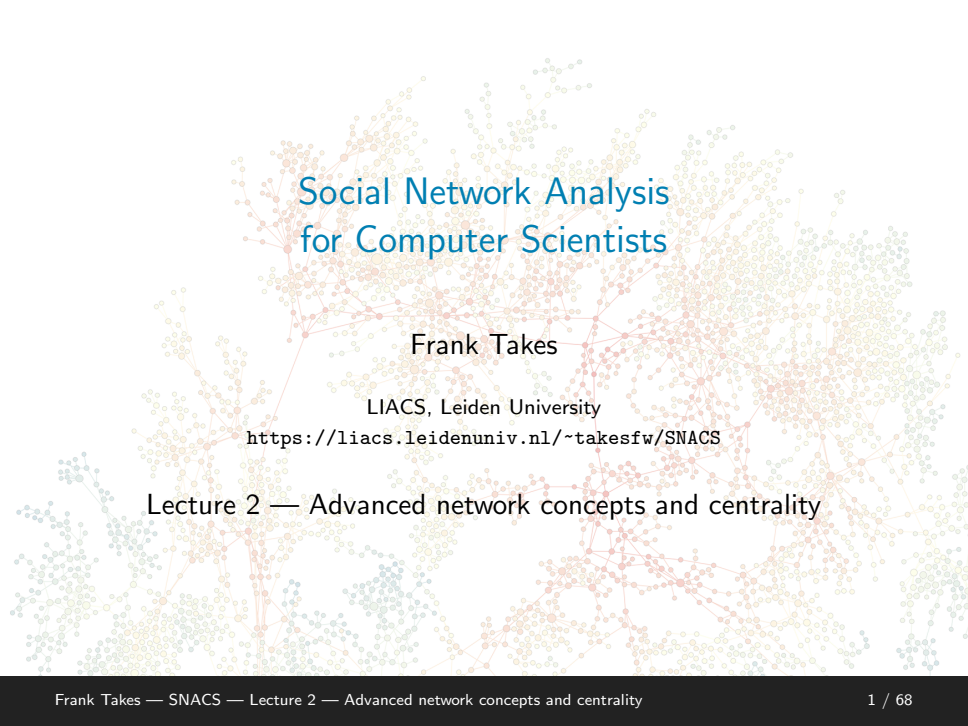




Social Network Analysis for Computer Scientists

Frank Takes

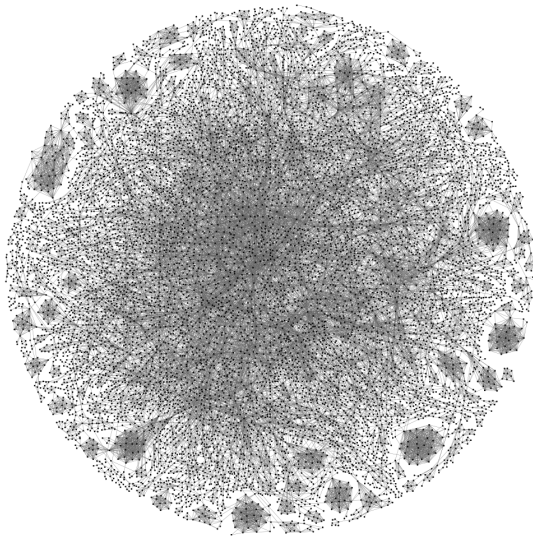
LIACS, Leiden University

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

Lecture 2 — Advanced network concepts and centrality

Recap

Networks



Notation

Concept

- Network (graph)
- Nodes (objects, vertices, ...)
- Links (ties, relationships, ...)
 - Directed — $E \subseteq V \times V$ — "links"
 - Undirected — "edges"
- Number of nodes — $|V|$
- Number of edges — $|E|$
- Degree of node u
- Distance from node u to v

Symbol

$G = (V, E)$

V

E

n

m

$\deg(u)$

$d(u, v)$

Real-world networks

- 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 (Erdős, Kevin Bacon), protein interaction networks, information networks (Wikipedia), webgraphs, financial networks (Bitcoin) ...

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

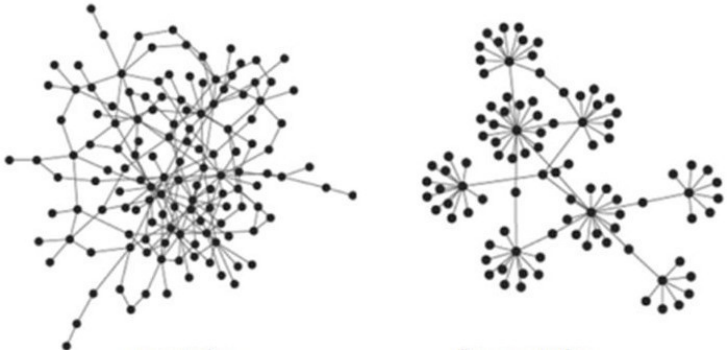


Figure: Degree **assortativity** (left) and degree **disassortativity** (right)

