Network visualization in Python

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

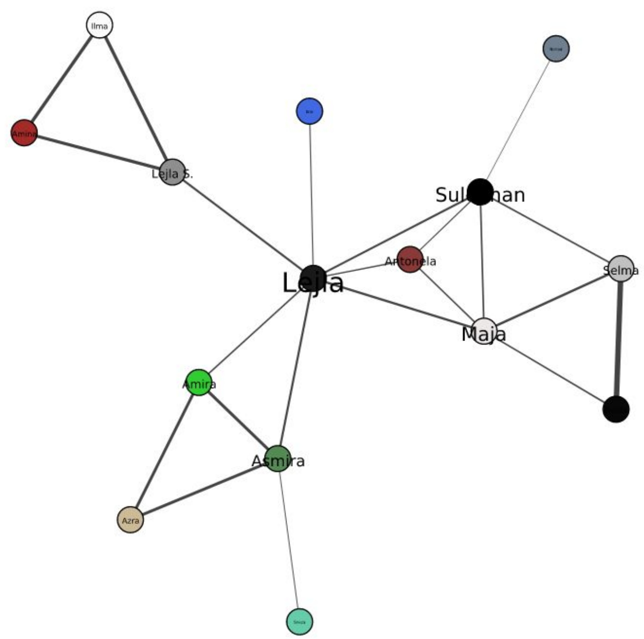
Lejla Džanko
lejla.dzanko@studenti.unipd.it

Overview

1. demonstration of basic usage of the *igraph* library in Python with the friendship network from previous labs
2. relationship between statistics and visualization
3. using *igraph* to find network properties and visualize them
4. task: apply concepts learned to hashtag network

igraph in Python

- install *python-igraph* and *cairocffi* (used for plotting support)
- create a networkx graph from weighted edgelist
- use it to create an igraph
- plotting the graph
- choosing the right layout
- saving the plot



Linking visualization to statistics

- [What do we see when we look at networks?]

We know:

- It is all about the relative distances between the nodes (which pairs are closer or more distant than others)
- Axes have no meaning (you could rotate, scale, flip the image without impacting its interpretation)
- Nodes that are closer have a tendency to be connected, but not always, not directly, and there is a massive amount of exceptions (basically, all long lines represent pairs of connected nodes that nevertheless end up far away)

Linking visualization to statistics

We know:

- A. Noack has shown that the visual clusters correspond to modularity clustering.
- This strategy is bad at representing the asymmetries of directed networks (because visual distances are mutual while topological distances are not).

Linking visualization to statistics

We conjecture:

- Force-driven layout placement are optimizing something, but we do not know exactly what. It just turns out that these algorithms are very useful in practice, and people appreciate the insights they get from it, but we have *no satisfying rationale* to explain that.
- The thing that is optimized is probably a distance between the nodes, because it is how we interpret the visualization. But we do not have any mathematical expression of that distance.

Linking visualization to statistics

- paper on hyptothetical distance that reasonably correlates with the distances in the visualization
- shows that that the visual distance does not correlate well with the geodesic distance (length of the shortest path, counted in number of links) or the the mean commuting time (another intuitive notion of distance in a graph)

https://reticular.hypotheses.org/files/2020/06/Grandjean-Jacomy-2019-Translating-Networks.pdf

Network properties

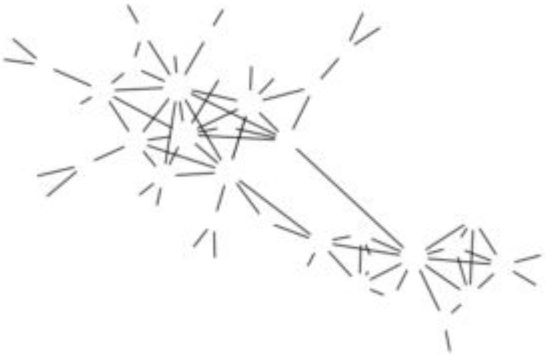
1. Size: how big is the network?

