Community Detection In Social Networks

Maneesh Boddu
dept. of Computer Science
Georgia State University
Atlanta, GA
mboddu1@student.gsu.edu

Purna Sai Pushkal Kalipindi
dept. of Computer Science
Georgia State University
Atlanta, GA
pkalipindi1@student.gsu.edu

***Abstract*—In today's world, technology is being developed rapidly and use of social websites like Twitter, Facebook has increased. Due to this social network gained large attention. Most of them uses social media to be in contact with their friends. And these real-world networks have community structures. It is group of nodes that have higher likelihood of connecting to each other than nodes from other community. There are algorithms to find these communities in networks, called community detection. Community detection is a task to group communities based on common interests, modules etc., using data present in graph topology. Detecting communities is a difficult task due to network topology and overlapping communities. In this paper, we detected communities using Girvan-Newman algorithm and used modularity to find optimized community. This Girvan-Newman algorithm is based on edge-betweenness which deletes edges that has highest betweenness. By deleting edges, the networks breaks into communities. These results are evaluated using modularity. Facebook dataset is used to detect communities.**

***Index Terms*—Social network analysis; Community detection; Girvan-Newman algorithm; Modularity;**

I. Introduction

A. Social Networks :

Network is interconnected system of things. Social networks are networks connecting people. These are complex systems and are most commonly used for interaction among people. Complexity networks are made up of interconnected nodes. There is a collection of entities in the network, and there is at least one relationship between this entities. Social networks can be modelled as graphs G(V,E), where nodes are entities and edges are relationship between entities. Entities in different groups form communities. Basically, this social graphs are undirected as for Facebook graphs.

B. Community :

Modern networks are growing exponentially in size, variety, and complexity. We often think of networks being organized into cluster, modules, groups or communities. Social network analysis is a process to represent the characteristics and structure of the networks by establishing similarities among entities. Important aspect of communities is they contain communities of entities that are connected by edges. Communities are formed based on this similarities. Communities in network are dense group of vertices, which are tightly coupled to each other inside the group and loosely coupled o rest of vertices in network.

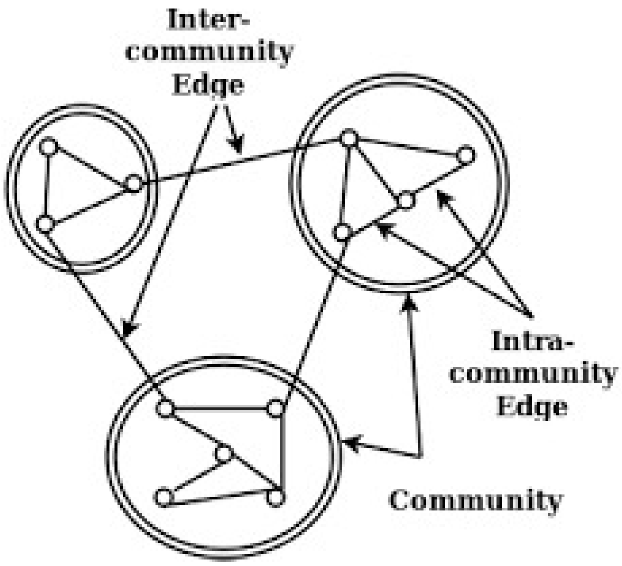


Fig. 1. Communities with its intra and inter community edges

From the above figure, we can see that edge within the community is intra-community edge and edge between the community is inter-community edge.

C. Community Detection

Community structure in social networks can give us clues about the nature of social interactions within the community represented. It can be considered as a summary of the whole network thus easy to visualize and understand. Sometimes, community can reveal the properties without releasing the individual privacy information. Plays key role in understanding functionality of complex networks. In community detection algorithms, main method is to find group of entities that are interested in a network. It provides information about common interests or beliefs among people which makes them unique compared to other communities. It is similar to network partitioning problems, where network will be partitioned into n groups and number of edges will be minimized. There are many approaches to detect communities. This community detection problem has many challenges from a computational perspective, as it is related to problem of clustering large. A large number of techniques has been suggested to find optimal communities in reasonably fast time In this paper we did analysis of community structure of largest social network i.e.Facebook.

There are two approaches for community detection :

- Agglomerative methods.
- Divisive algorithms.

