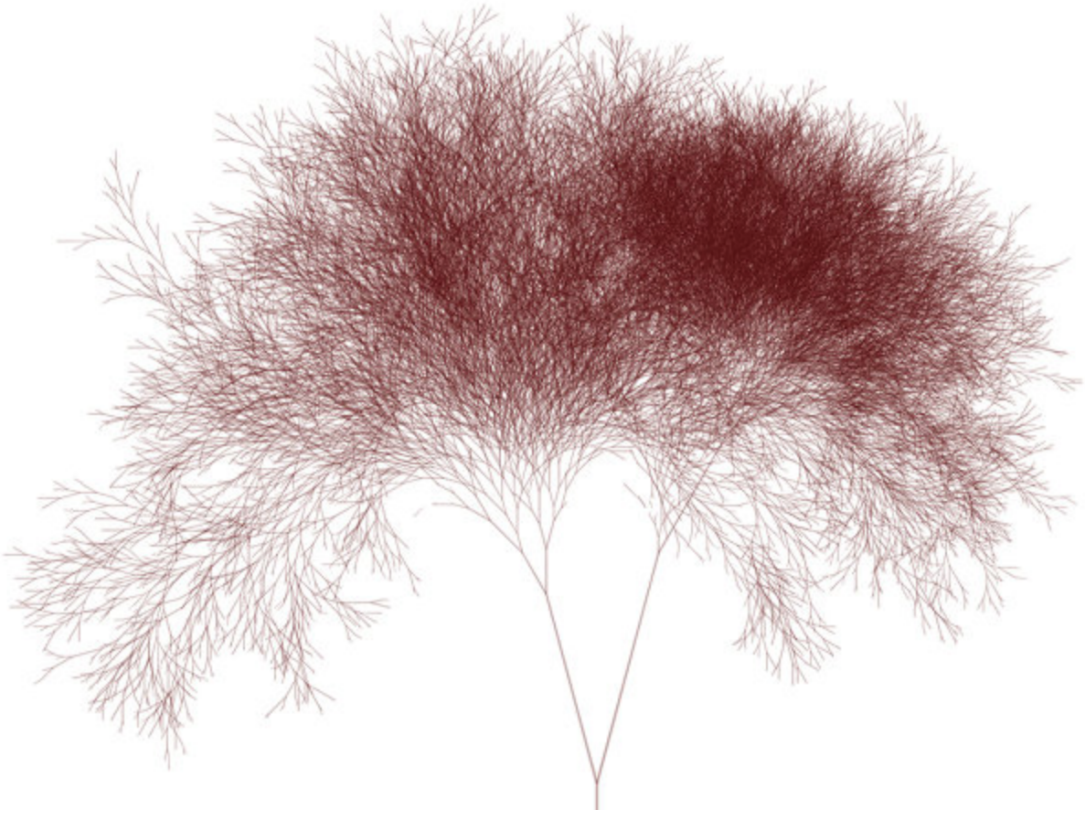


Image: M. Lima, Book of trees: Visualizing branches of knowledge, 2014.

Subgraphs

- Given a graph $G = (V, E)$
- **Subgraph** $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq (E \cap (V' \times V'))$
  (subset of the nodes and edges of the original network, commonly used when defining communities or clusters)
- **Subgraph** $G' = (V, E')$ with $E' \subseteq E$
  (only edges are left out, commonly used when modelling network evolution)
- Special subgraphs: spanning trees

Spanning trees

- A **spanning tree** is a tree and subgraph of a graph that covers all nodes of the graph
- In weighted graphs, a **minimal** spanning tree is one of minimal edge weight

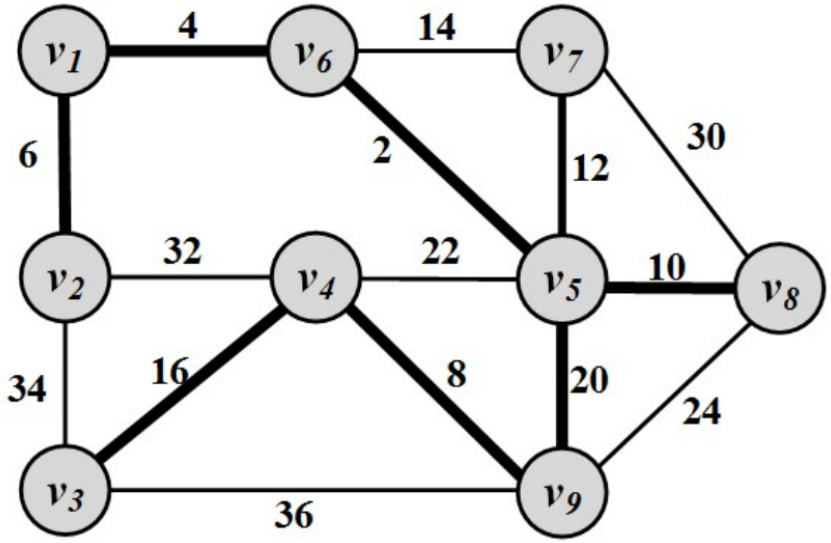


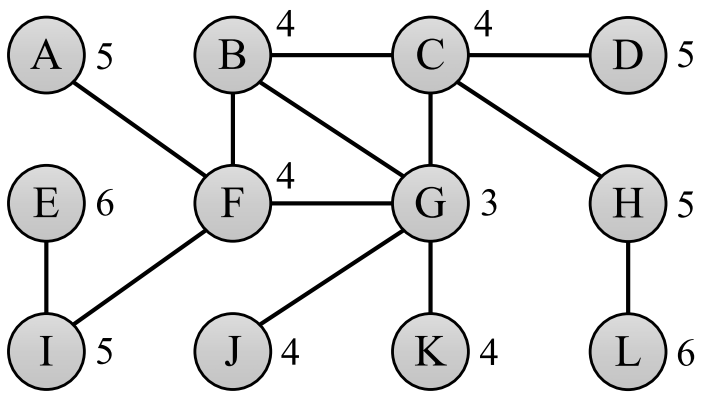
Image: Zafarani et al., Social Media Mining, 2014.

Diameter

- Distance $d(u, v) =$ length of shortest path from u to v
- Diameter $D(G) = \max_{u,v \in V} d(u, v) =$ maximal distance

Diameter

- Distance $d(u, v)$ = length of shortest path from u to v
- Diameter $D(G) = \max_{u,v \in V} d(u, v)$ = maximal distance
- Eccentricity $e(u) = \max_{v \in V} d(u, v)$ = length of longest shortest path from u
- Diameter $D(G) = \max_{u \in V} e(u)$ = maximal eccentricity
- Radius $R(G) = \min_{u \in V} e(u)$ = minimal eccentricity



Bridges

- **Bridge**: an edge whose removal will result in an increase in the number of connected components
- Also called **cut edges**, with applications in community detection

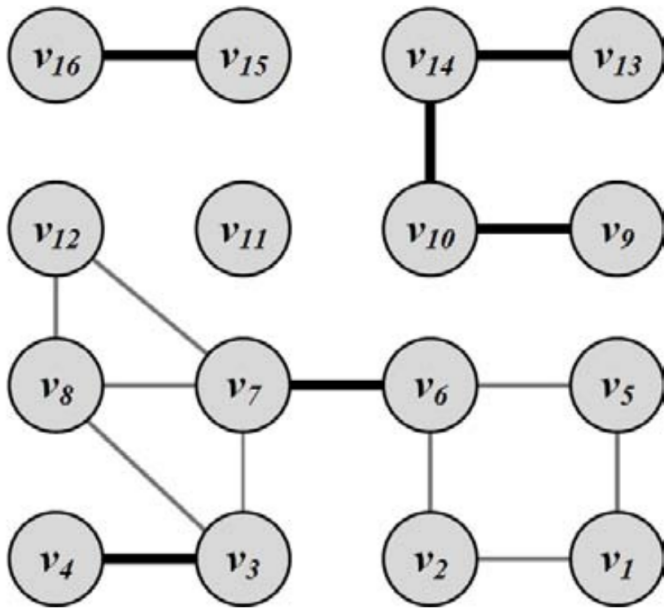


Image: Zafarani et al., Social Media Mining, 2014.