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

Attribute assortativity

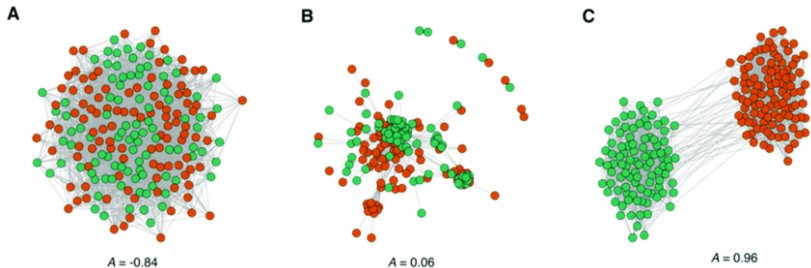


Figure: Attribute assortativity

Image: Moya-García, A. et al. Identification of New Toxicity Mechanisms ... *Genes*, 13(7), 1292, 2022.

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:

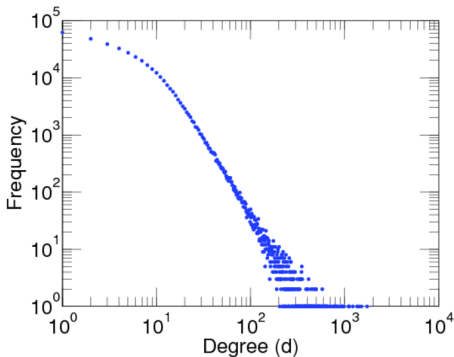
$$m_{<->} = |\{(u, v) \in E : (v, u) \in E\}|$$

- 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 :

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

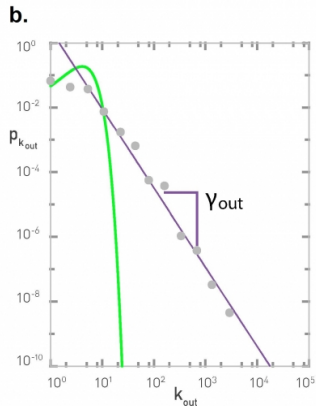
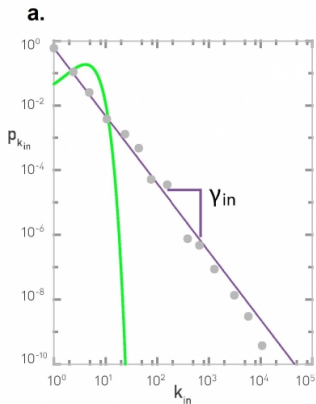
- This means that

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

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

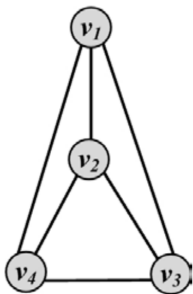
Power law exponent in directed networks



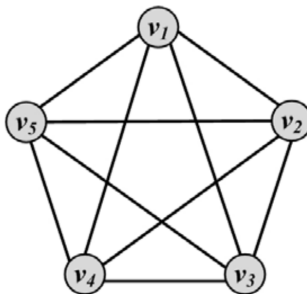
Source: A. Barabasi, Network Science, 2016.

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 $\frac{1}{2} \cdot n \cdot (n - 1)$

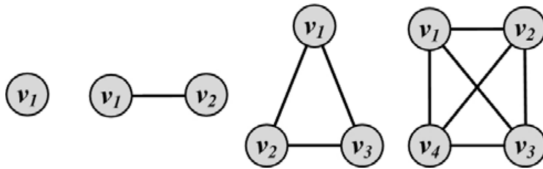


Figure: Complete graphs of size 1, 2, 3 and 4

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

Ego network

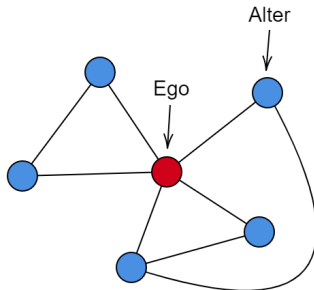


Figure: The **ego network** of a given node in a network consists of the set of nodes containing that node (“Ego”) and its direct neighbors (“Alters”), and all edges present between the nodes in this set

Image: Wikipedia “Egocentric network.png”, accessed 2022.

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

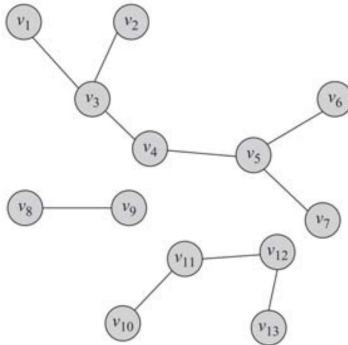


Image: Zafarani et al., Social Media Mining, 2014.

Trees

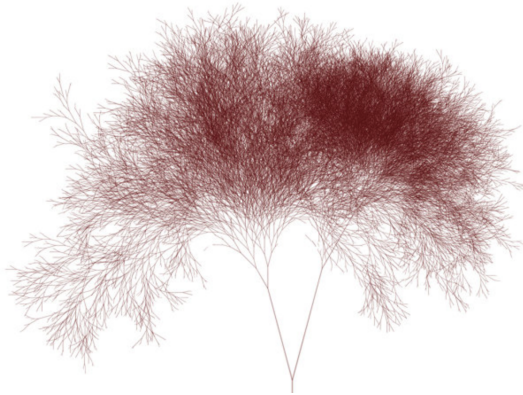


Image: M. Lima, Book of trees: Visualizing branches of knowledge, 2014.