Graphs may contain hierarchical structure.So, hierarchical clustering techniques may be used to identify the multi-level community structure of the graph. Main methods to find communities are hierarchical clustering methods. These methods can be agglomerative methods or divisive methods. It depends on how we want to divide communities. Both methods can be represented using dendrogram i.e., hierarchical tree clusters as nodes and nodes as leaves. In agglomerative process, dendrogram is built from leaves to the root, and in divisible process from root to the leaves. Both methods produce dendrogram.

1) Agglomerative methods: The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It follows Bottom-up approach. Traditional hierarchical agglomerative clustering algorithm starts with empty graph that has nodes without edges. It starts with vertices and removes all edges. Initially, it considers all nodes as a separate cluster and iteratively merge them based on high similarity, which in result ends with unique community. It will add edges iteratively. Edge weight can be calculated using different methods. These edge weights can be number of edge independent or node independent paths between nodes. Paths are said to be edge independent and node independent if they don't share edges or vertices. Number of this vertices or edges will be removed from graph are represented by this paths. Iteration will be done till all edges are removed. At end of the algorithm, communities will be formed.

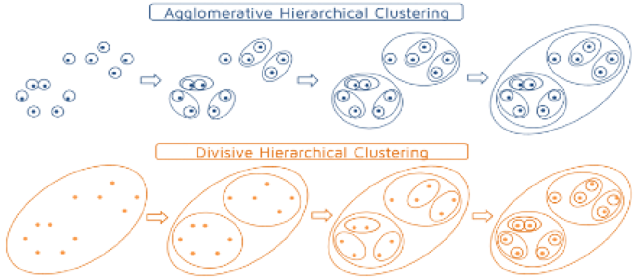

Agglomerative Hierarchical Clustering
Divisive Hierarchical Clustering

Fig. 2. Agglomerative and Divisive Hierarchical Clustering

2) Divisive methods: It follows Top-down approach. All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. Initially, it considers the entire network as single cluster. Then iteratively splits graph into partitions by eliminating links joining nodes with low similarity and ends up with unique communities. It will remove edges iteratively. Weight will be calculated in every step as weight of remaining edges will have effect on edge removal. In this paper, we are using Girvan-Newman algorithm which is also a divisive method.

II. Background and Motivation

In morphology and structure, social networks and forests have certain common characteristics. In below fig have a visualization of a social network of more than 30,000 members, which is like a dense forest. Around the giant portion are some boundary groups. Social network groups often consist of core vertices, core backbones and border vertices, with forest trees, shrubs and grass being identical in their morphology and structure. If six subgraphs are visualized, some are sparse, like shrubs and grass, some are dense, like trees. If we may regard a social network as a forest, trees, shrubs, grass are the communities in the forest. Social network groups have some relationships or have no relationships, this function is like these forest trees, shrubs and grass. New small communities can be extracted from big communities in social networks, this role is like these trees, shrubs and grass in the forest. Between social networks and the forest, there are several characteristics that are similar.

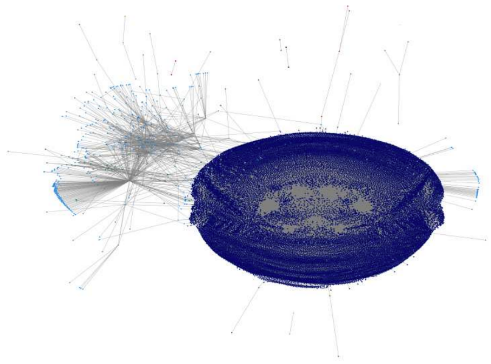


Fig. 3. visualization of an complete social network with more than 30000 users

A group is defined within the graph as a subset of vertices such that the links between the vertices are denser than the links with the rest of the network (Radicchi et al., 2004). Why do we regard a group as a tree? A tree consists of roots, trunks and leaves, and vertices and relationships are a group. Some relationships are strong and some relationships are weak; some vertices are vertices of the center and some vertices, like leaves, are on the boundary. We consider that the strong relationships are tree trunks, tree roots are some core vertices, leaves are some vertices on the border. The problem of community detection in networks is one of separating the vertices of a given network into non-overlapping groups in its most basic form. Connections within groups are relatively dense, whereas those within groups are comparatively dense are sparse(Newman, 2013).

