Social Network Analysis for Computer Scientists

Frank Takes

LIACS, Leiden University

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

Lecture 2 — Advanced network concepts & centrality
Recap
Networks
Notation

Concept

- Network (graph)
- Nodes (objects/vertices/actors/entities)
- Links (relationships/edges/ties/connections)
  - Directed — $E \subseteq V \times V$
  - Undirected
- Number of nodes — $|V|$
- Number of edges — $|E|$
- We assume no self-edges $(u, u)$ and no parallel edges

Symbol

$G = (V, E)$

$V$

$E$

$n$

$m$
Real-world networks

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

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

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

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

- Assortativity
- Reciprocity
- Power law exponent
- Signed graphs
- 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)

Reciprocity

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

  $$m_{\leftarrow\rightarrow} = |\{(u, v) | (v, u) \in E\}|$$

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

  $$r = \frac{m_{\leftarrow\rightarrow}}{m}$$

- Measures the extent to which relationships are mutual
- Value of 1 indicates ...
Reciprocity

- **Reciprocity**: measure of the likelihood of nodes in a directed network to be mutually linked

- Let $m_{\leftarrow\rightarrow}$ be the number of links in the directed network for which there also exists a symmetric counterpart:

  $$m_{\leftarrow\rightarrow} = |\{(u, v) | (v, u) \in E\}|$$

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

  $$r = \frac{m_{\leftarrow\rightarrow}}{m}$$

- Measures the extent to which relationships are mutual

- Value of 1 indicates ...a fully symmetric network.
Power law degree distribution
The probability $p_k$ of a node having degree $k$ depends on the power law exponent $\gamma$:

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

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, $\gamma$ has a value of around 2 to 3
Power law exponent in directed networks

Signed graphs

- Weighted graphs
- Signed edges
- **Friends** and **Foes**
- Indicator of social status

Planar graphs

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

(a) Planar Graph
(b) Non-planar Graph

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

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
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

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.

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
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
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

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)

Graph traversal: BFS

Graph traversal: BFS

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

![Graph traversal: BFS](image)

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
Dense subgraphs and cliques
Subgraphs and cliques

- Subgraph $G' = (V', E')$ of a graph $G = (V, E)$ with $V' \subseteq V$ and $E' \subseteq E$

- Complete graph: graph with all edges present

- **Clique**: complete subgraph

- Maximal clique: complete subgraph of maximal size (cannot be extended with another node)

- Maximum clique: largest possible clique in the graph

- **Clique problem**: find the maximum clique(s) of a graph (one of Karp’s 21 NP-complete problems introduced in 1972)
The subgraph with $V' = \{ t, v, y \}$ is not complete.
Subgraphs and cliques

- The subgraph with $V' = \{t, v, y\}$ is not complete
- **The subgraph with $V'' = \{t, u\}$ is complete and thus a clique of size 2**
The subgraph with $V' = \{ t, v, y \}$ is not complete.

The subgraph with $V'' = \{ t, u \}$ is complete and thus a clique of size 2.

The subgraph with $V''' = \{ t, u, v \}$ is a maximal clique of size 3.
The subgraph with $V' = \{t, v, y\}$ is not complete.
The subgraph with $V'' = \{t, u\}$ is complete and thus a clique of size 2.
The subgraph with $V''' = \{t, u, v\}$ is a maximal clique of size 3.
**The subgraph with $V'''' = \{v, w, x, z\}$ is a maximum clique of size 4.**
Dense subgraphs

- Density: $m/n(n - 1)$ (value $\in [0; 1]$)
- Complete graphs have density 1
- Density (alternative): $m/n$ (value $\in [0; n - 1]$)
- Dense subgraph: subgraph with a high density (almost a clique)
- **Densest subgraph problem**: find a densest subgraph in an undirected graph (NP-hard; reduction from the clique problem)
- Applications in community detection
- Many (exponential) algorithms have been proposed

Densest subgraph

- $S$ is a subgraph of $G$
- $S$ is not a clique, but it is dense (8 out of 10 edges present)

Iterative algorithm

- Density (alternative): $m/n$ (value $\in [0; n-1]$)
- Simple iterative algorithm to find densest subgraph by Charikar et al.:
  1. Compute the average degree of the graph
  2. Delete all nodes whose degree is below the average
  3. Keep track of the density at each step
  4. Go to step 1 if nodes were deleted in this iteration
  5. Output the densest graph seen over all iterations

Charikar et al., "Greedy approximation algorithms for finding dense components in a graph", in LNCS 1913, pp. 84–95, 2000.
Iterative algorithm

- Iteration 1: Current density $16/11 = 1.45$, avg. degree = 2.9
- Best density (iteration) = ...
Iterative algorithm

- Iteration 1: Current density $16/11 = 1.45$, avg. degree = 2.9
- Best density (iteration) = 1.45 (1)

Iterative algorithm

- **Iteration 2:** Current density $9/5 = 1.8$, avg. degree = 3.6
- Best density (iteration) = 1.45 (1)

