<https://methods.sagepub.com/dataset/StudentGuide/modularity-in-karate-1977?isEmbed=1>   
<https://journals.sagepub.com/doi/full/10.1155/2015/849140>   
<https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8>

<https://www.mygreatlearning.com/blog/introduction-to-spectral-clustering/>

<https://www.analyticsvidhya.com/blog/2020/04/community-detection-graphs-networks/>  
<https://gawron.sdsu.edu/python_for_ss/course_core/book_draft/Social_Networks/Networkx.html>

**Github**:

Girvan-Newman (Yelp dataset) : <https://github.com/hsuanhauliu/girvan-newman-community-detection>

ALL: <https://github.com/topics/community-detection>

<https://github.com/topics/girvan-newman>

Modularity Maximization:

<https://github.com/zhiyzuo/python-modularity-maximization>

ALL: <https://github.com/topics/modularity-maximization>

Spectral clustering: (graph Laplacian)

<https://github.com/wq2012/SpectralCluster>

<https://github.com/yfhanhust/MiniBatchSpectralClustering>   
<https://github.com/joonalillfors/spectral-clustering>   
<https://github.com/dppalomar/spectralGraphTopology>

Dolphin:   
<https://gist.github.com/jexp/e7a33c22d1403a3cc53e>

**Dataset: KARATE**

### Implementing the Girvan-Newman Algorithm for Community Detection in Python

I will use [Zachary’s karate club Graph](https://en.wikipedia.org/wiki/Zachary%27s_karate_club) to demonstrate how you can perform community detection using the Girvan-Newman Algorithm.

Zachary’s karate club is a social network of a university karate club. The network became a popular example of community structure in networks after its use by Michelle Girvan and Mark Newman in 2002.

We will start off with loading the required libraries: *networkx* and *matplotlib:*

import networkx as nx

import matplotlib.pyplot as plt

%matplotlib inline

# load the graph

G = nx.karate\_club\_graph()

# visualize the graph

nx.draw(G, with\_labels = True)

len(G.nodes), len(G.edges)

**(Gives the nodes and edges of the graph)**

in the Girvan-Newman method, the edges of the graph are eliminated one-by-one based on the EBC score. So, the first task is to find the EBC values for all the edges and then take off the edge with the largest value. The below function performs the exact task:

def edge\_to\_remove(graph):

G\_dict = nx.edge\_betweenness\_centrality(graph)

edge = ()

# extract the edge with highest edge betweenness centrality score

for key, value in sorted(G\_dict.items(), key=lambda item: item[1], reverse = True):

edge = key

break

return edge

Now the function below will use the above function to partition the graph into multiple communities:

def girvan\_newman(graph):

# find number of connected components

sg = nx.connected\_components(graph)

sg\_count = nx.number\_connected\_components(graph)

while(sg\_count == 1):

graph.remove\_edge(edge\_to\_remove(graph)[0], edge\_to\_remove(graph)[1])

sg = nx.connected\_components(graph)

sg\_count = nx.number\_connected\_components(graph)

return sg

Next, let’s pass the Karate Club graph to the above function:

# find communities in the graph

c = girvan\_newman(G.copy())

# find the nodes forming the communities

node\_groups = []

for i in c:

node\_groups.append(list(i))

**node\_groups**

These are the node sets belonging to two communities in the Karate Club graph according to the Girvan-Newman algorithm. Let’s use these node sets to visualize the communities discovered in the graph:

# plot the communities

color\_map = []

for node in G:

if node in node\_groups[0]:

color\_map.append('blue')

else:

color\_map.append('green')

nx.draw(G, node\_color=color\_map, with\_labels=True)

plt.show()