Discovering communities from networks is like discovering trees from the forest in this article. But how can a tree be found in the forest? In order to calculate the edges, we just have the edges and vertices, we need a metric, we call the metric a backbone degree. Let a group become like a tree, 8 The edges, then are the tree trunks. An edge consists of

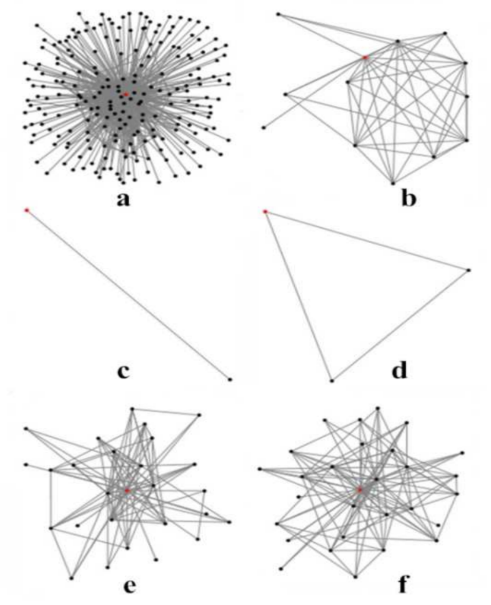


Fig. 4. visualization of six subgraphs that belongs to above social network

two vertices and a relationship, so these three variables must be determined by the backbone degree. The edges are like bamboo sections.Two joints and a bar are composed of each segment.

The relationships are the bars, and the measure of neighborhood overlap may be used to represent the intensity of the bar. To represent the strength of the joint, the network weight measure can be used. If this metric works, it is like this method to detect a group from a network: firstly finding the edge with the largest backbone degree, secondly finding the nearest backbone degree based on the vertex until the boundary of this community, repeating the above step in the rest of the vertices until all the vertices are divided. Controlling the option of the backbone will make the algorithm more dynamically extensible. For example, if no backbone in the split vertices is selected by the algorithm, the group detection is not overlapping, otherwise it is overlapping. We primarily address the non-overlapping detection of the population in this paper.

Meaning of the idea of community concludes how to establish a community in the network, at that point, how to characterize the idea of community? The instinct of community is a bunch of vertices that associations between the vertices are denser than associations with the remainder of the network. Radicchi proposed the idea of community quantitatively: in a solid community every vertex has more associations inside the community than with the rest of the diagram; in a powerless community the amount of all degrees inside the community is more significant than the amount of all degrees toward the remainder of the network. Luccio and Sami proposed the idea of the community called minimal groups in 1969, Lawler renamed them LS sets in 1973. LS set resembles a solid community. Another definition is called k-center; a k-center of a diagram G is a maximal associated subdiagram of G in which all vertices have a degree at any rate k. K-center resembles a powerless community. From examined above, we found that those thoughts of the community are brief and precise, however not to be envisioned, and no point by point portrayal of the inward structure. Is there a thought that can imagine the view of the community? This implies that the idea can give a clear limit, what's more, inner structure.

Alongside the advancement of informal network research lately, individuals set forward plenty of new ideas about informal community structures, for example, frail and solid connection, connect, easy route, neighborhood cover, and so on. In the event that we investigate the association between the vertices in a network, we discover those joins include: feeble and the solid connection, extension, easy route, etc. On the off chance that the contact as parts of the trees so that a community can be viewed as a tree, a social network can be a community backwoods. The network contains solid communities in community woods, and the other has vulnerable communities hedged. On the off chance that as indicated by this speculation, we can give more natural properties to the community and rethink community thought.

III. Related Work

A. A Large Scale Community Structure Analysis in Facebook:

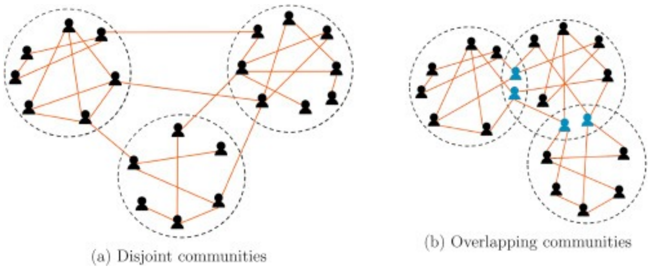
The major issue in current computational social sciences is the ability to understand the dynamics of actions such as communication and social relationships. This paper deals with analysis of patterns and the social dynamics with the Facebook data-set. The network community's mesoscopic structure and the perspectives of communities are discussed. To contribute to the data-set, a sample of user's social relationships, unveiled the users interacting and finally analyzing features using some patterns. This has lead to different techniques of sampling and clustering methodologies.

