Name: Jalal Mansoori

Karate Club Network Analysis
Using Networkx

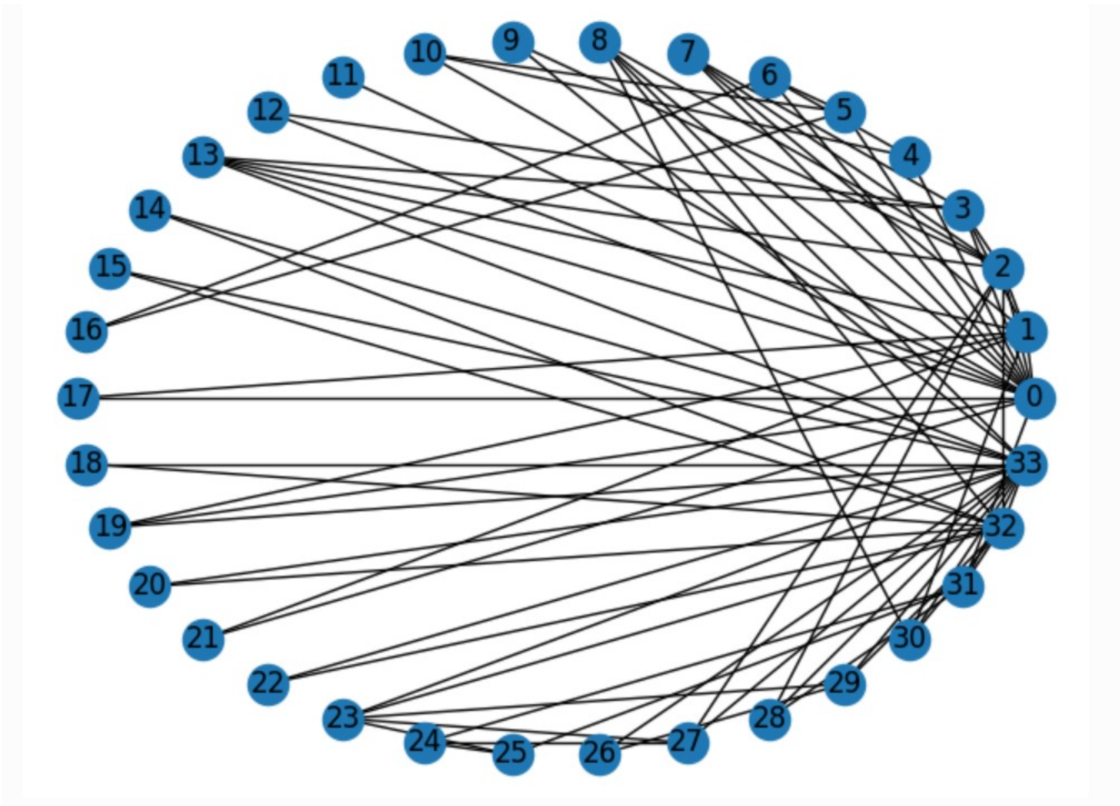
Little bit of History:

This hello world of social network analysis was studied by Wayne W. Zachary for a period of three years from 1970 to 1972. A conflict was arose between instructor and administrator after which some members formed new club around instructor and other member founded a new instructor or gave up.

Representation of nodes and edges in karate club network:

- Here nodes represents each member of club
- edges represents an interaction between pair of members who interact outside of club

Network Visualization



Name: Jalal Mansoori

Network Analysis

1) Distance Measures

Measuring distance can play a important role in study of information flow in network such as in karate club.

So how can we summarize the distance between all pair of nodes in our graph ?