Graph traversal

- Given a network, how can we explore it?
- Specifically: exploration starting from a particular source
- Node **adjacency**: two nodes are adjacent if there is an edge connecting them
- **Neighborhood**: set of nodes adjacent to a node $v \in V$:

$$N(v) = \{w \in V : (v, w) \in E\}$$

- Techniques to iteratively explore neighborhoods: DFS and BFS

Graph traversal: DFS

- Depth First Search (DFS)

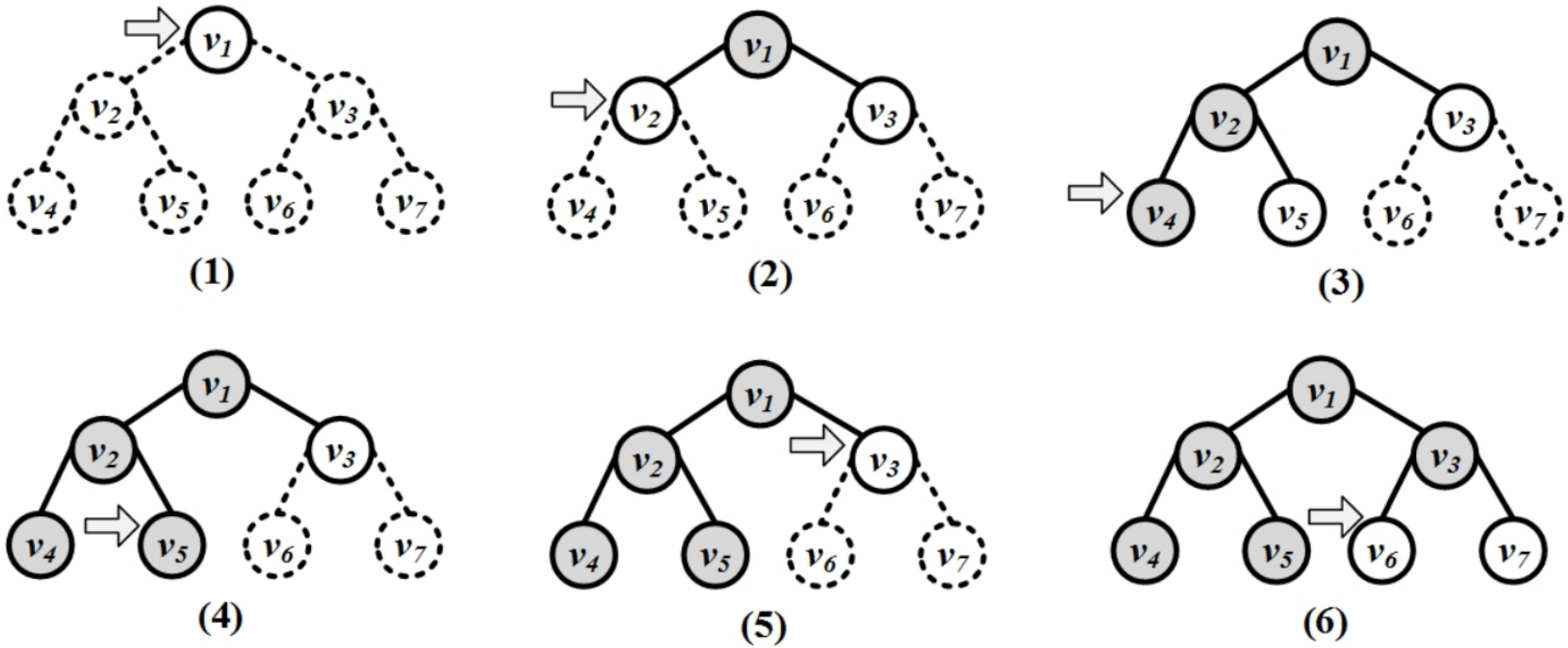
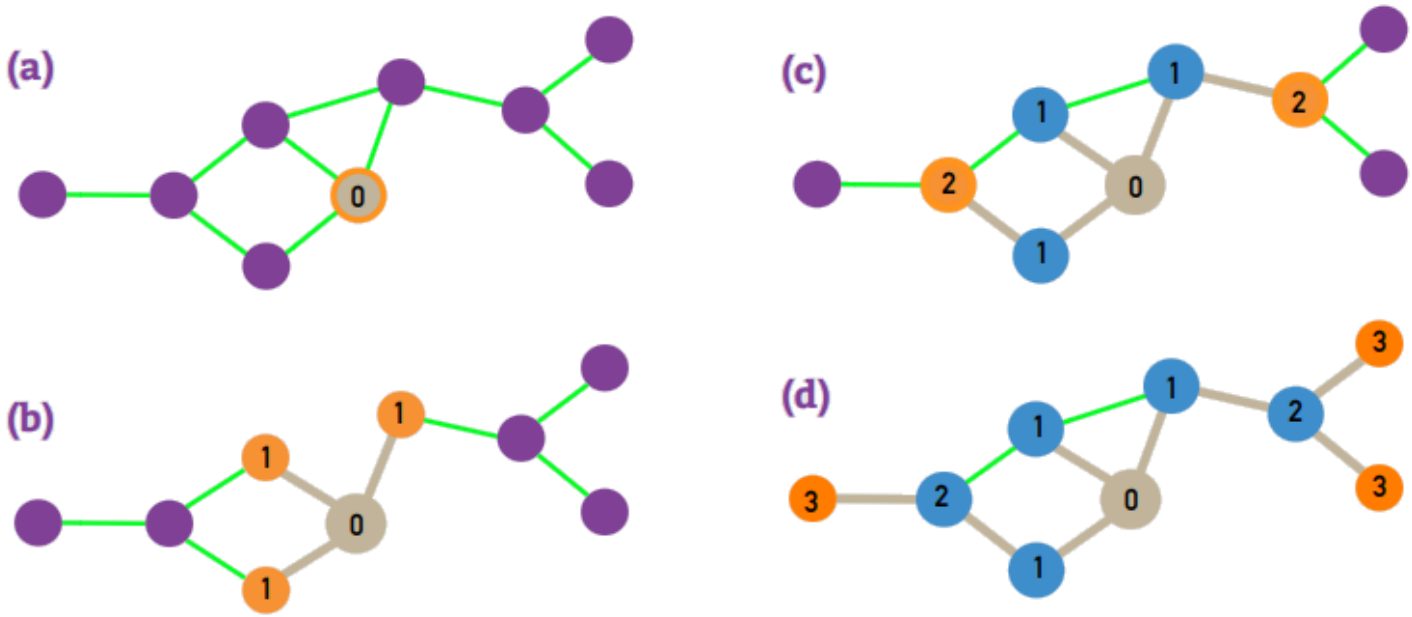


Image: Zafarani et al., Social Media Mining, 2014.

Graph traversal: BFS



Source: A. Barabasi, Network Science, 2016.

