VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Social and Information Networks

(CSE – 3021)

PROJECT REVIEW

Team Details:

Name	Reg. No
Vibhu Kumar Singh	19BCE0215
Avnish Tiwari	19BCE0634

Teacher – Ms. Manjula R.

Community Detection in Social Networks

Introduction:

A network is a set of nodes and links connecting them. Studying network properties belongs to the graph theory area which is a part of discrete mathematics. With the development of computers that can perform operations on very large amounts of data, modelling and analysis of network performance is extended to other scientific disciplines and becomes an interdisciplinary topic.

A community, with respect to graphs, can be defined as a subset of nodes that are densely connected to each other and loosely connected to the nodes in the other communities in the same graph. For example, social media platforms such as Facebook, Instagram or twitter, we end up being connected with people belonging to different social circles, these circles are nothing but communities.

Detecting communities in a network is one of the most important tasks in network analysis. In a large-scale network, such as an online social network, we could have millions of nodes and edges. Detecting communities in such networks becomes a herculean task.

Therefore, we need community detection algorithms that can partition the network into multiple communities.

There are primarily two types of ways of detecting communities in graphs:

- Agglomerative
- Divisive Methods

In our Project we have explored a divisive algorithm Known as the Girvan Newman Algorithm.

In divisive methods, we start with the complete graph and take off the edges iteratively. The edge with the highest weight is removed first. At every step, the edge-weight calculation is repeated, since the weight of the remaining edges changes after an edge is removed. After a certain number of steps, we get clusters of densely connected nodes.

Under the Girvan-Newman algorithm, the communities in a graph are discovered by iteratively removing the edges of the graph, based on the edge-betweenness centrality value.

Objective/Goal:

The expansion of the web nowadays and emergence of a large number of Social networking sites (SNS) have given users the power to easily interact on a shared platform. The social network can be represented by a graph consisting of a set of nodes and edges connecting these nodes.

We use Detection of these communities as it can be beneficial for numerous applications such as in E-commerce, finding a common research area, finding a set of like minded users for marketing and recommendations.

Our project presents a comparison between the brute force approach and the Divisive Algorithmic Approach like the Girvan-Newman Algorithm.

Scope:

The scope of the project would be to analyze the advantages and disadvantages of using the Girvan-Newman Algorithm over the brute-force approach. We also give an analysis on the time complexity of both the algorithms when run on various graphs of different sizes.

Abstract:

Real world networks often have community structure. It is characteristic that the groups of nodes are connected denser within themselves and rarely with each other. The Girvan Newman method for the detection and analysis of community structure is based on the iterative elimination of edges with the highest number of the shortest paths that go through them. By eliminating edges, the network breaks down into smaller networks, i.e. communities. Our Project depicts the implementation of the Girvan-Newman method where multi-edge removal is allowed, and presents the results of the application of both methods to the existing real social network (Zachary karate club), the computer-generated network and the tumor genes and their mutations network.

Literature Review:

S.no	Title	Source of Paper	Year of Publication
1	Clustering and community detection with imbalanced clusters	Research Gate	2016
2	Efficient vector influence clustering coefficient based directed community detection method	IEEExplore	2017

3	Multi-layer network local community detection based on influence relation	IEEExplore	2019
4	Divisive Algorithm Based on Node Clustering Coefficient for Community Detection	IEEExplore	2020
5	A community detection algorithm based on graph compression for large-scale social networks	ELSEVIER	2020

Objectives, Limitations and Future Scope:

1. **Clustering and community detection with imbalanced clusters. (2016) (research Gate)**

   Objective

   Present a graph partitioning issue that detects and minimizes cut partitions under minimum size check of partitions to detect Community with Imbalanced Clusters size.
   The community detection technique is graph partitioning and the algorithm used is semi supervised learning.
   They demonstrated the advantage of their approach through observation of real datasets and synthetic datasets of clustering data using community detection

   Limitations

   They used the Semi Supervised Learning Algorithm in which Iteration results are not stable. It is not applicable to network-level data. It has low accuracy.
   Also, the network type is not directed but undirected.