1. **Eccentricity**

The Eccentricity of a node n is the largest distance between n and all other nodes.

```
print(nx.eccentricity(G))
```

```
{0: 3, 1: 3, 2: 3, 3: 3, 4: 4, 5: 4, 6: 4, 7: 4, 8: 3, 9: 4, 10: 4, 11: 4, 12: 4, 13: 3, 14: 5, 15: 5, 16: 5, 1
7: 4, 18: 5, 19: 3, 20: 5, 21: 4, 22: 5, 23: 5, 24: 4, 25: 4, 26: 5, 27: 4, 28: 4, 29: 5, 30: 4, 31: 3, 32: 4,
33: 4}
```

2. Radius

The radius of a graph is the minimum eccentricity.

```
In [33]: print(nx.radius(G))
3
```

3. Diameter

Maximum shortest distance between any pair of nodes.

```
In [35]: print(nx.diameter(G))
5
```

Now with the help of eccentricity and radius we can easily find out nodes at **center** and **periphery**.

Center → eccentricity=radius
Periphery → eccentricity=diameter

Nodes at periphery

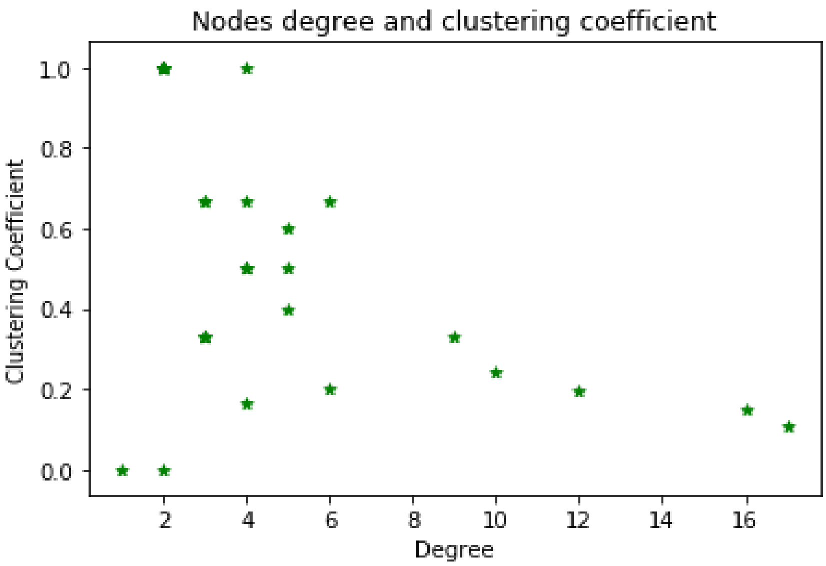
```
In [36]: nx.periphery(G)
Out[36]: [14, 15, 16, 18, 20, 22, 23, 26, 29]
```

Nodes at center

```
In [37]: nx.center(G)
Out[37]: [0, 1, 2, 3, 8, 13, 19, 31]
```

Name: Jalal Mansoori

2) Nodes Degree and Clustering Coefficient



So plot of karate club network between nodes degree and clustering coefficient showing for most of the nodes as degree **increases** their clustering coefficient is **decreasing**.

Average clustering coefficient in graph is between 0.4 and 0.6

```
In [38]: nx.average_clustering(G)
Out[38]: 0.5706384782076823
```

Network basic info

Network	Nodes	Links	Directed Undirected	N	L	Average Degree
Karate Club Network (1970)	Member of club	Interaction between pairs of members	Undirected	34	78	4.5882

Name: Jalal Mansoori

Bipartite Graph

```
In [45]:  from networkx.algorithms import bipartite
          bipartite.is_bipartite(G)

Out[45]:  False
```

Karate club graph is not a bipartite graph but if we are able to make it bipartite by following:

1. Representing one set of nodes as instructors like [0, 33] assuming is old and new instructor respectively.
2. Representing other nodes as members [1, 2, 3, 4, ...32]

Using bipartite graph we can create a network to answer following questions:

- Which instructors has common members connections ?
- Which members has common instructors ?

References

1) https://towardsdatascience.com/introduction-to-graphs-part-1-2de6cda8c5a5
2) https://networkx.github.io/documentation/stable/auto_examples/graph/plot_karate_club.html