Graph traversal: BFS

- **Breadth First Search** (BFS)
- Graph traversal in level-order

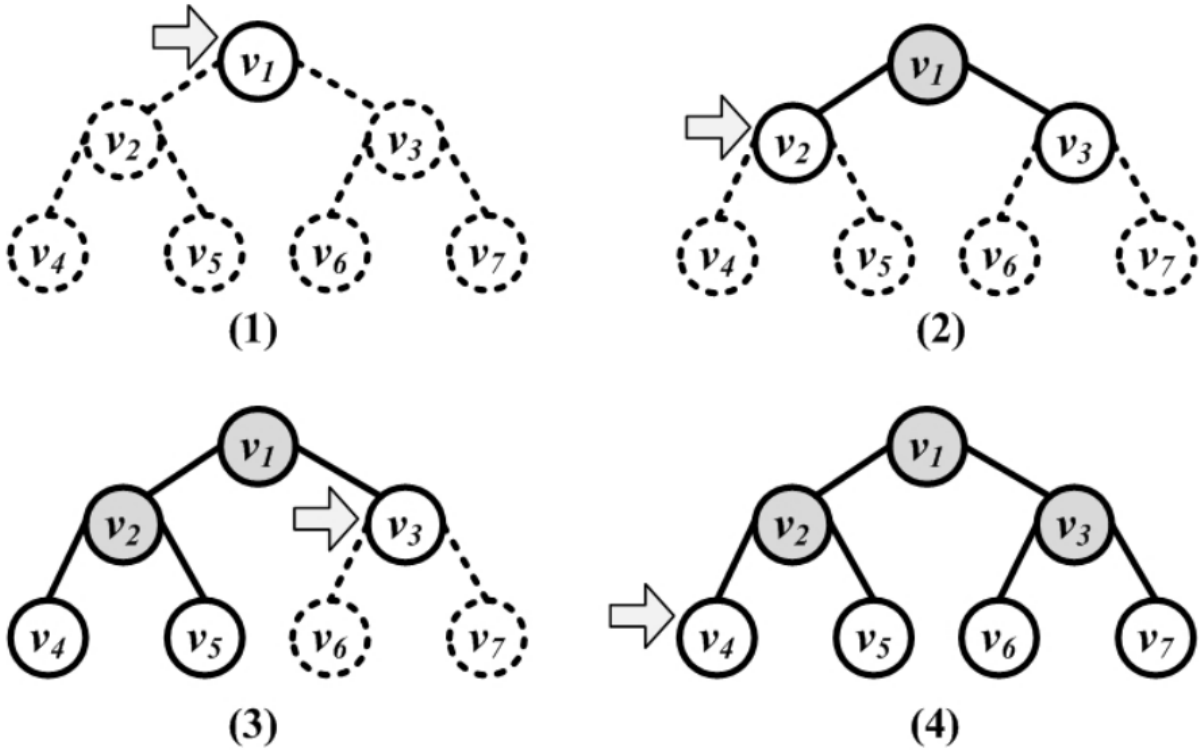


Image: Zafarani et al., Social Media Mining, 2014.

Graph traversal: BFS

- **Breadth First Search** (BFS)
- From source node, create a rooted spanning tree of the graph
- Graph traversal in level-order
- Often implemented using a queue
- BFS considers traversing each of the m edges once, so $O(m)$
- Important for computing various centrality measures

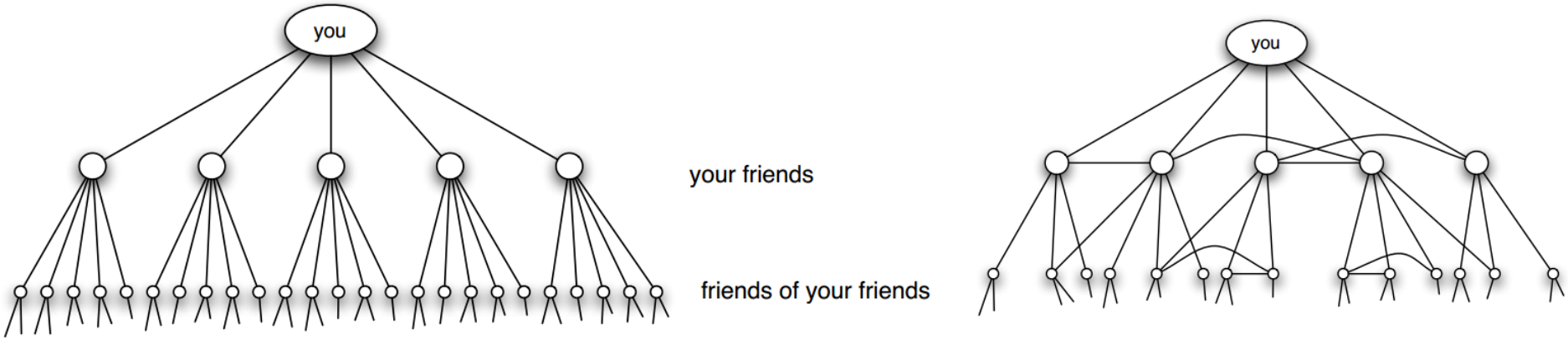


Image: Zafarani et al., Social Media Mining, 2014.

Universiteit Leiden

Centrality

Centrality

Universiteit Leiden

- Given a social network, which person is most important?
- What is the most important page on the web?
- Which protein is most vital in a biological network?
- Who is the most respected author in a scientific citation network?
- What is the most crucial router in an internet topology network?

Centrality

- **Node centrality**: the importance of a node with respect to the other nodes based on the structure of the network
- **Centrality measure**: computes the centrality value of all nodes in the graph
- For all $v \in V$ a measure M returns a value $C_M(v) \in [0; 1]$
- $C_M(v) > C_M(w)$ means that node v is more important than w


GAME OF THRONES
Watermark

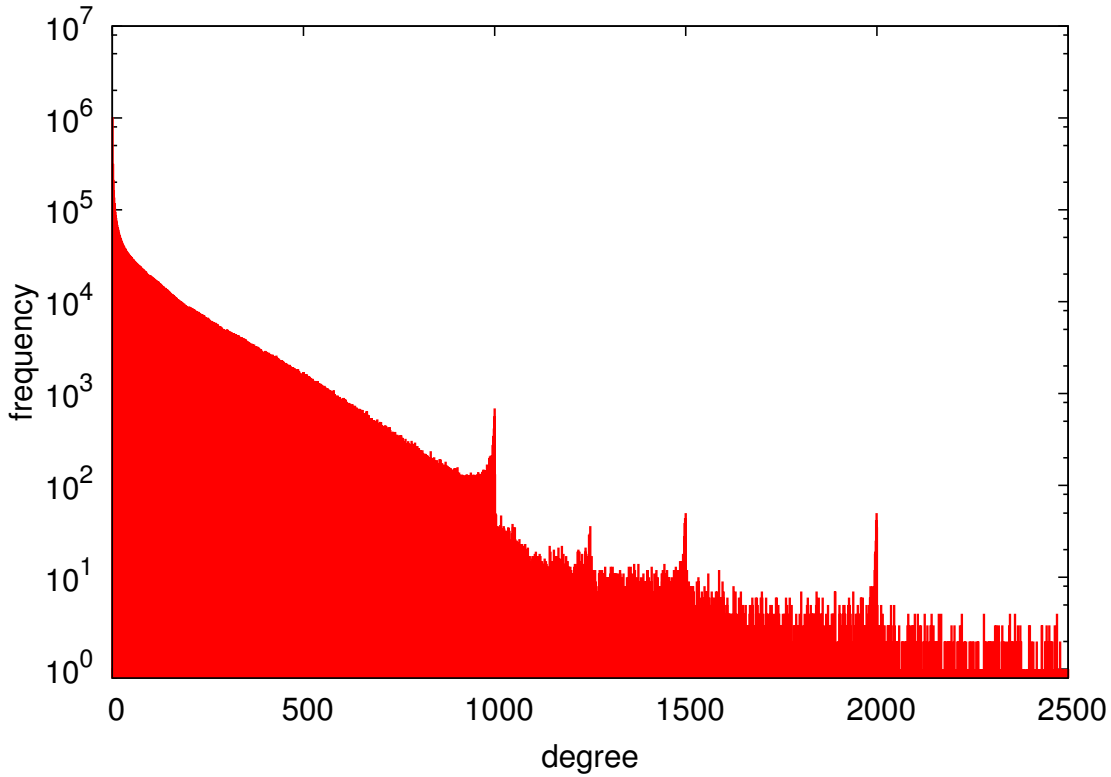
Degree centrality

- Undirected graphs – **degree centrality**: measure the number of adjacent nodes