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

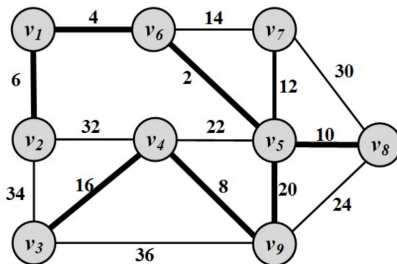


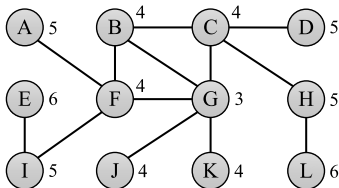
Image: Zafarani et al., Social Media Mining, 2014.

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

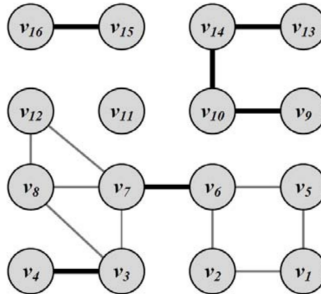


Image: Zafarani et al., Social Media Mining, 2014.

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)

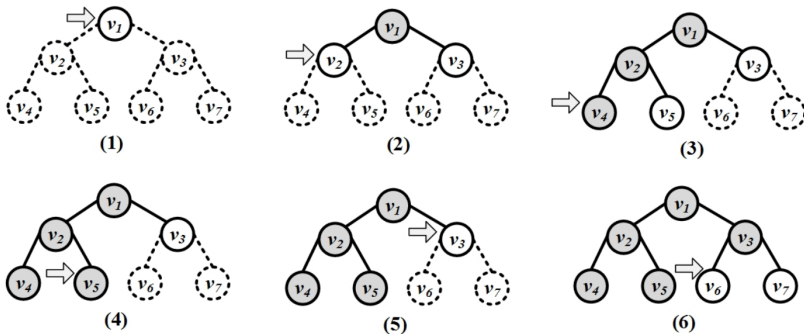
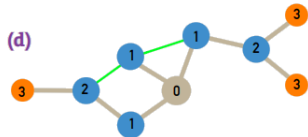
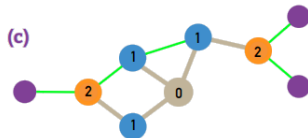
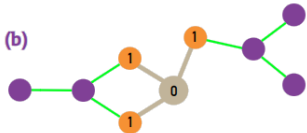
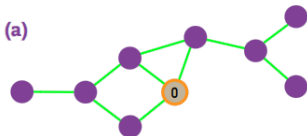


Image: Zafarani et al., Social Media Mining, 2014.

Graph traversal: BFS



Source: A. Barabasi, Network Science, 2016.

Graph traversal: BFS

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

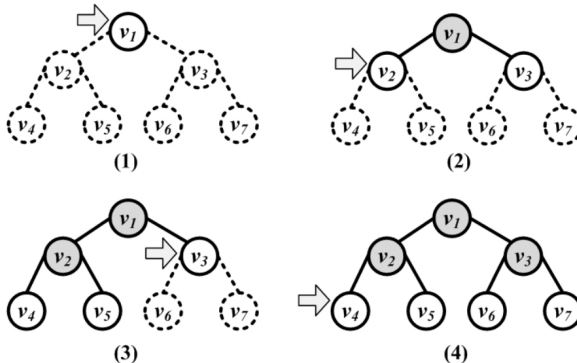


Image: Zafarani et al., Social Media Mining, 2014.

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

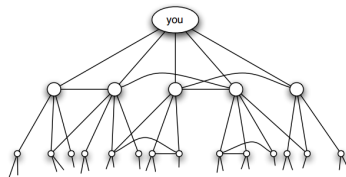
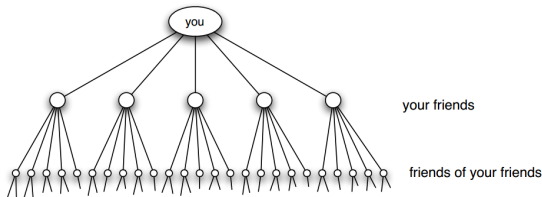


Image: Zafarani et al., Social Media Mining, 2014.

