

Manipulating bipartite networks and pathways with python

Suzana Santos
FGV-EMAp

July 19th 2022

Bipartite graphs

A bipartite network is composed of two disjoint sets of nodes such that the links occur between a node of one set and a node from the other set.

There is no specific class for representing a bipartite graph with networkx, but there are functions that help to handle bipartite graphs:

```
from networkx.algorithms import bipartite
```

Projections from a bipartite graph

- Function that builds the projected unipartite network of G on nodes 'nodes':

```
bipartite.projected_graph(G, nodes)
```

Pathways

- Check if there is a pathway between 'a' and 'b':
`nx.has_path(G, 'a', 'b')`
- List all shortest pathways between 'a' and 'b'
`nx.all_shortest_paths(G, 'a', 'b')`
- Check if there is a pathway between every two nodes (that is if the graph is connected):
`nx.is_connected(G)`

Connected components

- List all connected components of a graph
`nx.connected_components(G)`

Clustering coefficient

- Compute the clustering coefficient of each node

```
nx.clustering(G)
```