$$C_d(v) = \frac{deg(v)}{n-1}$$

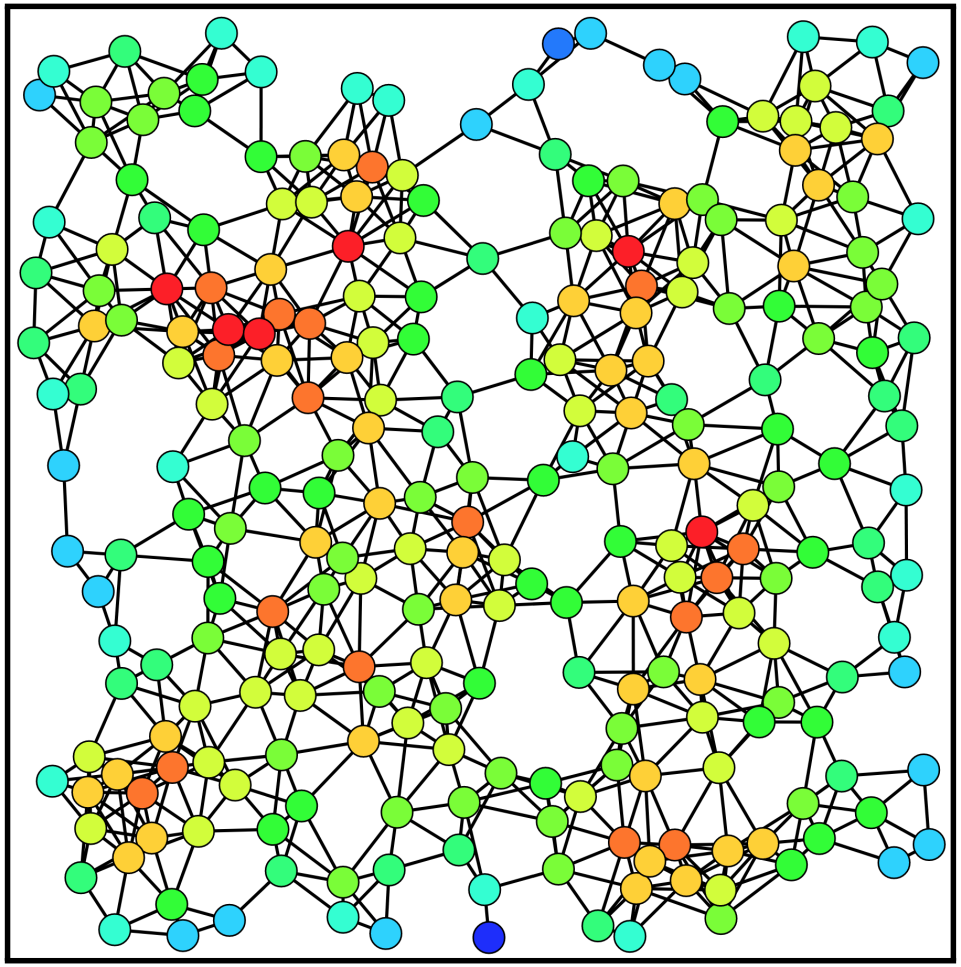
- Directed graphs — indegree centrality and outdegree centrality
- Local measure
- $O(1)$ time to compute

Degree distribution

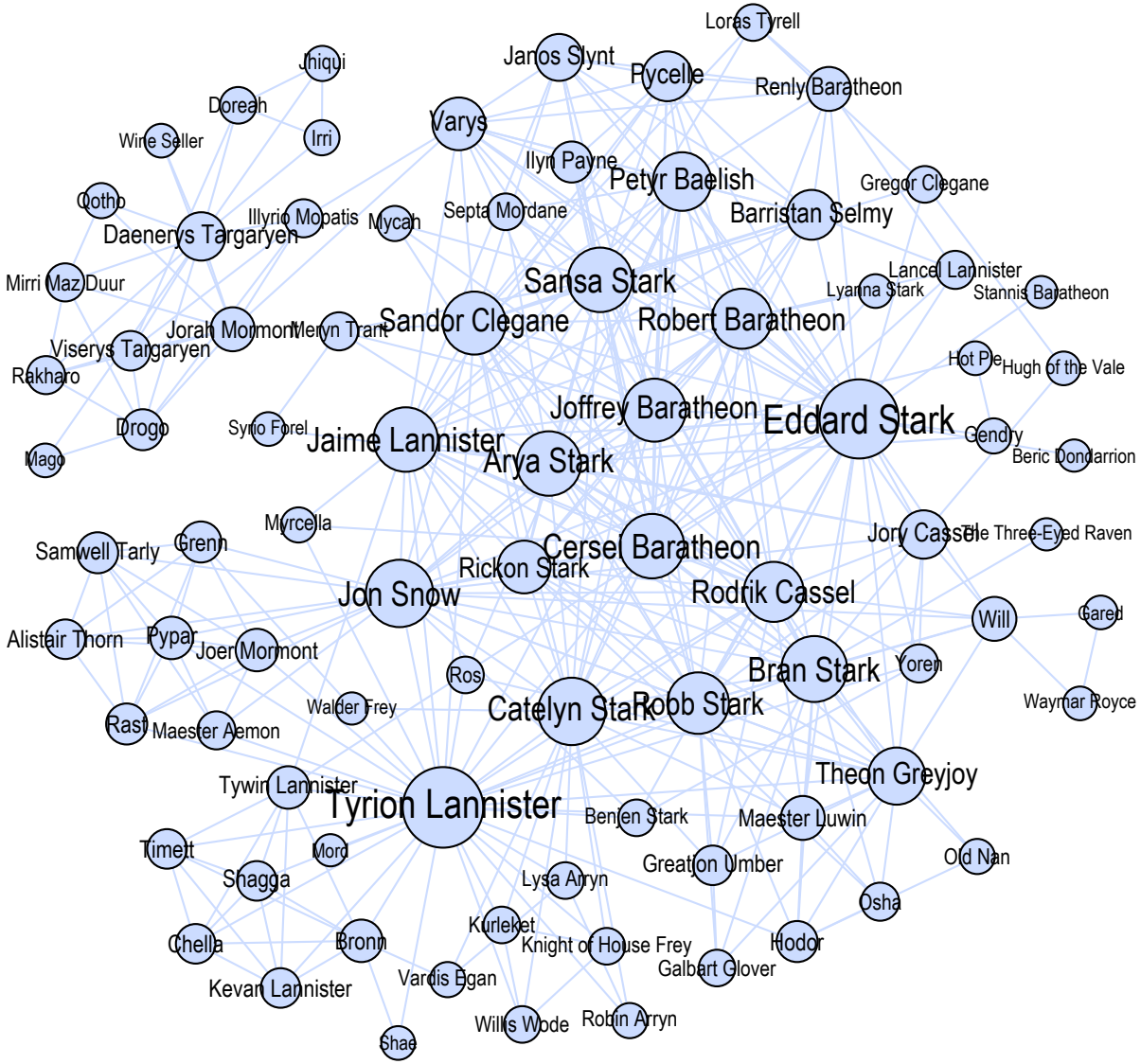


- Not so many distinct values in the lower ranges

Degree centrality



Degree centrality



Closeness centrality

- **Closeness centrality**: based on the average distance to all other nodes

$$C_c(v) = \left(\frac{1}{n-1} \sum_{w \in V} d(v, w) \right)^{-1}$$

- Global distance-based measure
- $O(mn)$ to compute: one BFS in $O(m)$ for each of the n nodes