- graph order (how many nodes)

- graph size (how many edges)

- hard to count visually for larger networks but can be intuitively understood

- useful for comparing networks and estimating density



Network properties

2. Density: how connected the nodes are overall?

- visually graph is less or more compact
- there can exist subcomponents that are more compact
- look for cluttered group of nodes, “hairballs”
- visual representation depends on the layout algorithm chosen

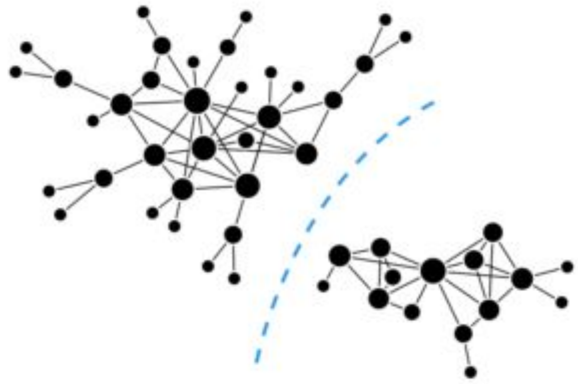
Network properties

3. Connectedness: is there a path between every node?

- visually we see only one component, not groups separated from each other

- one of the easiest properties to spot visually, if the graph is layouted well and no node groups are hidden between others

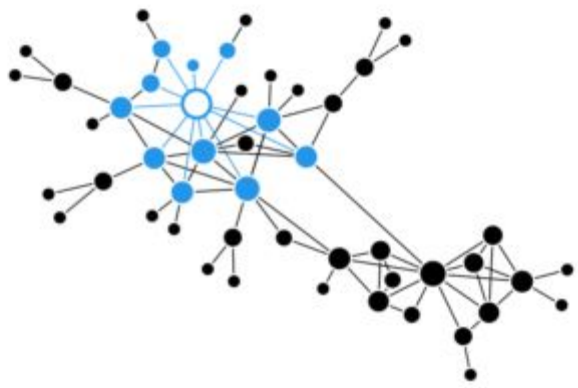
- in connected graph, number of connected components is one



Network properties

4. Degree: how connected a node is?

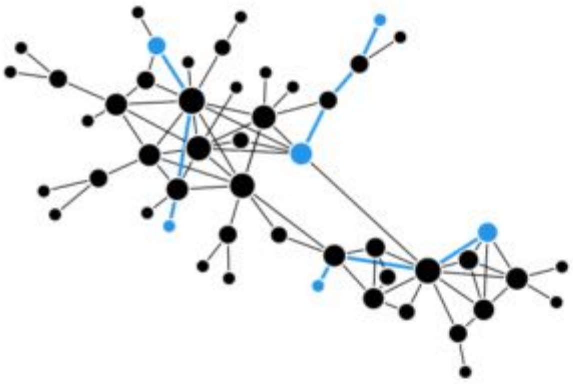
- visually observe edges entering node
- not easy to count visually
- simplest form of centrality
- for structural analysis focus on node degree distribution



Network properties

5. Average path length - how close are the nodes to each other

- impossible to find visually
- loosely related to density
- does not tell much about individual node
- complements diameter which can be affected by outliner nodes



Network properties

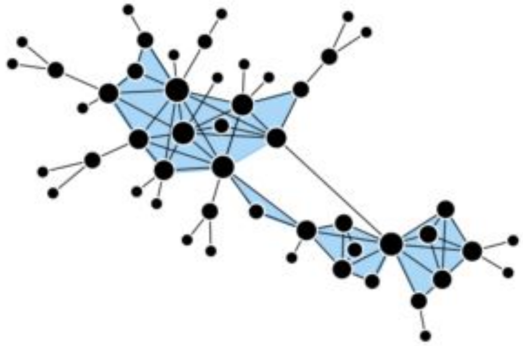
6. Global or average clustering coefficient - graph's tendency to be organized into clusters

- visually noticing clusters

- looking for triangles of fully connected nodes

- global clustering coefficient is number of closed triads over the number of possible triangles and average is average of local clustering coefficients

- gives an idea of entaglement and presence of more localized density



Network properties

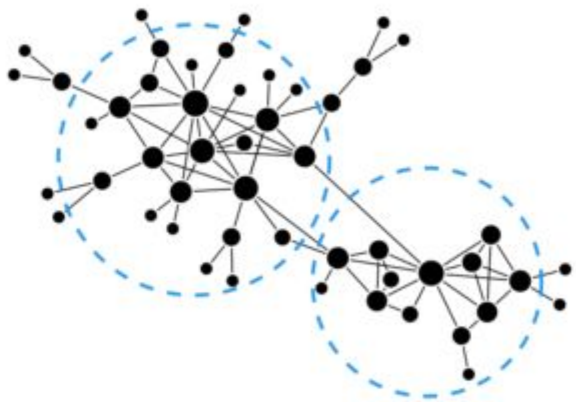
7. Communities - what group of nodes are more connected to each other