B. Generalized Louvain Method for Community Detection in Large Networks

The issue of finding the network structure in large systems has been broadly researched. A few effective methodologies dependent on nearby information has been proposed and are doable when analyzing extensive systems on account of their low computational costs. The fundamental downside of the current procedures is that they don't consider world-wide data about the topology of the system. This paper explains a strategy that has to favorable circumstances. The previous is that it misuses both the local and worldwide data. The last is by using some advancement, it proficiently gives great outcomes.

The Local Spectral Algorithm to identify medium-sized communities in huge social graphs. Second, other recent work has also focused on developing local and/or near linear time heuristics for community detection include. Third, there also exists work which views communities from a distinct perspective.

C. Conventional community detection


(a) Disjoint communities

(b) Overlapping communities

Fig. 5. Disjoint and Overlapping Communities

1) Disjoint community detection.: Many algorithms that are existing are based on modularity optimization, including the Clauset-Newman-Moore algorithm and Louvian method as discussed above. Nodes are divided onto groups that are disjoint and chance of connectivity between two nodes depends only on node memberships.

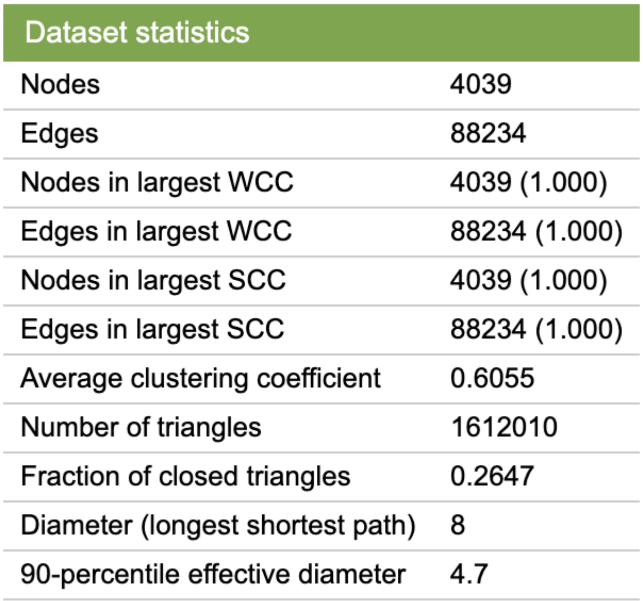
2) Overlapping community detection.: Clique Percolation algorithm is the first algorithm to analyse overlapping communities. In this a community is defined as a maximal k-clique percolation chain. Some of the recent methods based on seeds expansions are proposed. For instance, OSLOM uses a fitness function and combines small clusters into a considerable large one. Link communities is an algorithm that determines communities by using hierarchical clustering on the links where node groups that are overlapped are obtained.

IV. Workflow

A. Data Collection:

We focus on Facebook data since Facebook is the most generally utilized Digital Social Network (DSN) for scholastic examination and the information is moderately simple to acquire. We considered the data present at https://snap.stanford.edu/data/ego-Facebook.html.

Facebook doesn't have a framework to derive public information related to users and due to the lack of data availability we couldn't acquire public information directly from the platform, by means of a sampling process. The above data set consists of circles(friends list), node features and ego networks. The above data has been collected from a survey participants of the application. The Facebook data has been modified by replacing internal id's instead of the original values. Also, while feature vectors from this data set have been provided, the interpretation of those features has been concealed.



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

Fig. 6. Facebook Dataset Statistics

Table shows the list of nodes, edges and triangles present in the data set. The graph is a network with n*m pair of participants and friends. The number of triangles will be greater than value of random graph. For example, If A, B are friends and A, C are friends then there will be a higher chance that B,C to be friends. Likewise counting number of triangles will help to measure the extent to which the graph looks like a social network.

The input file contains two values, which represents nodes. The relationship between these nodes is represented by lines which creates an edge between these nodes.

B. Data Graph Generation:

As we have collected the data set, the next step is to generate graphs for the Facebook data. To analyze or visualize these nodes and data we have couple of tools like

1) Networkx
2) Pajek
3) IGraph
4) Gephi

Networkx has been used in the paper.