Closeness centrality

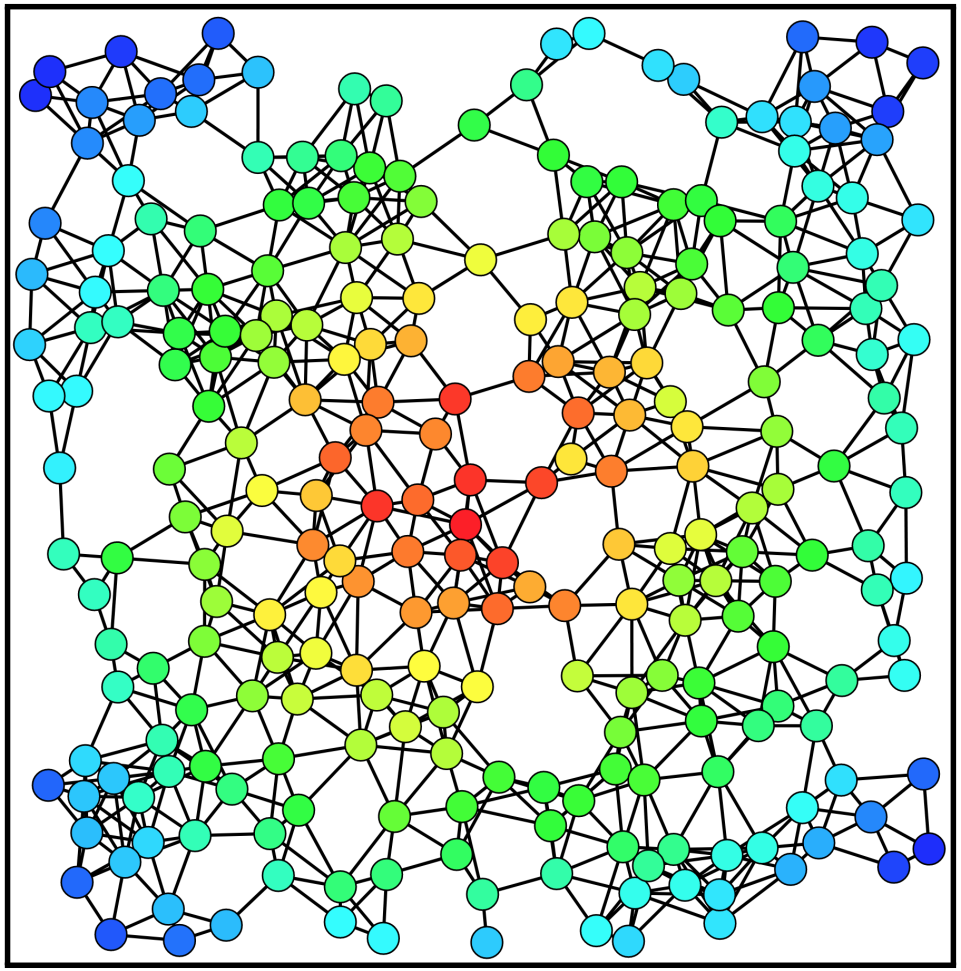
- **Closeness centrality**: based on the average distance to all other nodes

$$C_c(v) = \left(\frac{1}{n-1} \sum_{w \in V} d(v, w) \right)^{-1}$$

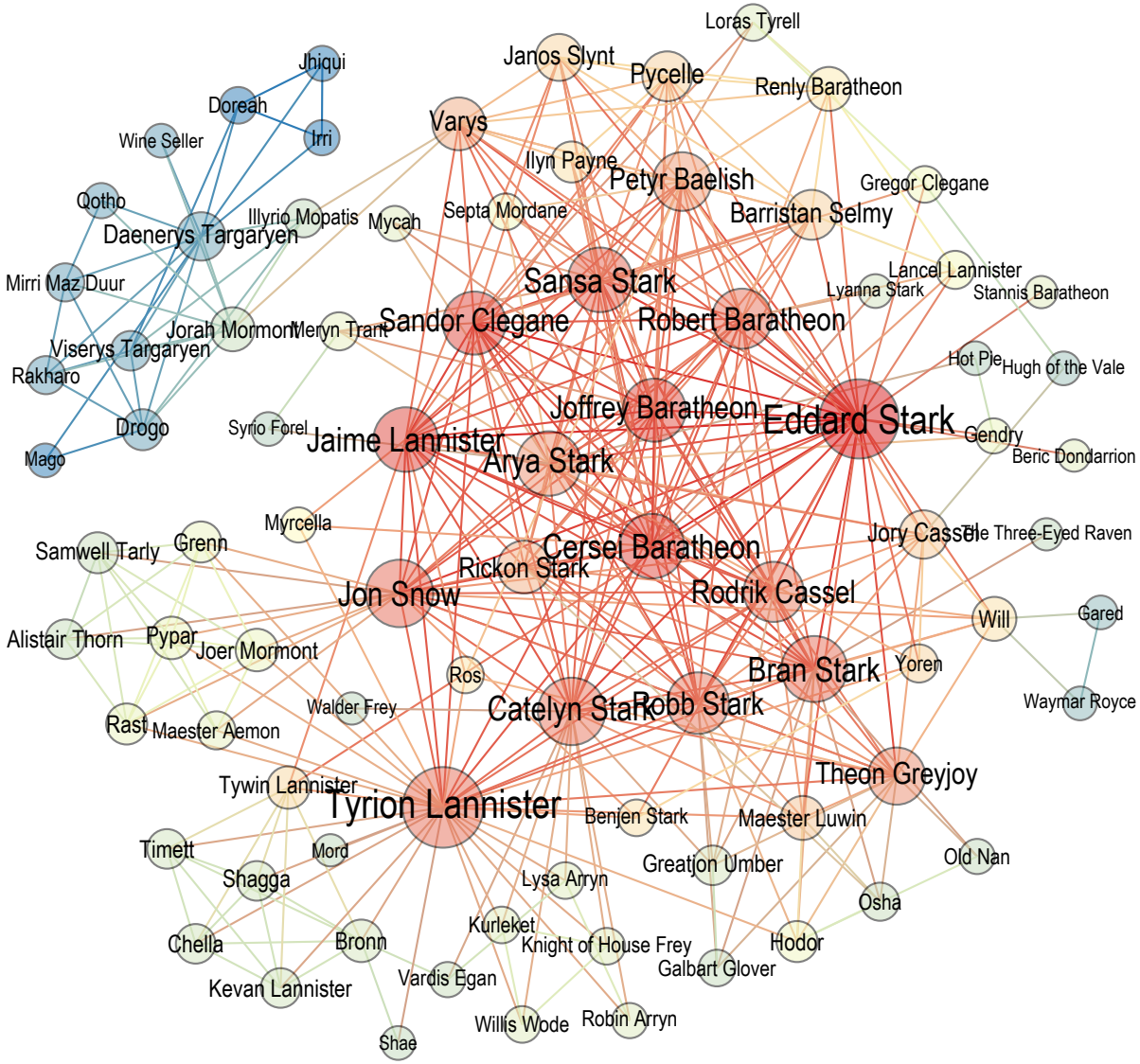
- Global distance-based measure
- $O(mn)$ to compute: one BFS in $O(m)$ for each of the n nodes
- Connected component(s)...
- **Harmonic centrality**: variant of closeness (not normalized)

