NETWORK ANALYSIS IN PYTHON I

Case study!

Data

- Github user collaboration network
- Nodes: users
- Edges: collaboration on same GitHub repository
- Goals:
  - Analyze structure
  - Visualize
  - Build simple recommendation system

Graph properties

```
In [1]: import networkx as nx

In [2]: G = nx.erdos_renyi_graph(n=20, p=0.2)

In [3]: len(G.edges())
Out[3]: 29

In [4]: len(G.nodes())
Out[4]: 20
```

Graph properties

```
In [5]: nx.degree_centrality(G)
Out[5]:
{0: 0.15789473684210525,
 1: 0.15789473684210525,
 2: 0.15789473684210525,
 3: 0.10526315789473684,...

In [6]: nx.betweenness_centrality(G)
Out[6]:
{0: 0.01949317738791423,
 1: 0.06091617933723196 5,
 2: 0.12768031189083823,
 3: 0.03313840155945419,...
```

Data

- Number of nodes
- Number of edges
- Degree centrality distribution
- Betweenness centrality distribution

NETWORK ANALYSIS IN PYTHON I

Let’s practice!

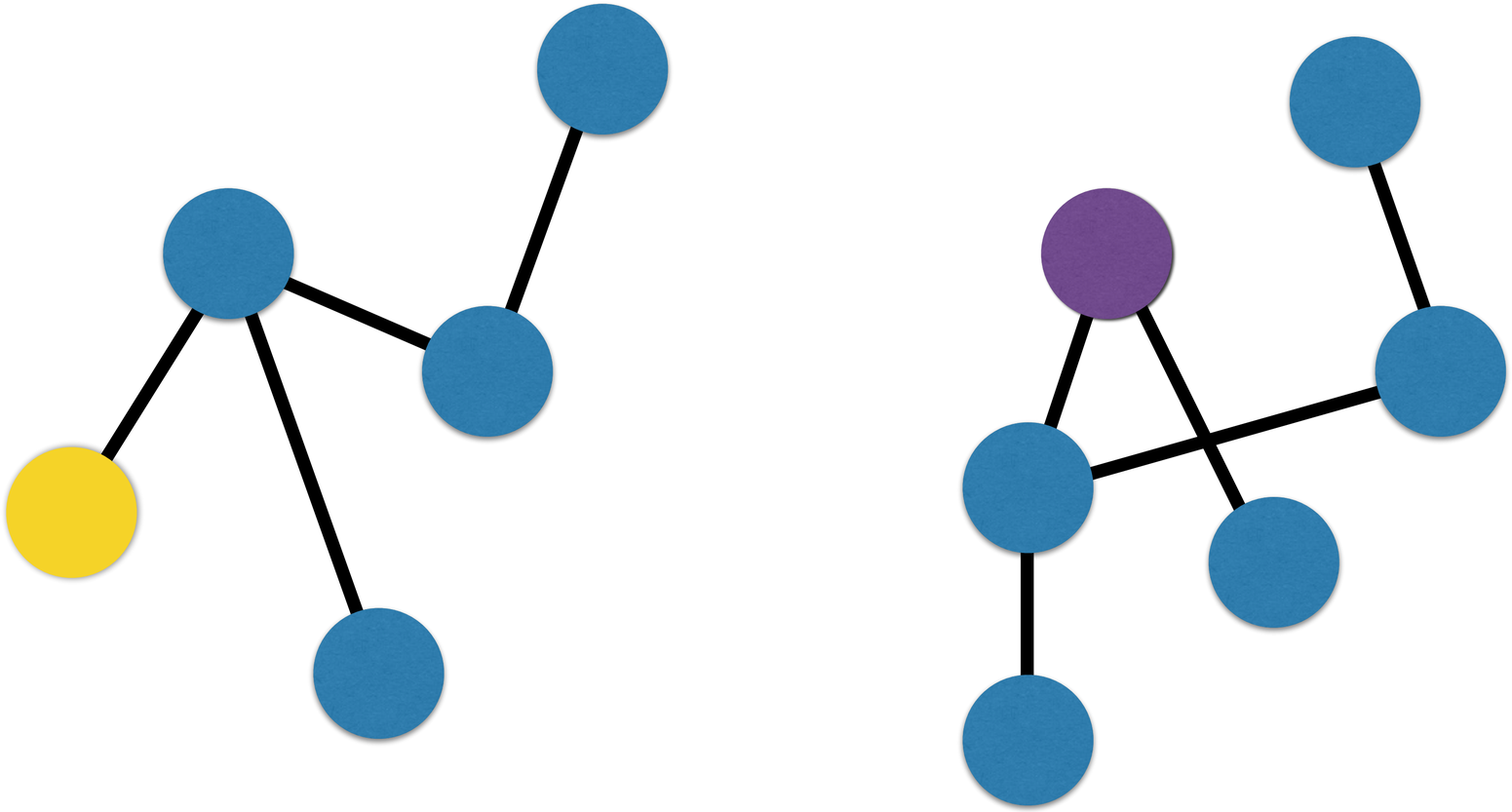
NETWORK ANALYSIS IN PYTHON I

Case study part II: Visualization

nxviz API

```
In [1]: import networkx as nx

In [2]: import nxviz as nv

In [2]: G = nx.erdos_renyi_graph(n=20, p=0.3)

In [3]: circ = nv.CircosPlot(G, node_color='key',
                             node_group='key')

In [4]: circ.draw()
```

Connected component subgraphs



NetworkX API

```
In [1]: import networkx as nx

In [2]: G = nx.erdos_renyi_graph(n=100, p=0.03)

In [3]: nx.connected_component_subgraphs(G)
Out[3]: <generator object connected_component_subgraphs at
0x10cb2c990>

In [4]: list(nx.connected_component_subgraphs(G))
Out[4]:
[<networkx.classes.graph.Graph at 0x10ca24588>,
 <networkx.classes.graph.Graph at 0x10ca244e0>]

In [5]: for g in list(nx.connected_component_subgraphs(G)):