Iterative algorithm

- **Iteration 2**: Current density $9/5 = 1.8$, avg. degree = 3.6
- **Best density (iteration) = 1.8 (2)**

Iterative algorithm

- **Iteration 3**: Current density $3/3 = 1$, avg. degree $= 2$
- Best density (iteration) $= 1.8$ (2) (unchanged)

Iterative algorithm

- No node deleted in previous iteration, algorithm terminates
- Best density (iteration) = 1.8 (2)

Iterative algorithm performance

FLICKR: Remaining graph vs passes

IM: Remaining graph vs passes
Dense subgraphs for community detection

Source: J. Leskovec, Community detection workshop (1), 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
  \[ C_d(v) = \frac{\text{deg}(v)}{n-1} \]

- **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

Figure: Character co-occurrence network. Node size based on degree.
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.

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

- \(\sigma(v, w)\) 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)

Betweenness centrality
Degree vs. betweenness centrality

Figure: Node size based on degree, color based on 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
Eccentricity centrality
Degree vs. eccentricity centrality
Centrality measures

- But what is the ground truth to verify these measures?
Centrality measures

- But what is the ground truth to verify these measures?
  - Relation with domain-specific ranking on nodes?
But what is the ground truth to verify these measures?
- Relation with domain-specific ranking on nodes?
- Expert knowledge?
Centrality measures

- But what is the ground truth to verify these measures?
  - Relation with domain-specific ranking on nodes?
  - Expert knowledge?
- Edge centrality: which edge/connection between nodes is most important?
  - Often the edges between nodes with a high node centrality value
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 Network Centrality

**Table: Centrality Indices**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Freeman</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sabidoni</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borgatti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borgatti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boldi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vigna</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominien</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kyihi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kittti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Garg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moreno</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bavelas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bavelas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leavitt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leavitt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borgatti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borgatti</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jongs et al.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tash/Chaalal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ibarra</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velante</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

©David Schoch (University of Konstanz)
Computing betweenness centrality
Betweenness centrality

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

\[ C_b(u) = \sum_{v, w \in V} \frac{\sigma_u(v, w)}{\sigma(v, w)} \]

- \( \sigma(v, w) \) 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)

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 \( u \neq w \):
  \[ \sigma(u, w) = \sum_{v \in P_u(w)} \sigma(u, v) \]
  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 $w$

![Graph with nodes and edges labeled with distances]

$u$ to $w$ through $b$, $e$, and $h$.
Counting shortest paths

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

\[
\begin{array}{c}
\text{1} \\
\text{0} \\
\text{1} \\
\text{2} \\
\text{3} \\
\text{2} \\
\text{3} \\
\text{3} \\
\text{4} \\
\text{5} \\
\end{array}
\]

Diagram of network with nodes and edge weights.
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

\[ \sigma(u, w) = \sigma(u, h) = 6 \]
Course project
Course project

- Project makes up 60% of your course grade
- Teams of exactly 2 students
- Deliverables:
  - **Presentation** on a course-related CS paper. 30 minutes for your talk, 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 algorithms on larger datasets
    - Validating algorithms using different metrics
    - Addressing future work posed in the paper
  - Short peer review document
  - Relevant project code and supplementary material
- Bonus for open-source or open science contributions
Presentation

- Present **one** paper from your set of preselected papers
- Show a nice demo, pictures, movies or visualization
- Have a clearly structured presentation
- Briefly discuss your project plans

- Demo presentation will follow
- Discussion with other students is expected (from both presenters and attendees)
Possible presentation structure

- Introduction and motivation of the problem
- Related and previous work
- Formal definition of the problem
- Solution of the problem, algorithms, techniques
- Experimental setup: datasets used, verification measures, etc.
- Results: how well and in which cases does the proposed technique work well?
- Conclusion and future work
Course project

- Read your papers
- Do a bit of research on related literature
- Define and confine the exact problem
- Determine which techniques you are going to compare
- 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
- \textsc{LaTeX}
- 6 to 10 pages, two columns
- Images, figures, graphs, diagrams, tables, references, . . .
- Between 5 and 9 sections
- Peer review after first 3 sections
- Option for “intermediary paper check” before final hand-in
Course project schedule

- Sep 28: deadline for e-mailing your team’s course project topic
- Oct 5: remaining teams divided over topics
- From the end of October: presentations by students
- Nov. 9: deadline for the first three sections (for peer review)
- Nov. 19: optional deadline for “intermediary paper check”
- Nov. 30: deadline for having decent code ready for code review
- Dec. 12: deadline for full course project
- Dec. 21: all grades submitted to student administration
Lab session next week

- From 9.00 to 10:45 in Snellius room 306/308
- Hands-on introduction to NetworkX
- Continue working on Assignment 1
Homework for next week

- Make serious progress with Assignment 1
- Try to fix a team mate for your project
- Make choice of participation in course explicit, should you want to drop the course
- Consult the list of project topics on course website, and think of what you may want to work on
- Ask questions (if any)