1) Networkx: NetworkX is a Python programming language package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

It provides functionalities like:

1) Tools for the study of the structure and dynamics of social, biological, and infrastructure networks.
2) a standard programming interface and graph implementation that is suitable for many applications.
3) a rapid development environment for collaborative, multidisciplinary projects.
4) an interface to existing numerical algorithms and code written in C, C++, and FORTRAN; and
5) the ability to painlessly work with large nonstandard data sets.

There are four types of social network Graph in social network Analysis.

1) In the first mode Network, We will have one set of node with ties that are connected to these nodes.
2) In the second mode Network, Vertices are divided into two sets. and these vertices are related to vertices in the other set.
3) Two mode Network Graph contains Networks with two sets of nodes with ties that are established among nodes belonging to different sets.
4) In the Multi relational Network there will be multiple kinds of relations between nodes.

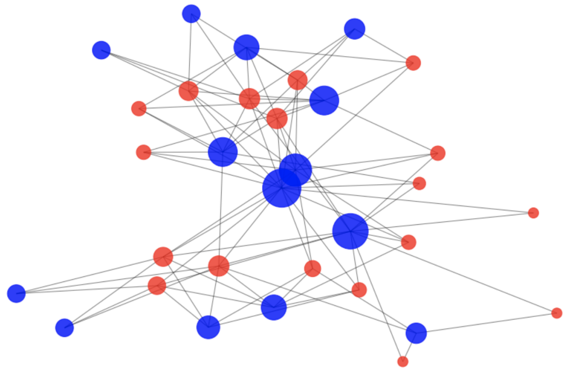


Fig. 7. The above graph shows a sample graph generated by Networkx with labels.

C. Graph Partitioning

Once we generate graphs, it must be divided into communities.It divides the graph into clusters, such that the number of links in a cluster is more denser than the number of edges between the clusters. In this paper, we used Girvan-Newman algorithm for generating communities. More about this algorithm will be discussed in Algorithm design section.

D. Evaluation

This Girvan-Newman algorithm generates only communities, but it doesn't give information about which split is best. So we used modularity to detect optimum community. More about the modularity is discussed in further report.

V. Algorithm Design

To analyse a community network of a known structure several methods have been proposed and used with varying success rate. Some of them which are widely used are Max Flow Min cut Algorithm.

Minimum cut approach is designed basically for the abstraction of the materials which are flowing through the edges in a particular network. However it is applicable on both set of directed or undirected graphs. In this, the network is divided into certain number of parts which are approximately of equal size.

Maximum flow works in accordance with the Maximum Flow Problem. It states that usually in a flow network we have to maximize the amount of flow from source (starting node) to the sink (last node). However the conversation of flow is followed on every intermediate node i.e. Incoming Flow=Outgoing Flow

Hierarchical methods have several shortcomings with respect to detecting the communities in a social network. To remove those shortcomings, Newman and Girvan presented their algorithm to detect the communities in social networks in 2002.. They brought a new concept, popularly known as "edge between" to detect the community in large and complex networks. According to the algorithm, we simply focus on those edges that are least central to the network and those edges are considered as most "between" communities, instead of calculating the measure of the edges which are central to the network. That means, "edge between" score of a particular edge can be calculated as the number of times it appears in the shortest path matrix of the graph. Then, we remove the particular edge which has the highest "edge between" score according to the algorithm and we get first two communities. If there are more than one links or connections between the communities, then we will remove the edges which connect both the communities serially according to the highest "edge between" score. We will remove all the edges in the network in this way until we get the single nodes. The procedure of Newman-Girvan algorithm is stated below:

A. Newman-Girvan algorithm

The Girvan Newman strategy for the detection and investigation of community structure relies on the iterative end of edges with the most noteworthy number of the briefest ways that go through them. By disposing of the edges, the organization separates into littler organizations, for example networks.

The Algorithm was to discover which edges in an organization happen most as often as possible between different sets of nodes by discovering edges betweenness. The edges joining networks are then expected to have high edge betweenness. The hidden community structure of the organization will be a lot of fine-grained once we take out edges with high edge betweenness. For the evacuation of each edge, the estimation of edge betweenness is O(EN); subsequently, this current calculation's time multifaceted nature is $O(E^2N)$.

- Calculate the betweenness score for all the edges in the network.
- After removal of the edge, betweenness score will be recalculated for all the remaining edges in the network.
- Calculate the betweeness for remaining edges.
- Step 2 will be repeated until we remove all edges or we get the single node in the network.