Centrality

Centrality

- 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



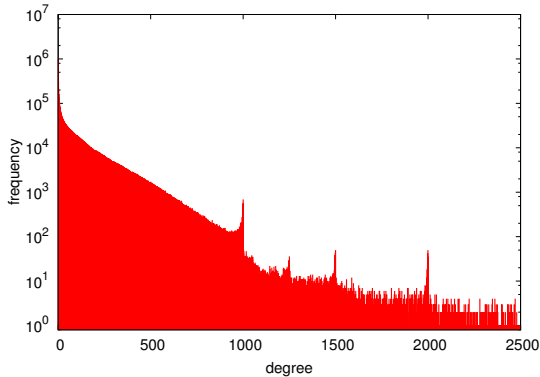
Degree centrality

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

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

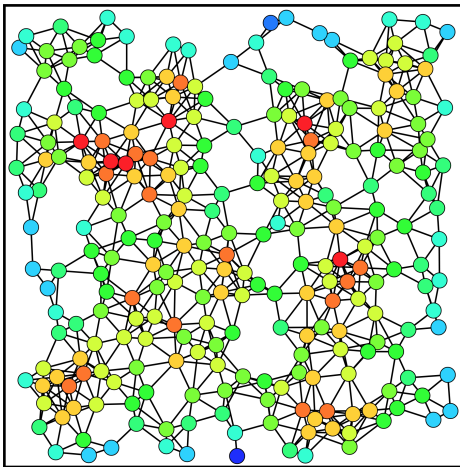
- Directed graphs — indegree centrality and outdegree centrality
- Local measure
- $O(1)$ time to compute

Degree distribution

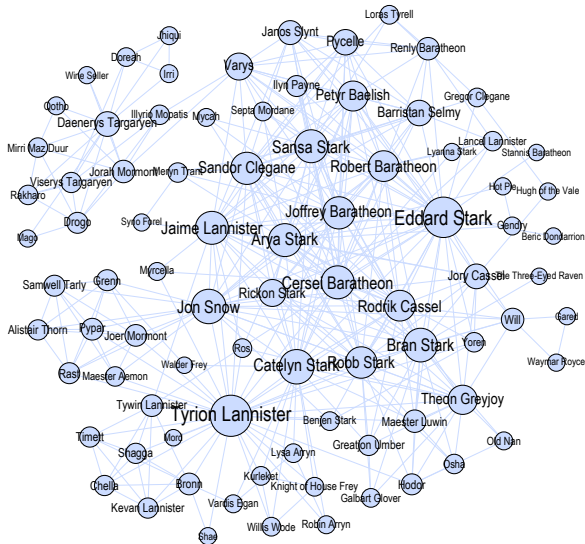


- Not so many distinct values in the lower ranges

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

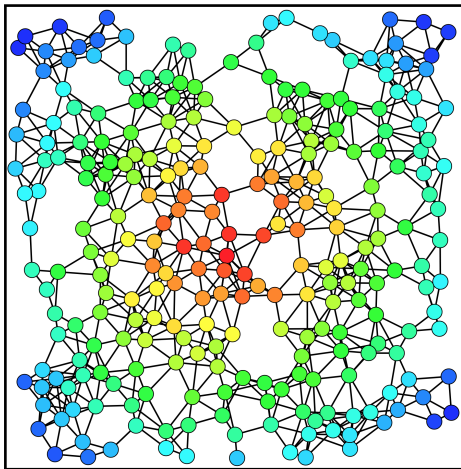
$$C_c(v) = \left(\frac{1}{n-1} \sum_{w \in V} d(v, w) \right)^{-1}$$

- 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)

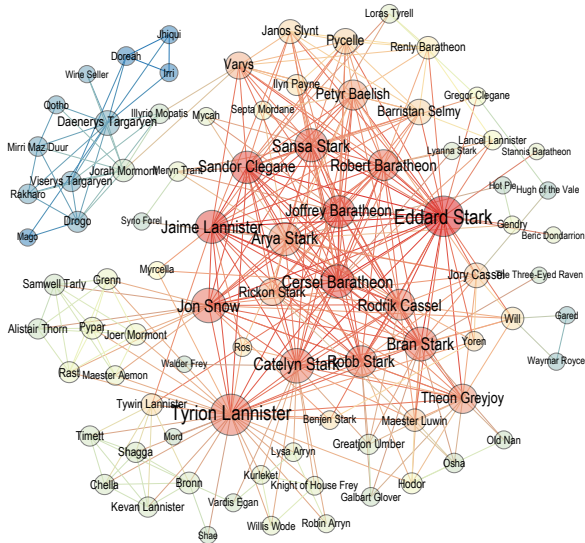
$$C_h(v) = \sum_{w \in V} \frac{1}{d(w, v)}$$

Sabidussi, G., The centrality index of a graph, *Psychometrika* 31(4): 581–603, 1966.

Closeness centrality



Degree vs. closeness centrality



Betweenness centrality

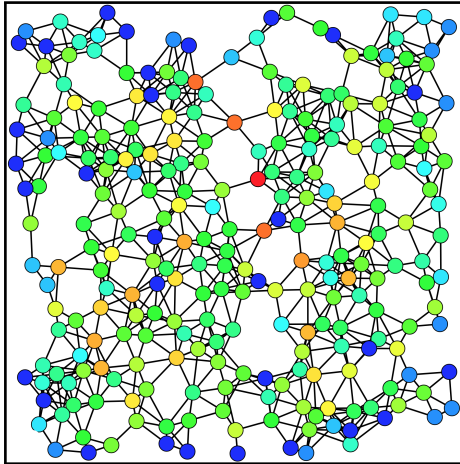
- **Betweenness centrality:** measure the number of shortest paths that run through a node