   Future Scope

   They proposed the partition constrained min-cut (PCut) framework, which seeks min-cut partitions under minimum cluster size constraints
   Since constrained min-cut is NP-hard, they would adopt existing spectral methods (SC, GRF, GTAM) as a black-box subroutine on a parameterized family of graphs to generate candidate partitions and solve PCut on these partitions in the future.

2. **Efficient vector influence clustering coefficient based directed community detection method(2017) (IEEEXplore)**

   Objective

   In this paper the author Proposed a triangle structure in directed graphs based on conventional community detection algorithms. In this network they evaluate information transfer gain (ITG) of nodes for clustering the graph, then combining the

different clusters based on the influence coefficient ITG. Social media produces a large data set in the real world, to analyze the data, using an artificial network to process this data fast and accurately and they claim their algorithm time complexity correctness is acceptable.
The community detection technique used is hierarchical clustering.

Limitations
There are certain limitations like this algorithm could only be used for undirected network type and not directed.Also the data set should be real and not synthetic.
Also, there are certain issues in hierarchical clustering like:
Lack of a global objective function and when the 2 clusters are merged they cannot be split up again.

Future Scope
Their follow-up research would include: (1) optimizing the stop condition in the iterations to fetch up the greediness of the algorithm, (2) implementing efficiency of our parallel algorithm and experimenting in a distributed environment, and (3) integrating our algorithm to form a visualization analysis software application.

3. **Multi-layer network local community detection based on influence relation(2019) (IEEEXplore)**

Objective
Present an impact connection put together multilayer network technique on local community detection model, to join this community the immediate and backhanded impact connection network is utilized. In view of the similarity, it is used to measure the length of the path and calculate the weight of the local node.
It compared six datasets and discovered the rightness and steadiness of this technique.
Dynamic community detection technique is used with clustering algorithm being used as the primary algorithm.

Limitation
The clustering algorithm has certain limitations like it is not suitable for non-convex data, relatively sensitive to the outliers, the number of clusters needed to be preset, and the clustering result sensitive to the number of clusters.
Also, we can see that it could only be used for undirected network type and not directed. Also, the data set should be real and not synthetic.

Future Scope
In multilayer data sparse network connection and large data sets, the method of this paper can identify the same or higher quality data sets and have better time efficiency.

In the sensor network, they will consider the changes of local community structure in the multi-layer network by adding sensor nodes dynamically in the future.

4. **Divisive Algorithm Based on Node Clustering Coefficient for Community Detection(2020) (IEEEXplore)**

Objective

The authors Proposed to remove intercommunity edges using a divisive algorithm over community detection. The community detection used is Partitioned clustering. In iteration, the network size increases linearly and finding the relationship between node clustering coefficients defined from a micro perspective through an undirected graph and node clustering coefficients into the divisive algorithm can greatly improve the time efficiency. The time complexity of this algorithm is O(Nd2) and it works on real as well as synthetic data sets.

Limitations

The CCE rule is not given in the form of theorems because the increase in the node clustering coefficient is not a necessary and sufficient condition for the increase in the node network density.
The CCE rule is a typical local index, so the result of the algorithm is unstable when the network is traversed randomly.
Their method is only suitable for global non-overlapping community detection, however the community structures in complex networks are diverse.
While the time complexity is improved there are certain problems with Partitioned Clustering like Poor cluster descriptors and High sensitivity to initialization phase, noise and outliers. Also, whenever a point is close to the center of another cluster; it gives poor results due to overlapping of data points.

Future Scope

It would be a considerable challenge to detect community structure on these large-scale networks. Using local indicators to detect communities in large-scale networks will be one of the main developments in the future.

5. **A community detection algorithm based on graph compression for large-scale social networks**

Xingwang Zhao , Jiye Liang , Jie Wang
(ELSEVIER, 2020)

Objective

This paper proposes a community detection algorithm based on graph compression. Specifically, a compressed graph is first obtained by iteratively merging vertices with a degree of 1 or 2 into their neighbors with a higher degree. Then, two indices, i.e., the

density and quality of vertices, are defined to evaluate the probability of vertices as community seeds. By considering these two measures together, in a compressed social network, the number of communities and the corresponding initial community seeds are determined simultaneously. After obtaining the community structure of the compressed social network via seed expansion, the community results are propagated to the original social network.

