Visualizing Scale Free Network using Networkx Python Library

Submitted as part of mandate requirement of T1-21-22- AI 608
Network Science for the Web

Submitted by : MT2020013 Apoorv Panse

Guided by : Prof. Srinath Srinivasa

Abstract

The most common phenomena occuring in universe is the normal distribution of various random events. Where most of the data values cluster around mean. Many continuous data in nature follows normal distribution. Now when it comes to social networks, we see that the distribution of connections is not normal, but it follows power law distribution. Which is crucial in determining if the network is scale free or not. Knowing if a network is scale free is useful to further analyse it for finding influencers, information spreaders, controlling epidemics and so on. In this article I will explain few terminologies about Scale Free Networks and then move on to show how we can use NetworkX library available in Python to prove that a social network is scale free using SNAP Facebook dataset example.

Scale Free Networks

The network is said to be scale-free if the properties or attributes of the network are independent of the enormity or its size. Overall when the network grows or scales up, its underlying structure remains the same. Now this is very useful in observing social phenomenon because as population grows, and network increases, the underlying characteristics of that network may remain the same. Which can be use to model various social strategies.

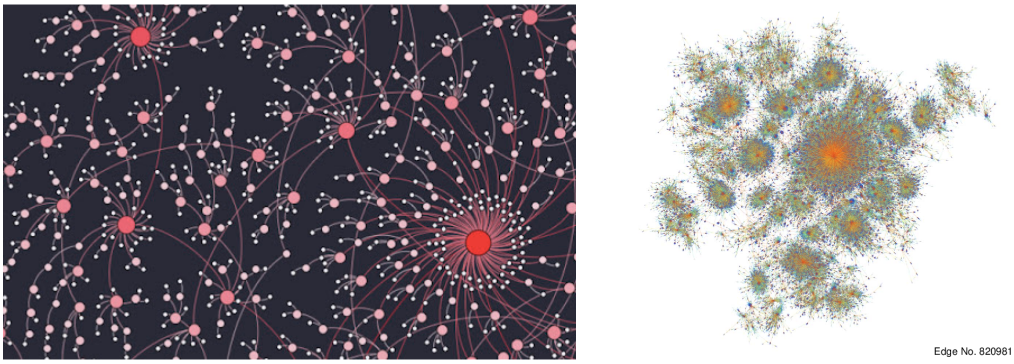


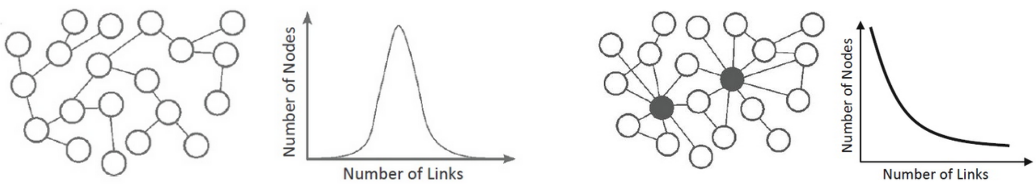
Fig Sources : http://social-dynamics.org/scale-free-network/
https://www.networkpages.nl/scale-free-networks-a-controversial-topic-properly-solved-by-extreme-mathematics/

A network is deemed scale free if the fraction of nodes with degree k follows a power-law distribution $k^{-\alpha}$, where $\alpha > 1$

Power Law

This kind of distribution is based on the relationship in which relative change in one quantity gives rise to a proportional relative change in other quantity. This is useful because it helps in identifying the regularities in the network especially in large complex systems. Because large network may have characteristics that is independent of its scale. Social networks tends to follow power law as it has preferential attachment. That means it is far more attractive to connect with people who already have more connections. This is also based on the idea of rich getting richer. LinkedIn, Twitter, Facebook and all major social network tends to follow power law distribution when it comes to connectivity of people. This leads to Scale Free Networks.

In larger network, small world principle holds, that says if a node is connected to other nodes in small number of steps, then it may act as a hub between all other nodes connecting to it.



The Barabasi-Albert model

The **Barabasi-Albert model** (a.k.a. BA model) introduced in 1998 explains the power-law degree distribution of networks by considering two main ingredients: growth and preferential attachment (Barabasi and Albert 1999). The `algorithm` used in the BA model goes as follows.

1. Growth: Starting with a small number (m_0)of connected nodes, at every time step, we add a new node with $m(< m_0)$ edges that link the new node to m different nodes already present in the network.
2. Preferential attachment: When choosing the nodes to which the new node connects, we assume that the probability P that a new node will be connected to node i depends on the degree k_i of node i, such that $P \sim \frac{k_i}{\sum_i k_i}$

Numerical simulations and analytic results indicate that this network evolves into a scale-invariant state with the probability that a node has k edges following a power law with an exponent $\gamma = 3$. The scaling exponent is independent of m, the only parameter in the model.

