**Girvan-Newman Community Detection Algorithms**

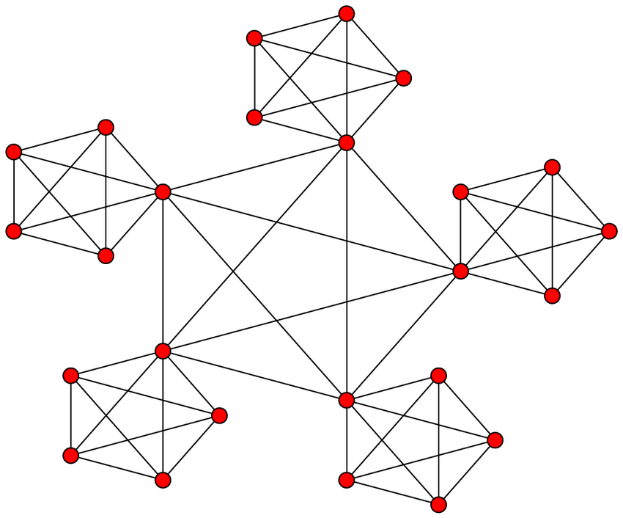
It is the technique for the detection and analysis of community structure depends upon the iterative elimination of edges with the highest number of the shortest paths that pass through them.

It follows the divisive hierarchical clustering based on edge betweenness.

**Divisive hierarchical clustering:**Start with the complete graph and take off the edges iteratively. Edge with the highest weight is removed first and repeated.

**What is community?**

“A community, with respect to graphs, can be defined as a subset of nodes that are directly connected to each other and loosely connected to the nodes in the other communities in the same graphs.”



If we think about the social media platforms such as Facebook, Instagram and Twitter, where we are connect with other people. Where we are connected with the different social circles, that social circle can be a group of relatives, university mates or colleagues, etc.

These social circles are communities.

**Need of community detection:**

* We need community detection to discover people with common interest and keep them connected.
* We need community detection in machine learning to detect groups with similar properties and extract groups for various purpose.
* We need community detection in stock market.

Over large scale, it is very difficult to configure the different communities, so we need the algorithms to partition the network in different communities.

**Explanation:**

This algorithm removes the edges of graph based on edge between centrality values.

**Edge between centrality values:**

It is the number of shortest paths that passes through the edge in a network. Each edge is given an EBC score based on the shortest paths among all the nodes in graph.

**Shortest path:**

Shortest path means the least distance amount between two nodes.

**Steps:**

1. Repeat until no edges are left.
   1. Calculate EBC for every edge in graph.
   2. Remove the edge with highest EBC.
   3. Calculate EBC for remaining edges.
   4. Connected components are communities.

**Applications:**

**Examples:**

Take one node let say A.Directly connected nodes are B and D.

Shortest path of AB and AD.

Shortest path from A to C and E goes through D and B.

Shortest path to last node F goes through nodes B,D,C and E.

Edges are arranged according to the shortest path now we will add scores to them.

**Adding Scores:**

B=1,D=1

Because there is one shortest path to A.

C=1

Because there is only one shortest path from C to A.

E=2

There are 2 shortest paths from E to A, ABE, ADE.

F=3

There are 3 shortest path from F to A , ABCF, ABEF, ADEF.

**Computing scores foe edges:**

Here we will move backward direction from F to A.

FC=1/3=0.33

FE=2/3=0.667

CB=1+0.33(FC) =1.33

EB= (1+0.667(FE))/2=0.835

ED= (1+0.667 (FE))/2=0.835

BA= (1+1.33+0.835 (EB))/1=3.165

DA= (1+0.835 (ED))/1=1.835

**Repeat these steps for each node:**

|  |  |
| --- | --- |
| **Edges** | **Edge betweenness** |
| **AB** | 3.165+1.5+1.33+0.835+0.5+0.667 =8 |
| **AD** | 1.835+0.5+0.33+1.835+0.5+0.33 =5.33 |
| **BC** | 3.165+1.5+1.33+0.835+0.5+0.667 =8 |
| **BE** | 0.835+2+0.835+0.835+2+0.835 =7.34 |
| **CF** | 1.835+0.5+0.33+1.835+0.5+0.33 =5.35 |
| **DE** | 3.165+1.5+1.33+0.835+0.5+0.667 =8 |
| **EF** | 3.165+1.5+1.33+0.835+0.5+0.667 =8 |

**Divide scores by 2 and get the EBC**

AB=8/2=4

AD=5.33/2=2.67

BC=8/2=4

BE=7.34/2=3.67

CF=5.35/2=2.67

DE=8/2=4

EF=8/2=4

According to algorithm after computing EBC remove the highest score edges. As we can see AB,DE,BC and EF have highest score.so we will strike them out and it will give us three sub graphs.

These 3 are subgraphs called communities.

**Pseudo code:**

Repeat

Let n be number of edges in graph

For i=0 to n-1

Let b[i] be betweenness centrality of edge i

If b[i] > max-b then

Max-b=b[i]

Max-b-edge =i

}}

Remove edge I from graph

Until number of edges in graph is 0

**Advantages:**

1. It helps in finding the community from graph.
2. It is suitable in small input of data.

**Disadvantages:**

This algorithm is not efficient in time with networks containing large number of nodes and data.

**Complexity Analysis:**

It has the time complexity, increasing up to O (n^3) or O (m^2n) on a scattered graph having m edges and n nodes. As a result, Girvan-Newman is generally not used on large-scale networks. Its maximum node count is a few thousands nodes or less.