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

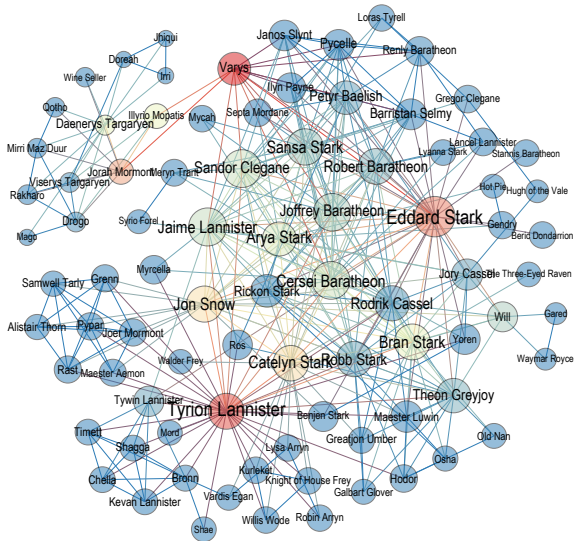
- $\sigma(v, w)$ is the number of shortest paths from v to w
- $\sigma_u(v, w)$ is the number of such shortest paths that run through u
- Divide by largest value to normalize to $[0; 1]$
- Global path-based measure
- $O(2mn)$ time to compute (two “BFSes” for each node)

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

Betweenness centrality



Degree vs. betweenness centrality



Centrality measures compared

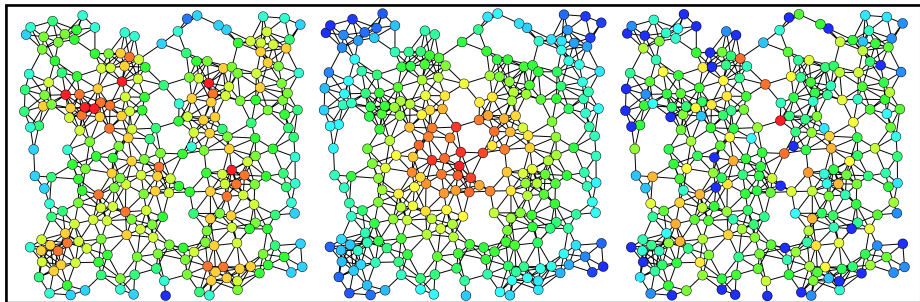


Figure: Degree, closeness and betweenness centrality

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)

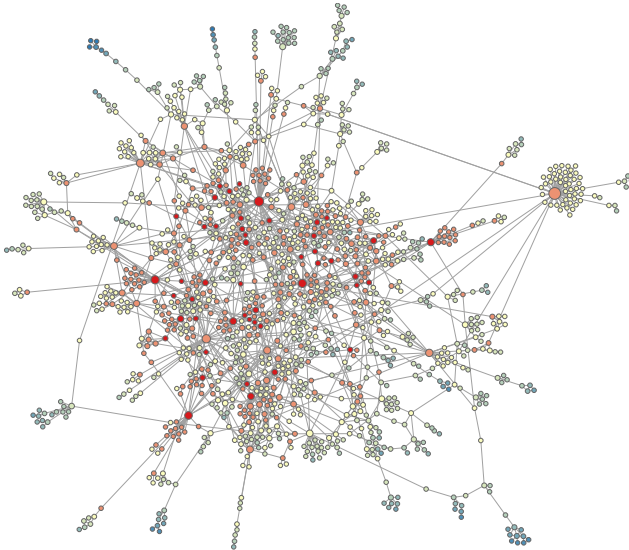
$$e(v) = \max_{w \in V} d(v, w)$$

- **Eccentricity centrality**:

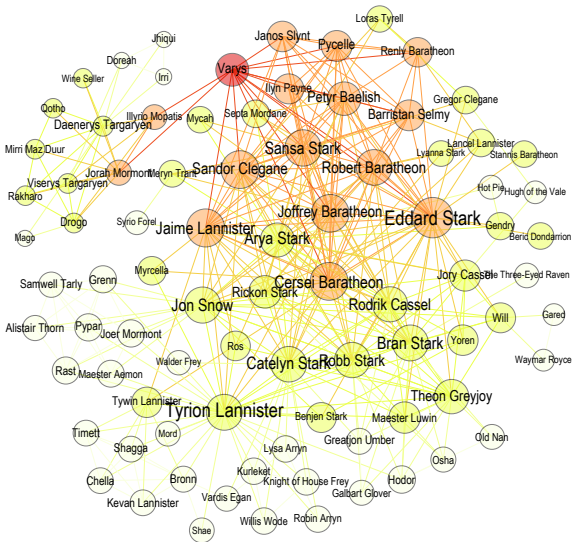
$$C_e(v) = \frac{1}{e(v)}$$

- 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. Kusters, Computing the Eccentricity Distribution of Large Graphs, *Algorithms*, vol. 6, nr. 1, pp. 100-118, 2013.