$$C_h(v) = \sum_{w \in V} \frac{1}{d(w, v)}$$

Closeness centrality



Degree vs. closeness centrality



Betweenness centrality

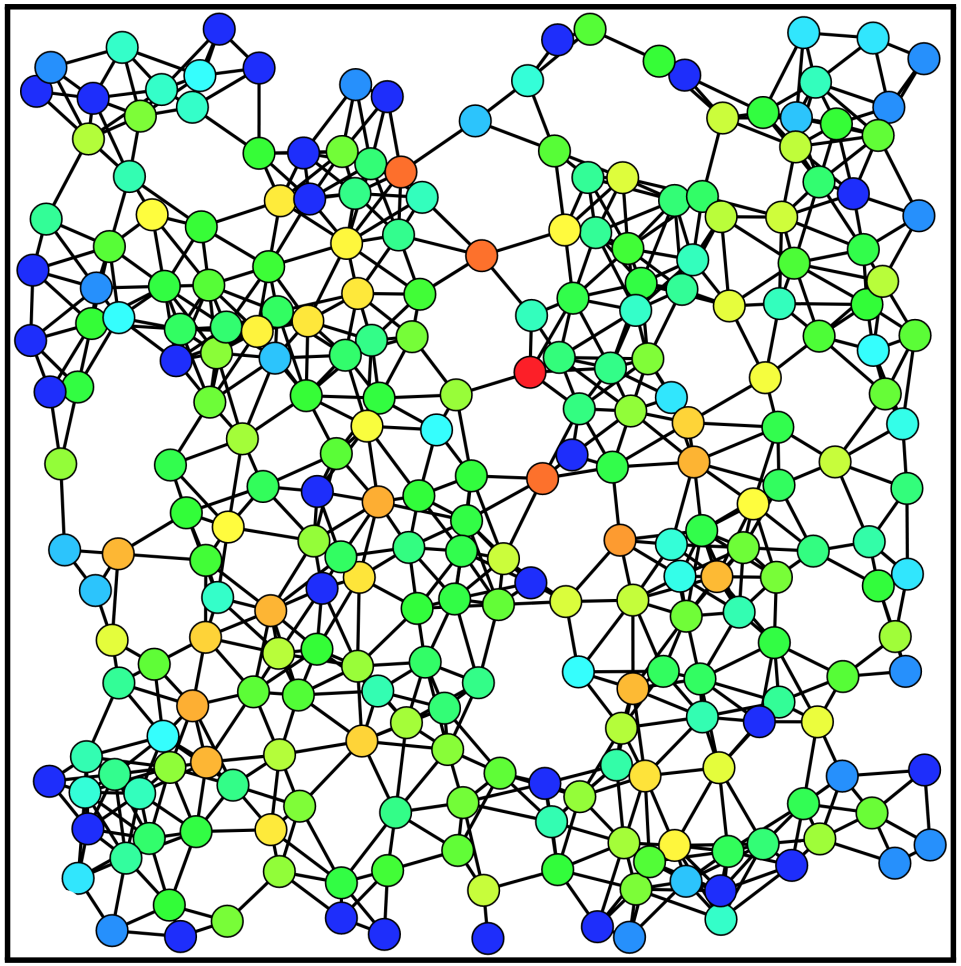
- **Betweenness centrality**: measure the number of shortest paths that run through a node

$$C_b(u) = \sum_{\substack{v,w \in V \\ v \neq w, u \neq v, u \neq w}} \frac{\sigma_u(v,w)}{\sigma(v,w)}$$

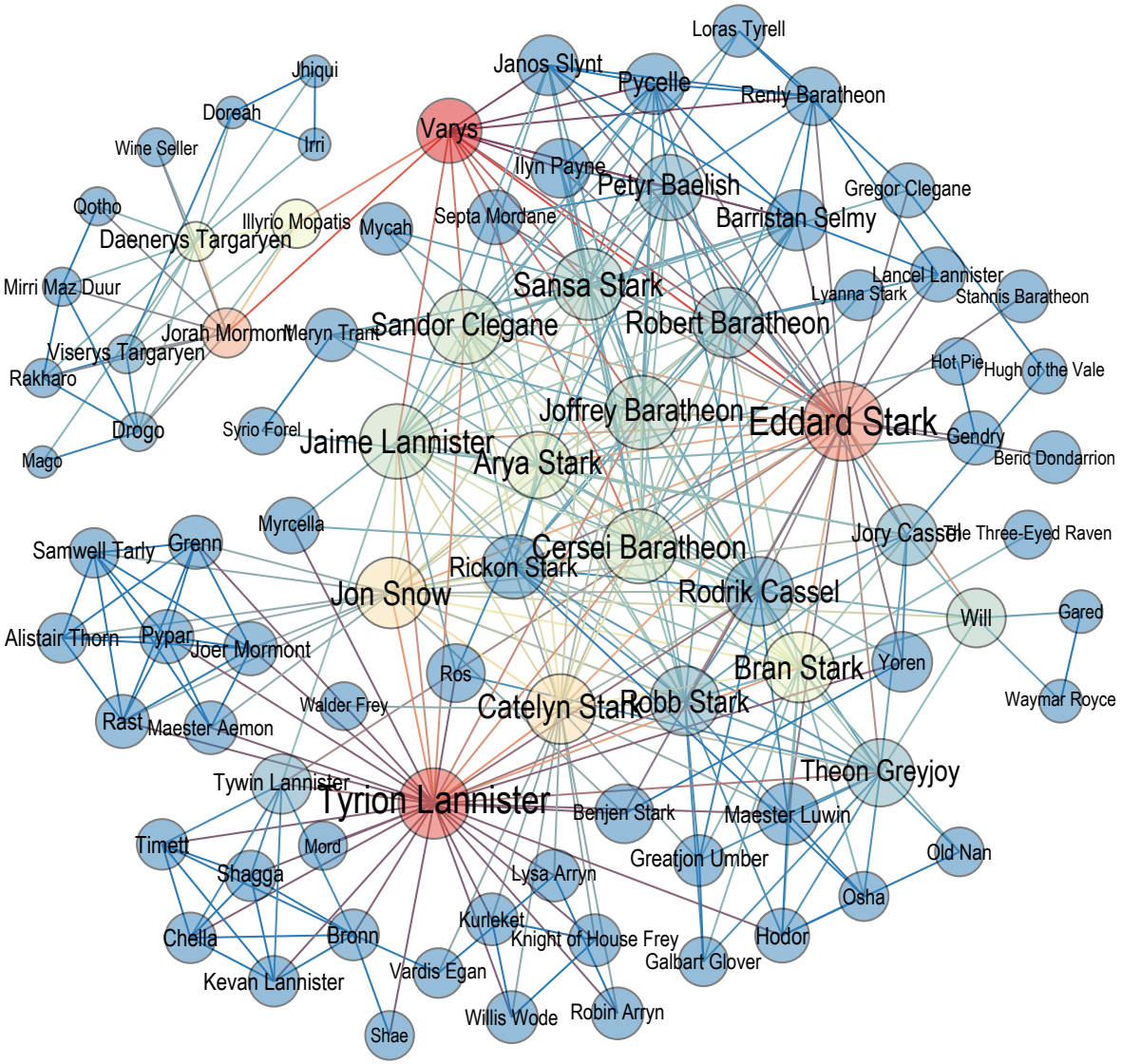
- $\sigma(v,w)$ is the number of shortest paths from v to w
- $\sigma_u(v,w)$ is the number of such shortest paths that run through u
- Divide by largest value to normalize to $[0;1]$
- Global path-based measure
- $O(2mn)$ time to compute (two "BFSes" for each node)