Limitation

The proposed algorithm based on graph compression can improve the efficiency while maintaining the effectiveness for large-scale networks, but it is only suited for undirected networks.

Future Scope

How to extend the graph compression strategy to the community discovery of attribute networks and multilayer networks is the focus of future research.

Overall Description:

Brute-Force Algorithm:

Using the brute force strategy, we will partition the graph's nodes into two or more communities. The brute force method is attempting every possible division of nodes into communities and determining whether or not they are correctly divided. For this work, we'll apply the brute force method.

This method is applied on a barbell graph as well as on the famous Zachary's Karate Club Graph.

Girvan Newman Algorithm:

For the detection and analysis of community structures, the Girvan-Newman algorithm relies on the iterative elimination of edges that have the highest number of shortest paths between nodes passing through them. By removing edges from the graph one-by-one, the network breaks down into smaller pieces, so-called communities.

The idea is to find which edges in a network occur most frequently between other pairs of nodes by finding edge betweenness centralities. The edges joining communities are then expected to have a high edge betweenness. The underlying community structure of the network will be much more fine-grained once the edges with the highest betweenness are eliminated which means that communities will be much easier to spot.

Implementation:

The algorithm used for brute-force is as follows:

- Create a graph of N nodes and its edges or take an inbuilt graph like a barbell graph.
- Now take two lists as FirstCommunity and SecondCommunity.
- Now start putting nodes into communities like put 1st node in FirstCommunity and rest N-1 nodes to SecondCommunity and check its inter and intra edges.
- Now we will make combinations using itertools.
- Repeat steps 3 and 4 for every combination.
- Now check which division is best by taking the ratio of intra/number of inter-community edges.
- Now find the value of FirstCommunity and SecondCommunity with maximum ratio and print that value.

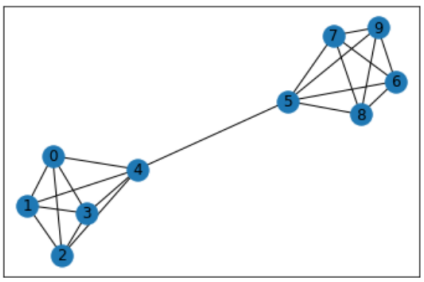
The modules used are the following:

```
In [2]: import networkx as nx
        import itertools
        from time import perf_counter
```

Barbell Graph

Visualising the input Graph

```
In [4]: G = nx.barbell_graph(5, 0)
        nx.draw_networkx(G)
```



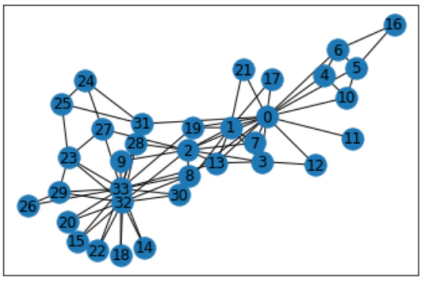
Now when this algorithm is run on a barbell graph, the output is as expected:

```
[ [0, 1, 2, 3, 4] ] , [ [5, 6, 7, 8, 9] ]
```

Karate-Club Graph

Visualising Zachary's Karate Club Graph

```
In [6]: G1 = nx.karate_club_graph()
        nx.draw_networkx(G1)
```



Since the brute-force algorithm divides and checks every possible combination of the input graph, it is clear that the time complexity is exponential. When we run the algorithm on the above graph, the output is not displayed since the execution time is really high (around 2^34, ie: 10^10).

```
In [*]: start = perf_counter()
        communities_using_brute(G1)
        end = perf_counter()
        execution_time = (end - start)
        execution_time
```

Here, it can be seen that the algorithm is still running.

The algorithm for Girvan-Newman is as follows:

- For every edge in a graph, calculate the edge betweenness centrality.
- Remove the edge with the highest betweenness centrality.
- Calculate the betweenness centrality for every remaining edge.
- Repeat steps 2–4 until there are no more edges left.