After step 2 we will get first level of regions i the partitioning of the graph. It is efficient technique for computing the betweenness of edges. After step 4 this may break some of the existing components into smaller components; if so, these are regions nested within the larger regions. After elimination, resulting community structure of network will be much fine-grained.

Unlike the earlier approaches, Girvan-Newman algorithm provides result of reasonable quality, due to which it has been implemented in a number of standard software packages.

B. EDGE BETWEENNESS

Betweenness calculation is the most important part of this algorithm. The betweenness measure of an edge can be defined as the total number of shortest paths between all node pairs of a network/graph passing through it.

We can calculate betweenness using three different measures: geodesic edge betweenness, random-walk edge betweenness and current-flow edge betweenness.

In order to calculate edge betweenness it is necessary to find all the shortest paths in the graph. The algorithm starts with one vertex, calculates edge weights for paths going through that vertex, and then repeats it for every vertex in the graph and sums the weights for every edge.

For this algorithm we denote $Adj(v)$ as the set of all vertices adjacent to $v \in V$.

The first part of the algorithm for vertex marking:

```
1) For initial vertex $s \in V$ let $d_s = 0$, $w_s = 1$, $b_i = 0$.
2) Let $d_v = \inf$, $w_v = 0$, $b_v = 1$ for all $v \neq s \in V$.
3) Create queue $Q$, $Q \leftarrow \{s\}$. Create list $L$, $L \leftarrow \{s\}$.
4) While Q is not empty:
    a) Dequeue $i \leftarrow Q$.
    b) For each vertex $j \in Adj(i)$:
        i) If $d_j = \inf$ then $d_j = d_i + 1$, $w_j = w_i$. Enqueue $j \rightarrow Q$. Push $j \rightarrow L$.
        ii) If $d_j \neq \inf$ and $d_j = d_i + 1$ then $w_j += w_i$.
        iii) If $d_j \neq \inf$ and $d_j < d_i + 1$, do nothing.
```

Efficient implementation of this part of the algorithm could be done by using abstract data type queue.

The second part of the algorithm starts from the vertex that was last marked in the first part of the algorithm and visits vertices in reverse order than they were visited in the first part of the algorithm. Only one shortest path from source passes through the last marked vertex.

The second part of the algorithm for edge betweenness calculation:

```
1) While $L$ is not empty:
    a) Pop $i \leftarrow L$.
    b) For each vertex $j \in Adj(i)$:
        i) If $d_i < d_j$ then $b_i = 1 + \sum_j \sigma_{ij}$.
        ii) If $d_i > d_j$ then $\sigma_{ij} = \frac{w_j}{w_i} * b_i$.
```

Both parts of the algorithm are performed for all source vertices s and edge betweeness for every edge is calculated as a sum of the edge betweennesses calculated in every step.

Using this method it takes o(mn) time to calculate betweenness for one pair and O(m2n) or O(n3) time for recalculations, in worst-case on a sparse graph. While the other two methods take O(n3) time to calculate betweenness for one pair and O(n4) f or recalculations

Also a modified version of this algorithm has been published to reduce its complexity; it says that the leaf nodes can be removed from the graph without calculating betweenness, as the weightage for an edge connected to leaf will be minimum always. This algorithm also includes” recalculation”. In each cycle, first all the leaf nodes should be removed and the betweenness measure has to be recalculated.

The betweenness centrality of a node v is given by the expression:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \qquad (1)$$

where σ_{st} is the shortest path from node s to node t and $\sigma_{st}(v)$ is the number of paths that pass through v.

1) Scaling: Betweenness centrality of a node scales with the number of pairs of nodes as implied by the summation indices. Therefore, the calculation may be rescaled by dividing through by the number of pairs of nodes not including v, so that $g \epsilon [0, 1]$. The division is done by (N-1)(N-2) for directed graphs and (N-1)(N-2)/2 for undirected graphs, where N is the number of nodes in the giant component. This scales for the highest possible value, where one node is crossed by every single shortest path.

2) Applications: Betweenness centrality finds wide application in network theory. It represents the degree of which nodes stand between each other. To exemplify, in a telecommunications network, a node with higher betweenness centrality would have more control over the network, because more information will pass through that node. It applies to a wide range of problems in network theory, including problems related to social networks, biology, transport and scientific cooperation.