Eccentricity centrality



Degree vs. eccentricity centrality



Computing betweenness centrality

Counting shortest paths

- **Bellman criterion:** v lies on a shortest path from u to w if $d(u, v) + d(v, w) = d(u, w)$.
- Predecessors: $P_u(w)$ is the set of predecessors of node w on a shortest path from u , formally:

$$P_u(w) = \{v \in V : (v, w) \in E, d(u, w) = d(u, v) + 1\}$$

- Counting the number of shortest paths $\sigma(u, w)$ for $u \neq w$:

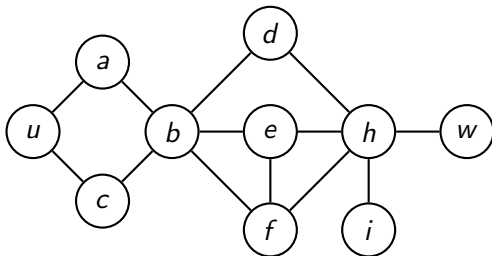
$$\sigma(u, w) = \sum_{v \in P_u(w)} \sigma(u, v)$$

$$\text{with } \sigma(u, u) = 1$$

Counting shortest paths

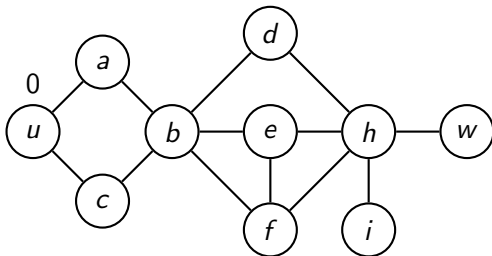
Task: compute $\sigma(u, w)$, the number of shortest paths from u to w

...



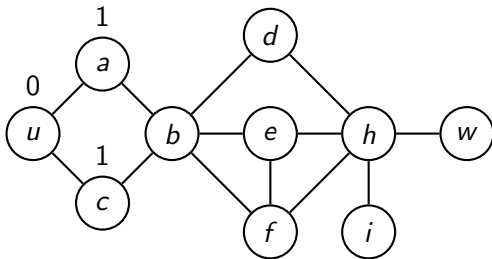
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
First a BFS from u to find w



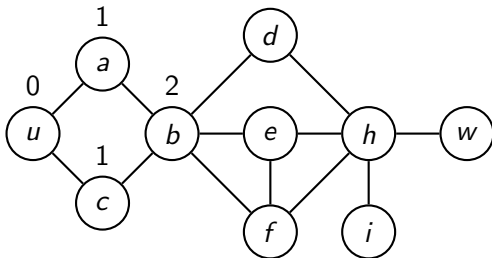
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
First a BFS from u to find w



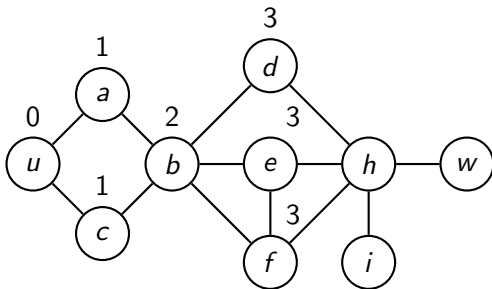
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
First a BFS from u to find w



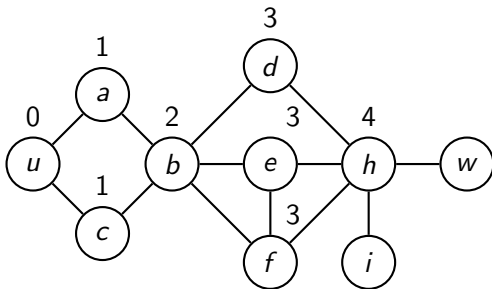
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
First a BFS from u to find w



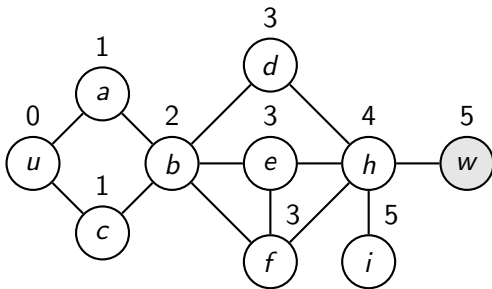
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
First a BFS from u to find w



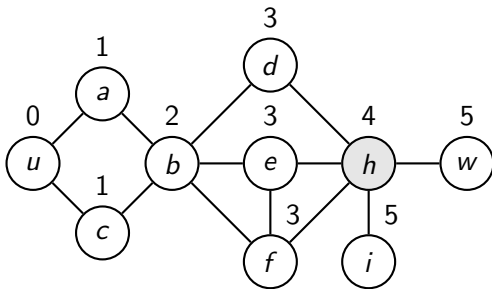
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
 $\sigma(u, w)$? Ask predecessors $P_u(w) = \{h\}$ for its value



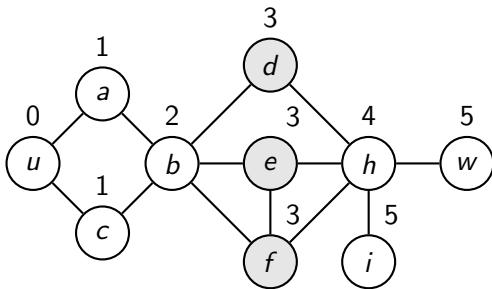
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
 $\sigma(u, h)$? Ask predecessors $P_u(h) = \{d, e, f\}$ for its value



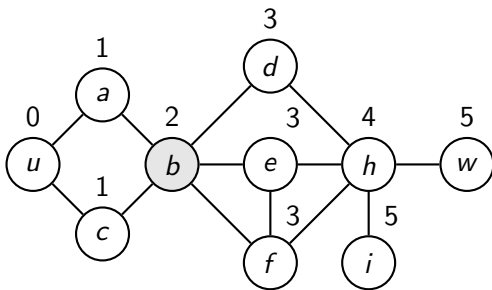
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
Ask predecessors ...