Facebook Network Analysis

We have computing power to study networks and analyse it on a large scale. Social networks like Facebook have billions of node (people) and the connectivity between them can be studied using social network analysis techniques. If we try to define it, SAN is basically the process of investigating different social structures using mathematical techniques, network principles and graph theory. Below is the graph of smaller dataset of Facebook connections using SNAP database.

```
import networkx as nx
with open("./facebook.txt") as file:
    data = file.read()
edges = []
for pair in data.split("\n"):
    vertices = pair.split(" ")
    edge = (int(vertices[0]),int(vertices[1]))
    edges.append(edge)
graph = nx.Graph(edges)
graph
```

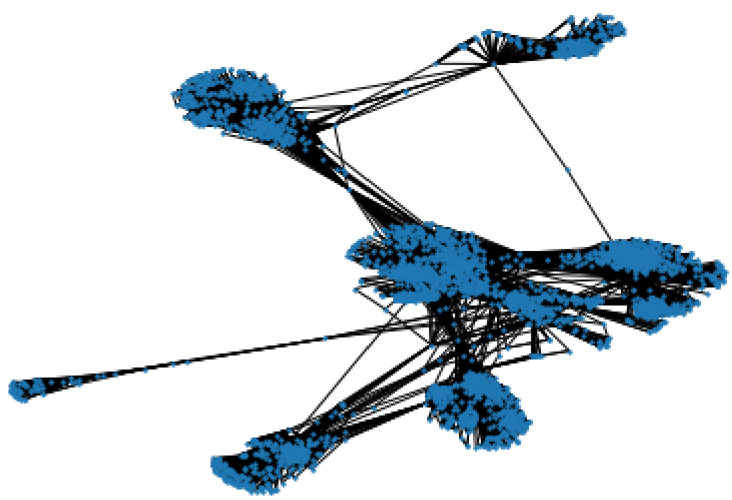
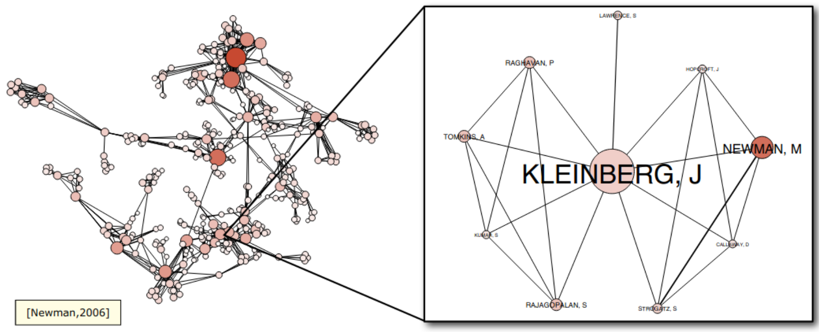


Fig : Graph Visualization of Facebook Network.

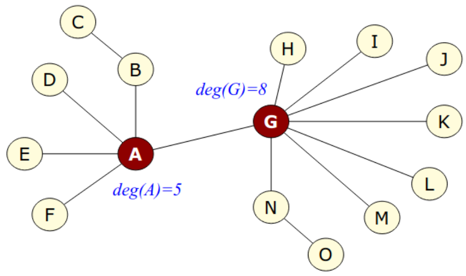
Finding Properties of Facebook Network Using Networkx

Terminologies

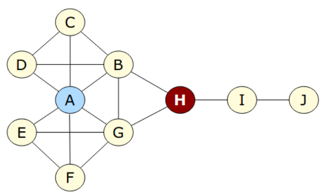
- Graph : It is a way of representing relationships among collected objects.
- Nodes : It is the collected object itself. People can be represented by nodes.
- Edges : Relationships between the nodes. In SAN, it is simply unidirectional or bidirectional connection between people.
- Degree of a Node : Number of edges connected to that perticular node.
- In-Degree : Number of edges incident on a node.
- Out-Degree : Number of edges outgoing from a node.
- Ego-centric Network : It focuses on individual and not the network as a whole. When we build local network of an individual based on nodes connected to focal ego.



- **Degree Centrality** : It is nothing but count of number of edges a vertex has. Nodes with high degree tends to be the hub and have higher connectivity and strong relationships with other vertices.



- **Path** : Sequence of vertices joining one another. The number of edges involved in that sequence is the path length.
- **Diameter** : Longest shortest path between any two nodes in the graph.
- **Betweenness centrality** : Nodes that usually appear in most of the shortest paths between other nodes tend to have higher betweenness centrality score.



  In above graph, A has higher degree centrality and H has high betweenness centrality. It is clear that having less degree centrality will not necessarily result in lower betweenness centrality score.
- **Eigen Vector Centrality** : It is proportional to the centrality scores of its neighbours. It depicts that a vertex have higher significance if it is connected to other vertex of higher significance. So if a node is connected to an influencial node, it has higher centrality score than nodes that are connected to less influencial nodes.
- **Load Centrality** : The load centrality of a node is the fraction of all shortest paths that pass through that node.