- looking for visually dense clusters separated from other nodes

- force-direct algorithms known to show clustering well if properly calibrated

- modularity algorithms

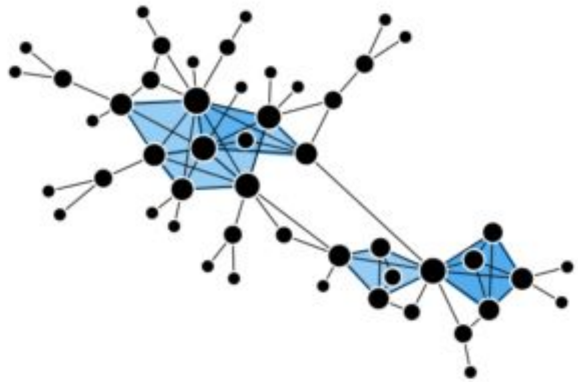
- very useful for network analysis



Network properties

8. Cliques - groups of fully connected nodes

- looking for fully connected groups
- hard to visually find all for complex graphs
- clique detecting algorithms
- “stricter” communities

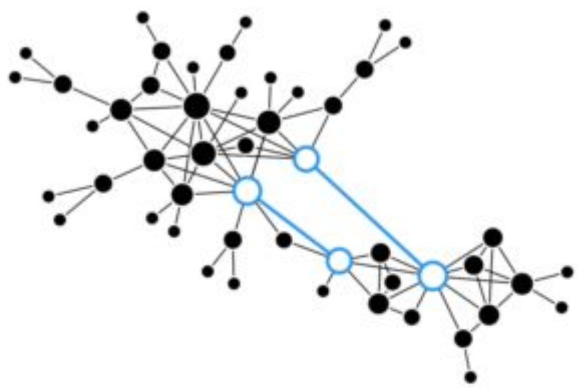


Network properties

9. Centralities

9.1 Betweenness: being a bridge

- connecting distinct groups
- easier to spot bridge edges than nodes
- betweenness centrality: number of shortest paths that go through node

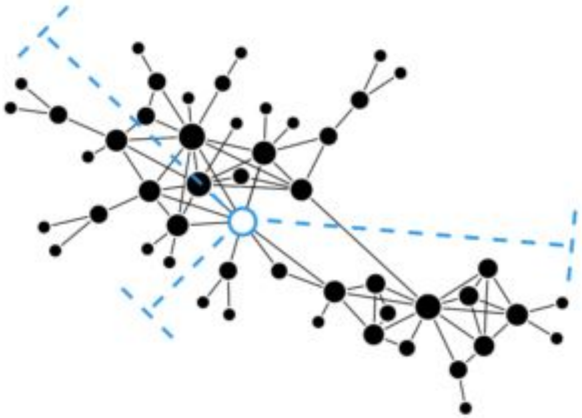


Network properties

9. Centralities

9.2 Closeneses: being in the middle of the network

- center of “land masses” of the graph
- visual estimation, harder in sparse graphs
- most aligned with the notion of the “middle” of the network

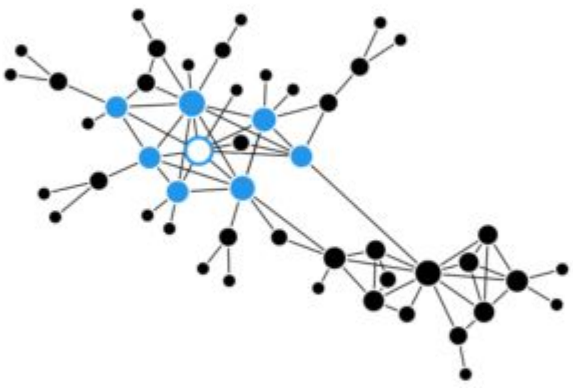


Network properties

9. Centralities

9.3 Prestige: being connected to well connected nodes

- hard to see visually, except if the size of the nodes is visually proportional to the degree centrality, which helps to identify nodes in the hubs' surroundings
- eigenvector centrality or PageRank
- relates to notion of assortativity



Task

- apply what we have shown today to the network of hashtags or your project network

Questions?