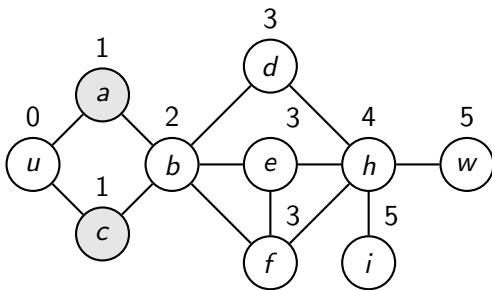
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
Ask predecessors ...



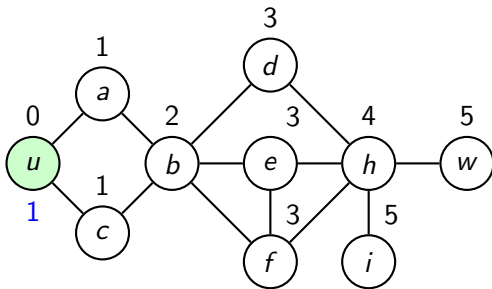
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
Ask predecessors ...



Counting shortest paths

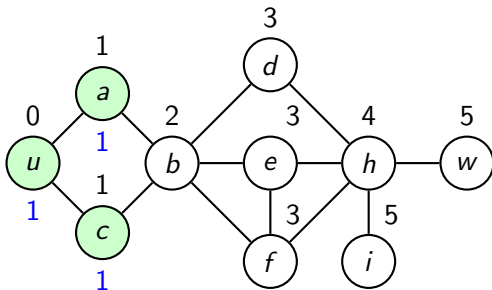
Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
 $\sigma(u, u) = 1$, now propagate back...



Counting shortest paths

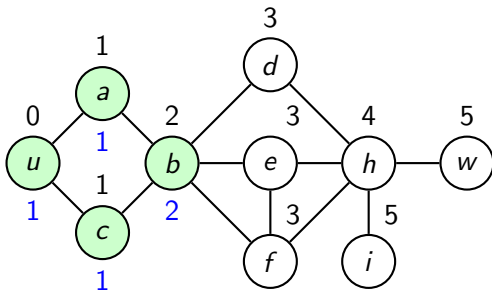
Task: compute $\sigma(u, w)$, the number of shortest paths from u to w

$$\sigma(u, a) = \sigma(u, c) = 1$$



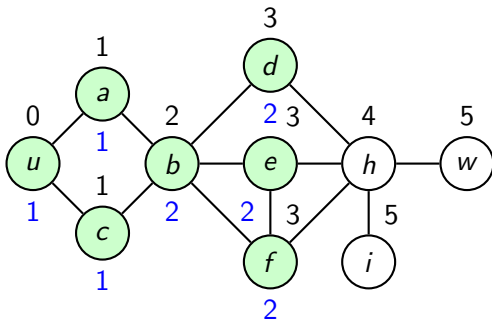
Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
 $\sigma(u, b) = \sigma(u, a) + \sigma(u, c) = 2$



Counting shortest paths

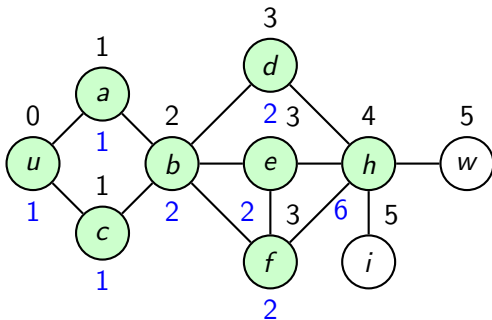
Task: compute $\sigma(u, w)$, the number of shortest paths from u to w
 $\sigma(u, d) = \sigma(u, e) = \sigma(u, f) = \sigma(u, c) = 2$



Counting shortest paths

Task: compute $\sigma(u, w)$, the number of shortest paths from u to w

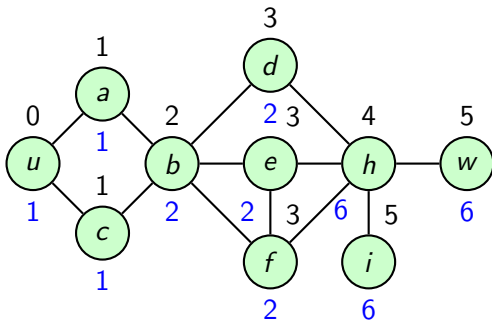
$$\sigma(u, h) = \sigma(u, d) + \sigma(u, e) + \sigma(u, f) = 6$$



Counting shortest paths

Done!