C. Modularity

Modularity measures the group interactions compared with the expected random connections in the group. In a network with m edges, for two nodes with degree di and dj, expected random connections between them are

$$d_i d_j / 2m \qquad (2)$$

The algorithms used for community structure detection in networks give an overview of possible communities. Understanding the network structure, it is possible to predict critical connections in the network, and therefore, control the network.

However, the question is when an optimal partitioning of a network into groups is reached. To support this, it is not necessary to decompose a network into communities of the size one. Qualitative measure of network decomposition is called modularity.

$$M_c = \sum_{c=1}^{n_c} [L_c/L - (K_c/2L)^2] \qquad (3)$$

where n_c = Number of communities, L_c = Total number of links in communities c L = Total number of links,k_c= Total node degrees in community c

Fraction of edges that fall into same community minus expected value of same quantity if edges fall at random without regard for community structure.

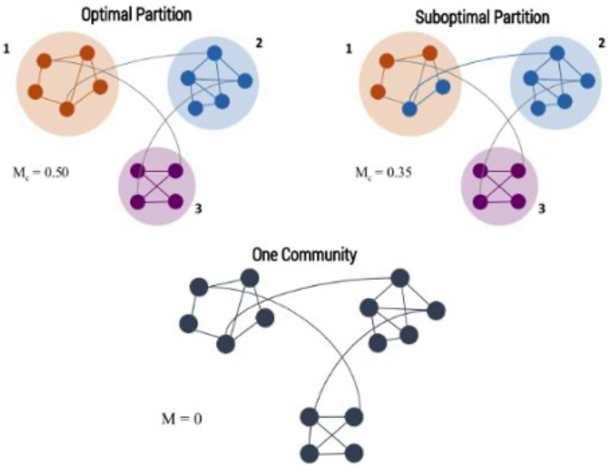


Fig. 8. Partitions with Modularity

The higher the value of modularity, the best is the partition of the network. Modularity always lies between [-1,1]. Modularity is 0 if all nodes are clustered into one group. This can automatically determine optimal number of clusters. Networks with high modularity implies dense connections between nodes within modules and sparse connections between nodes in different modules.

If modularity value lies between [0.3,0.7], it is considered as a significant community structure. Modularity is useful for selecting number of clusters.

The various techniques for modularity optimization are Greedy Techniques (hierarchial clustering), Simulated annealing, External Optimization, Spectral optimization. Another modularity maximization method is the Louvain Method in which communities are repeatedly optimized until global value of modularity is maximized. The method uses greedy optimization to form communities from large networks which has run time of O(nlogn). This method is more efficient for identifying communities in large networks. The method has been proved successful for networks of sizes up to million nodes and billions of links.

However, it has been found that modularity maximization suffers a resolution limit and fails to identify modules for small scale communities. This random approach assumes that each node in a network can get attached to any other node. This assumption is however not reasonable if the network is very large. For this reason, optimization of modularity in large network would fail to form small network communities.

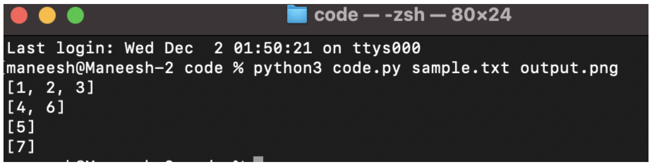
VI. Limitations

Despite the fact that it's been generally utilized, however it additionally has its own restrictions as it computes the betweenness very often for all the edges of the graph, in light of the fact that after expulsion of an edge with higher betweenness, betweenness of others likewise get influenced. Its run time complexity increments up to O(m2n) on a graph having m edges and n nodes. Because of it's high time complexity, it isn't advantageous to utilize this algorithm for large scale network, can be executed on the networks having less than three thousand hubs

Not time efficient for large number of nodes and data. It tends to give relatively poor results for dense networks. For community partition by betweenness, it's not possible to place an individual in two different communities and everyone is assigned to community. It provides no guide to how many communities a network should be split into i.e. where to cross-cut.

VII. Results

We first used the Girvan-Newman strategy on the very small dataset and it successfully executed the code and shown the results with grouping the small community. Below Image shows the output result.



```
Last login: Wed Dec  2 01:50:21 on ttys000
maneesh@Maneesh-2 code % python3 code.py sample.txt output.png
[1, 2, 3]
[4, 6]
[5]
[7]
```

Fig. 9. Results of tiny dataset

The below graph shows generated graphs for the communities. Colors are used to uniquely represent communities.

