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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = (V, E)$</td>
<td>Network (graph)</td>
</tr>
<tr>
<td>$V$</td>
<td>Nodes (objects, vertices, ...)</td>
</tr>
<tr>
<td>$E$</td>
<td>Links (ties, relationships, ...)</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of nodes — $</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of edges — $</td>
</tr>
<tr>
<td>$deg(u)$</td>
<td>Degree of node $u$</td>
</tr>
<tr>
<td>$d(u, v)$</td>
<td>Distance from node $u$ to $v$</td>
</tr>
</tbody>
</table>
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
- 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)

Attribute assortativity

Figure: Attribute assortativity

Reciprocity

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

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

\[
m_{\leftarrow} = |\{(u, v) \in E : (v, u) \in E\}|
\]

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

\[
r = \frac{m_{\leftarrow}}{m}
\]

- Measures the extent to which relationships are mutual
- Useful to compare between networks
Power law degree distribution

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

Useful to compare between similar networks
Power law exponent in directed networks

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

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.

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

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

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

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

- **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

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

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

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

- 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

David Schoch (University of Konstanz)
Homework for next week / Upcoming lab session

- Make serious progress with Assignment 1
- Make choice of participation in course explicit. Un-enroll no later than September 25; anyone registered after that date will get a grade
- Consult the list of project topics on course website, and think of what you may want to work on
- Topic selection on Brigthspace opens Wednesday September 27 at 9:00; first come, first serve
- **Today:** stick around if you are already certain that you will take the course, and want to find a teammate already
- Next lab session: Friday September 29 from 9:00 to 10:45 in Snellius computer rooms
- Chance to ask final questions about Assignment 1