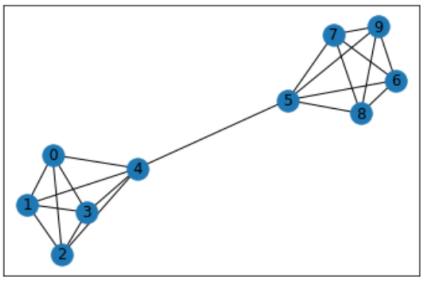
The modules used are the following:

```
In [1]: import networkx as nx
        import matplotlib.pyplot as plt
        from time import perf_counter

        %matplotlib inline
```

Barbell Graph

Visualising the input Graph

```
In [4]: G = nx.barbell_graph(5, 0)
        nx.draw_networkx(G)
```



Now when this algorithm is run on a barbell graph, the output is as expected:

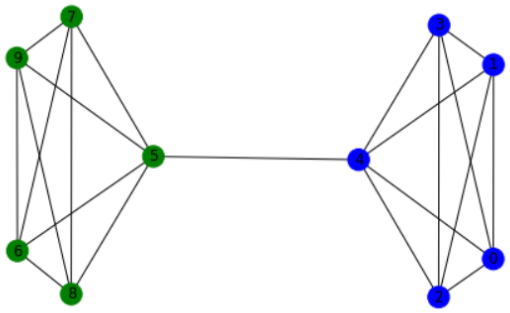
```
In [8]: # find the nodes forming the communities
        node_groups1 = []

        for i in c1:
          node_groups1.append(list(i))
```

```
In [9]: node_groups1
Out[9]: [[0, 1, 2, 3, 4], [5, 6, 7, 8, 9]]
```

The following communities are detected:

Visualising the ouput

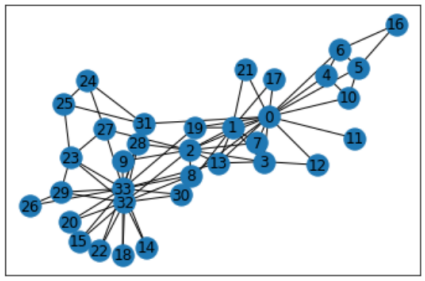
```
In [10]: color_map1 = []
         for node in G1:
             if node in node_groups1[0]:
                 color_map1.append('blue')
             else:
                 color_map1.append('green')

         nx.draw(G1, node_color=color_map1, with_labels=True)
         plt.show()
```



Karate-Club Graph

Visualising Zachary's Karate Club Graph

```
In [6]: G1 = nx.karate_club_graph()
        nx.draw_networkx(G1)
```



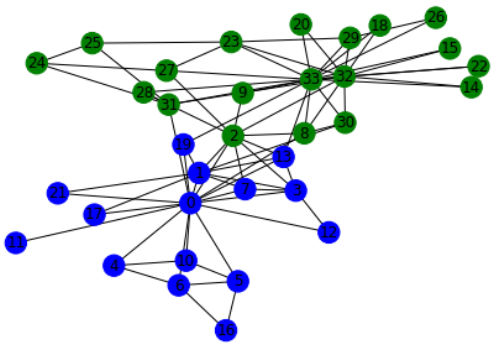
Since, the time-complexity of Girvan-Newman algorithm is far better than the brute-force method, it easy run on the Karate Club Graph and the following communities are detected:

Ouput

```
In [15]: node_groups

Out[15]: [[0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 19, 21],
          [2, 8, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]]
```

Visualising the ouput

```
In [16]: 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()
```



The nodes colored in green belong to one community and the blue nodes represent the other community.

Result Analysis and Discussion:

We discussed the various community detection algorithms that are used on real-world social networks. Among them, we have chosen two for analysis.

Time Complexity Analysis of Brute-force vs Girvan-Newman:

Input Graph	**Brute-Force**	**Girvan-Newman**
Barbell (5,0)	0.099 seconds	0.0027 seconds
Karate Club Graph	---	0.1798 seconds

As a result, it is clear that the Girvan-Newman Algorithm is much faster than the Brute-force Algorithm. Also, it is worth noting that it is not practical to use the brute force method on social networks of large sizes.

Appendix:

CODE: https://github.com/Vibhukumar10/Community-Detection-In-Social-Networks