$$\sigma(u, w) = \sigma(u, h) = 6$$



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

Periodic Table of Network Centrality																	18 VIIIA						
1 IA																	18 VIIIA						
1	8000 1979 DC Degree	2 IIA															518 1989 IC Information C.						
2	234 1971 BC Betweenness	239 2008 EBC Endpoint BC																26 1989 kPC kPath C.	275 2002 EGO Ego	51 2004 HYPER Hypergraphs	279 1997 AFF Affiliation C.	399 2 001 α -C α -Cent.	178 1995 ECC Eccentricity
3	942 1966 CC Closeness	239 2008 PBC Proxy BC																9068 1999 HITS Hubs/Authority	573 2006 g-kPC geodesic kPath	296 1999 GROUP Groups/Classes	80 2006 HYPSC Hypers. SC	34 2010 t-SC t-Subgraph	116 1998 RAD Radiality
4	1279 1972 EC Eigenvector	239 2008 LSBC Localised BC	224 1971 EBC Edge BC	53 2009 CBC Commun. BC	236 2007 Δ C Delta Cent.	5 2010 MDC MD Cent.	8 2015 EYC Entropy C.	2 2013 CAC Comm. Ability	56 2007 EPTC Entropy PC	281 1971 CCoef Clust. Coef.	42 2012 PeC PeC	427 2007 BN Bottleneck	43 2009 EI Essentiality I.	573 2006 e-kPC e-djacent kPC	573 2006 v-kPC v-djacent kPC	505 2010 WEIGHT Weighted T.	17 2013 TCom Total Comm.	116 1998 INT Integration					
5	1306 1983 KS Katz Status	239 2008 DBBC DBounded BC	979 2005 RWBC RWalk BC	477 1991 TEC Total Effects	42 2009 LI Lobby Index	11 2008 MC Mod. Cent.	8 2014 COMCC Community C.	48 2012 ECCoef ECCoal	8 2015 SMD Super Mediat.	1 2014 UCC United Comp.	4 2012 WDC WDC	119 2008 MNC MNC	43 2009 KL Clique Level	179 2005 BIP Bipartivity	426 1988 GPI GPI Power	116 1991 kRPC k-Reachability	58 2007 SCodd odd Subgraph	586 2004 RWCC RWalk CC					
6	8053 1999 PR Page Rank	239 2008 DSBC DScaled BC	261 1983 σ Stress	477 1991 IEC Immediate Eff.	1 2014 DM Degree Mass	10 2012 LAPC Laplacian C.	8 2012 ABC Attention BC	1099 2001 STRC Straightness C.	8 2015 SNR Silent Node R.	15 2011 HPC Harm. Prot.	26 2011 LAC Local Average	119 2008 DMNC DMNC	3 2013 LR Lurker Rank	2457 1987 β -C β -Cent.	X HYP Hyperbolic C.	27 2002 kEPC k-edge PC	13 2007 FC Functional C.	8 2014 HCC Hierar. CC					
7	484 2005 SC Subgraph	613 1991 FBC Flow BC	14 2012 RLBC RLimited BC	477 1991 MEC Mediative Eff.	69 2010 LEVC Leverage Cent.	35 2010 TC Topological C.	X X SDC Sphere Degree	15 2010 ZC Zonal Cent.	14 2013 CI Collab. Index	11 2013 CoEWC CoEWC	45 2012 NC NC	108 2010 MLC Moduland C.	X X RSC Resolvent SC	1 2014 SWIPD SWIPD	36 2009 XXXX LinComb	0 2014 BCPR BCPR	0 2014 TPC Turnable PC	0 2015 EDCC Effective Dist.					

citations year
C
Name

8800 1979 Freeman Conceptual	942 1966 Salidouni Automatic	573 2006 Borgatti/Evans Conceptual	1130 2005 Borgatti Conceptual	24 2014 Bold/Vigna Automatic	252 1974 Niemi Automatic	6 1981 Kishi Automatic	3 2012 Kitt Automatic	3 2009 Garg Automatic
------------------------------------	------------------------------------	--	-------------------------------------	------------------------------------	--------------------------------	------------------------------	-----------------------------	-----------------------------

2065 1934 Moreno Historic	1546 1950 Bavelas Historic	780 1948 Bavelas Historic	1475 1951 Leavitt Historic	297 1962 Borgatti/Evans Conceptual	3649 2001 Jeong et al. Empirical	4167 1998 Tsai/Ghoshal Empirical	961 1993 Ibarra Empirical	71 2008 Valente Empirical
---------------------------------	----------------------------------	---------------------------------	----------------------------------	--	--	--	---------------------------------	---------------------------------

- “Traditional”
- Betweenness-like
- Friedkin Measures
- Miscellaneous
- Path-based
- Specific Network Type
- Spectral-based
- Closeness-like

©David Schoch (University of Konstanz)

Upcoming lab session & Homework for next week

- **Today:** stick around if you are already certain that you will take the course, and want to find a teammate for the project already
- Lab session: Introduction to NETWORKX
- Work on Assignment 1

- “Homework”: make serious progress with Assignment 1
- Make choice of participation in course explicit. Un-enroll if you wish to drop the course