U. Brandes, "A faster algorithm for betweenness centrality", Journal of Mathematical Sociology 25(2): 163–177, 2001

Betweenness centrality



Degree vs. betweeness centrality



Centrality measures compared

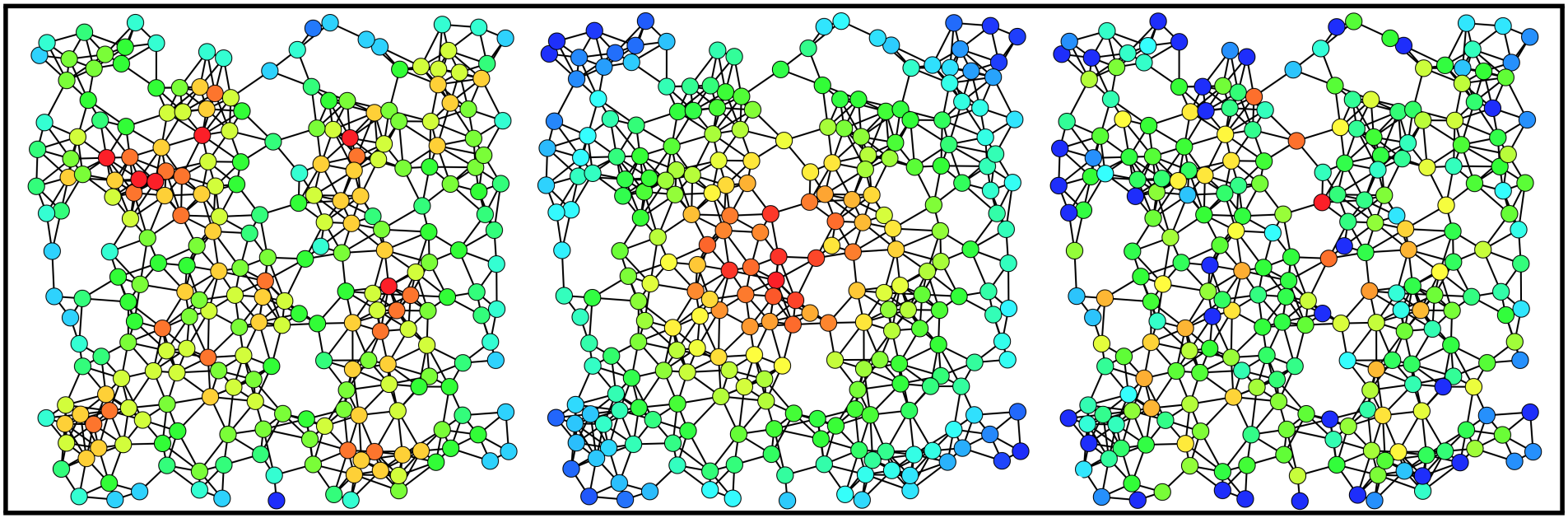


Figure: Degree, closeness and betweenness centrality

Source: "Centrality"' by Claudio Rocchini, Wikipedia `File:Centrality.svg`

Eccentricity centrality

- Node **eccentricity**: length of a longest shortest path (distance to a node furthest away)

$$e(v) = \max_{w \in V} d(v, w)$$

- **Eccentricity centrality**:

$$C_e(v) = \frac{1}{e(v)}$$

- Worst-case variant of closeness centrality
- $O(mn)$ to compute: one BFS in $O(m)$ for each of the n nodes

Eccentricity centrality

- Node **eccentricity**: length of a longest shortest path (distance to a node furthest away)

$$e(v) = \max_{w \in V} d(v, w)$$

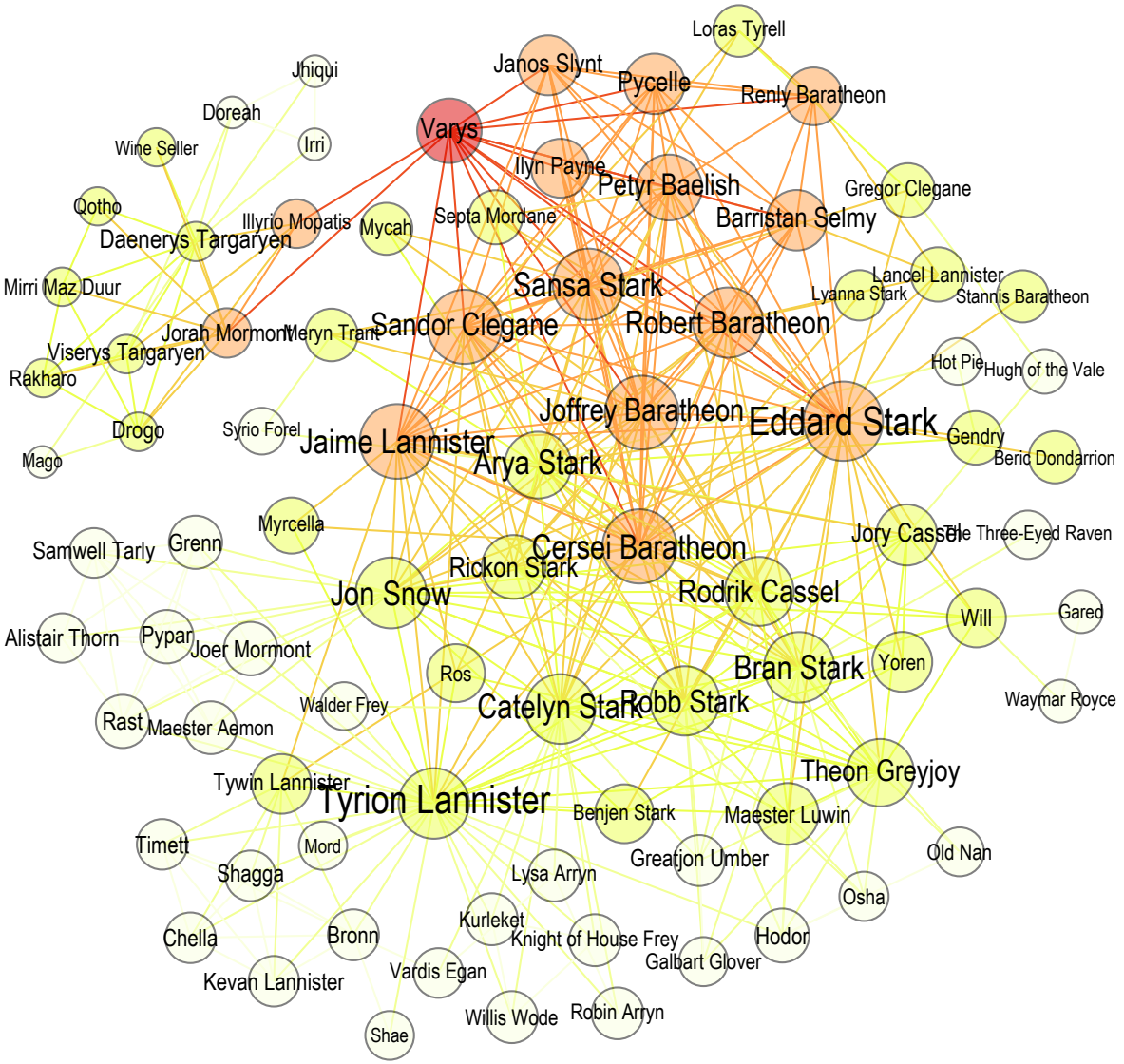
- **Eccentricity centrality**:

$$C_e(v) = \frac{1}{e(v)}$$

- Worst-case variant of closeness centrality
- $O(mn)$ to compute: one BFS in $O(m)$ for each of the n nodes
  - Large optimizations possible using lower and upper bounds, see F.W. Takes and W.A. Kosters, Computing the Eccentricity Distribution of Large Graphs, *Algorithms*, vol. 6, nr. 1, pp. 100-118, 2013.