   ...:     print(len(g.nodes()))
   ...:
Out[5]: 99
   ...: 1
```

NETWORK ANALYSIS IN PYTHON I

Let’s practice!

NETWORK ANALYSIS IN PYTHON I

Case study part III: Cliques

Cliques

- Definition:
  - Groups of nodes
  - Fully connected
- Simplest clique: edge
- Simplest complex clique: triangle

Maximal cliques

- Definition:
  - A clique
  - Cannot be extended by adding a node

Finding cliques

```
In [1]: import networkx as nx

In [2]: G = nx.erdos_renyi_graph(n=100, p=0.15)

In [3]: nx.find_cliques(G)
Out[3]: <generator object find_cliques at 0x10ca8bca8>

In [4]: for clique in nx.find_cliques(G):
   ...:     print(len(clique))
```

NETWORK ANALYSIS IN PYTHON I

Let’s practice!

NETWORK ANALYSIS IN PYTHON I

Case study part IV: Final tasks

Final tasks

- Find important users
- Find largest communities of collaborators
- Build a collaboration recommendation system

Final tasks

- Find important users
- Find largest communities of collaborators
- Build a collaboration recommendation system

Final tasks

- Find important users
- Find largest communities of collaborators
- Build a collaboration recommendation system

Final tasks

- Find important users
- Find largest communities of collaborators
- Build a collaboration recommendation system

NETWORK ANALYSIS IN PYTHON I

Let’s practice!

NETWORK ANALYSIS IN PYTHON I

Final thoughts

What you've learned

- The basics of networks and network analysis
- How to find important nodes
- How to identify communities of nodes
- How to apply these concepts in a case study
- How to use the NetworkX and nxviz packages
- How to write network algorithms
- Practical applications of network analysis

NETWORK ANALYSIS IN PYTHON I

See you in the next course!