- **Small World Networks** : Even if the nodes are not directly connected, every node can be reacher to other node with small number of hops. Facebook and other social networks see this phenomenon where there is very high chance of having not more than 6 hops between any two nodes. Also termed as six degrees of separation.

Analysis of Facebook dataset

- The dataset is contained in a file Facebook.txt is the modification of SNAP facebook dataset where each line represents connection between two nodes.

Dataset statistics	
Nodes	4039
Edges	88234
Nodes in largest WCC	4039 (1.000)
Edges in largest WCC	88234 (1.000)
Nodes in largest SCC	4039 (1.000)
Edges in largest SCC	88234 (1.000)
Average clustering coefficient	0.6055
Number of triangles	1612010
Fraction of closed triangles	0.2647
Diameter (longest shortest path)	8
90-percentile effective diameter	4.7

```
graph.number_of_edges()
88234

graph.number_of_nodes()
4039

Average Shortest Path length :
nx.average_shortest_path_length(graph)
3.6925068496963913

Average Degree :
stat.mean([graph.degree(n) for n in graph.nodes()])
43.69101262688784
```

```
nx.spring_layout(graph)
def func(items):
    sorted_items = sorted(items, key = lambda x:x[1], reverse = True)
    sortedCounts = {k:v for k,v in sorted_items}
    impNodes = list(sortedCounts)[:10]
    return impNodes
deg = func(nx.degree_centrality(graph).items())
load = func(nx.load_centrality(graph).items())
between = func(nx.betweenness_centrality(graph).items())
nx.draw(graph, node_color = "yellow", node_size = 2, alpha = 1)
nx.draw(graph, nodelist = deg, node_color = "red", node_size = 30, alpha = 1)
nx.draw(graph, nodelist = load, node_color = "blue", node_size = 30, alpha = 1)
nx.draw(graph, nodelist = between, node_color = "green", node_size = 30, alpha = 1)
```

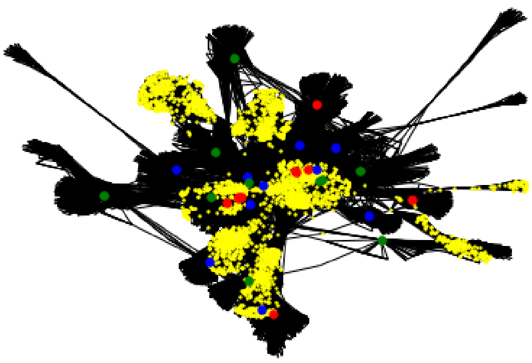
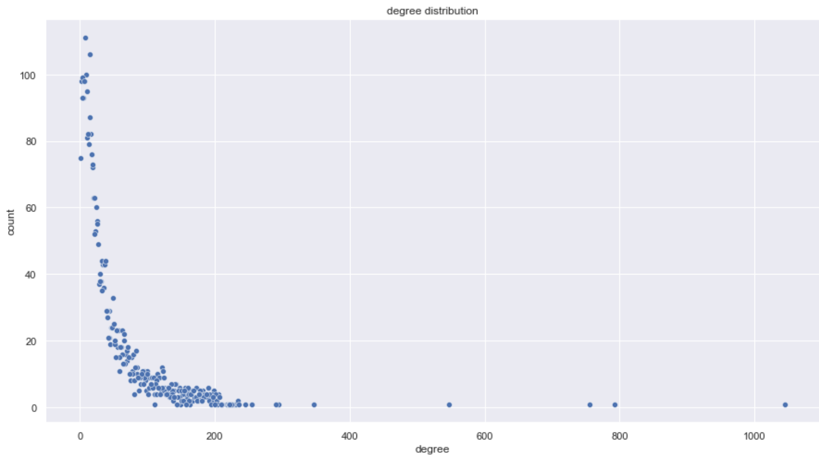


Fig. Top Centrality Measure visualization

- Top ten nodes by degree centrality represented in Red.
- Top ten nodes by load centrality represented in Blue.
- Top ten nodes by betweeness centrality represented in Green.
- Other nodes represented by yellow.

Degree Distribution Curve :



Above graph shows that the degree distribution curve follows powerlaw. Which indicates that it is a scale free network. The code of analysis is attached in the zip file (MT2020013.ipynb).

References

1. https://www.futurelearn.com/info/courses/complexity-and-uncertainty/0/steps/1855
2. https://necsi.edu/power-law
3. https://en.wikipedia.org/wiki/Social_network_analysis
4. https://snap.stanford.edu/data/egonets-Facebook.html
5. https://mathinsight.org/scale_free_network
6. https://github.com/swinton/Web-Science-Summer-School-2011
7. J. McAuley and J. Leskovec. Learning to Discover Social Circles in Ego Networks. NIPS, 2012.
8. https://networkx.org/documentation/stable/index.html
9. http://www.scholarpedia.org/article/Scale-free_networks