Eccentricity centrality

Degree vs. eccentricity centrality



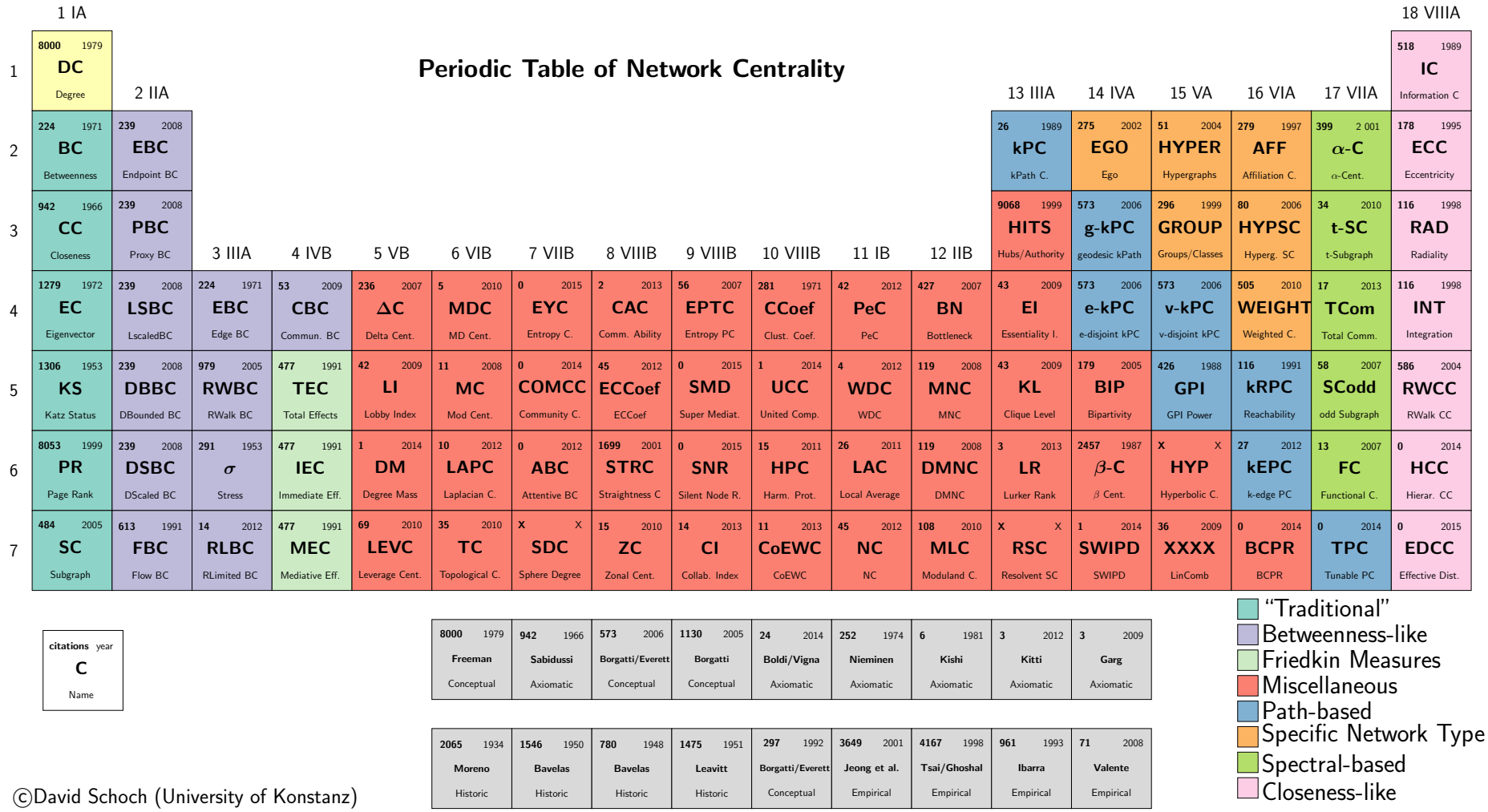
Centrality measures

- Distance/path-based measures:
  - Degree centrality $O(n)$
  - Closeness centrality $O(mn)$
  - Betweenness centrality $O(mn)$
  - Eccentricity centrality $O(mn)$

  (complexity is for computing centralities of all n nodes)
- Many more: Eigenvector centrality, Katz centrality, ...
- Approximating these measures is also possible
- Also: propagation-based centrality measures like PageRank

Periodic table of centrality

Universiteit Leiden



Universiteit Leiden

Course project

Course project

- Project on specific SNA subtopic, 60% of your course grade
- Teams of exactly 2 students
- Deliverables:
  - **Presentation** on a topic-related paper. $\leq$ 30 minutes for your talk, $\approx$15 minutes for questions and discussion
  - **Paper** with a **contribution** to SNA that goes beyond what is done in the paper you study, e.g.:
    - Comparing similar algorithms from different papers
    - Testing the algorithm(s) on larger datasets
    - Validating algorithms using different metrics
    - Addressing future work posed in the paper
    - Replicate the study using more extensive parameter testing
  - Short peer review document (during peer review session)
  - Relevant project code and supplementary material
  - Bonus for open-source or open science contributions
- Topics divided over teams based on first come, first serve

Presentation

- Present
  - **the assigned paper** on the topic of your project
  - **your contribution** (what you present here depends on progress)
- Convey the main message of the paper in an understandable way
- Show a nice demo, pictures, movies or visualization
- Have a clearly structured presentation
- Briefly discuss your project plans
- Demo presentation will follow
- Discussion with (and engagement of) other students is expected (from both presenters and attendees)

Course project

- Read your paper, understand the main problem
- Do a bit of research on related literature
- Determine which algorithms/techniques/parameters/datasets/ subproblems you are going to compare (i.e., your contribution)
- Program (or obtain code of) the different algorithms and techniques
- Obtain and describe applicable datasets for comparing the algorithms
- Perform and report on experiments to compare the algorithms
- Determine and discuss results
- Write a sensible conclusion

Course project paper

- Scientific paper
- LaTeX
- 6 to 10 pages, two columns
- Images, figures, graphs, diagrams, tables, references, ...
- Between 5 and 9 sections
- Peer review and code review
- Option for "intermediary paper check" before final hand-in

Common pitfalls/excuses

Universiteit Leiden

- Only starting to read your paper after Assignment 2 (you are late)
- Starting just before the first paper deadline (you are very late)
- Starting writing only just before first review (your paper will likely be too meager)
- Starting writing code only just before (your algorithms will be slow, you do not have enough time to run your experiments, you claim it is "because everyone is using the servers")
- Copying from the internet
- A presentation without pictures
- Literally reading out every sentence on your slides
- Too much text on your slides
- Not writing the paper in LaTeX

Course project schedule

- Sep 30: deadline for registering as a group in Brightspace for the course project topic
- Oct 1: remaining students/topics matched
- From mid October onwards: presentations by students
- Nov. 11: deadline for the first three sections (for peer review)
- Nov. 22: optional deadline for "intermediary paper check"
- Nov. 26: deadline for having decent code ready for code review
- Dec. 13: deadline for full course project paper
- Dec. 23: all grades submitted to student administration

Homework for next week

- Make serious progress with Assignment 1
- Make choice of participation in course explicit. Un-enroll no later than September 22; anyone registered after that date will get a grade
- Next week: consult the list of project topics on course website, and think of what you may want to work on
- Ask questions
- **Today:** stick around if you are already certain that you will take the course, and want to find a teammate already

Upcoming lab session

- From 9:15 to 11:00 in Snellius rooms 302/304 etc.
- Instructions on course website
- Hands-on introduction to `NetworkX`
- Continue working on Assignment 1