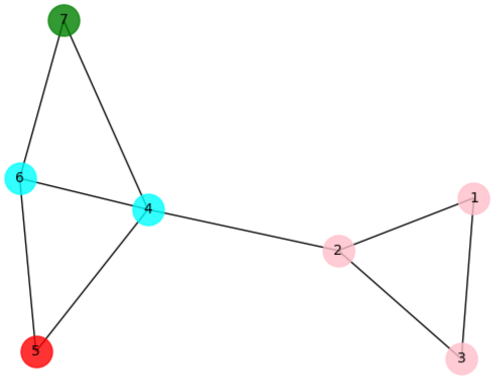
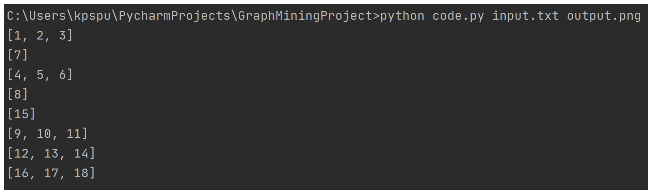


Fig. 10. Graphical Representation of tiny dataset

Later we moved on to the second stage by executing it on the small dataset with increased nodes and edges. This time it clearly shows the community detection with clear color representation and spacing. It shows successful execution with the single edge removal. Below Image shows the output result.

The below graph shows generated graphs for the communities. Colors are used to uniquely represent communities.

Now we used the Facebook mini dataset with the 200 nodes and 1000 edges. The Girvan-Newman algorithm takes around 10 mins to show the results and results are not quite impressive



```
C:\Users\kpspu\PycharmProjects\GraphMiningProject>python code.py input.txt output.png
[1, 2, 3]
[7]
[4, 5, 6]
[8]
[15]
[9, 10, 11]
[12, 13, 14]
[16, 17, 18]
```

Fig. 11. Results of small dataset

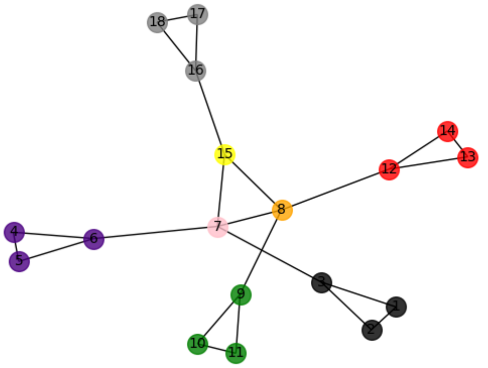
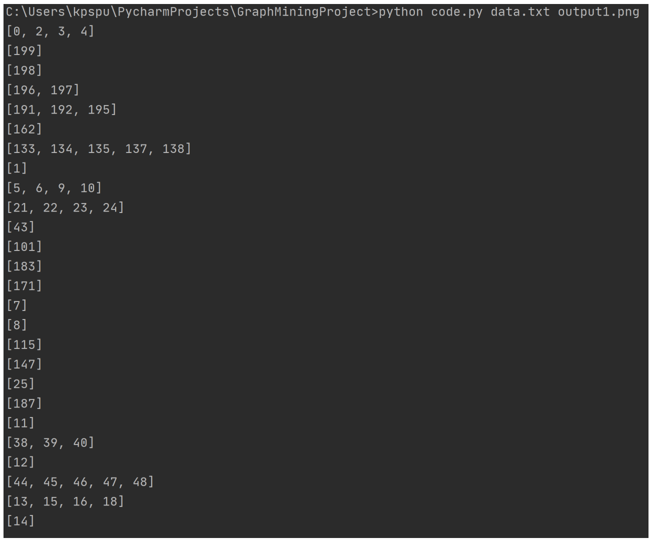


Fig. 12. Graphical Representation of small dataset

as all the nodes are overlapped and and community detection is not clearly visible. Below Image shows the output result.



```
C:\Users\kpspu\PycharmProjects\GraphMiningProject>python code.py data.txt output1.png
[0, 2, 3, 4]
[199]
[198]
[196, 197]
[191, 192, 195]
[162]
[133, 134, 135, 137, 138]
[1]
[5, 6, 9, 10]
[21, 22, 23, 24]
[43]
[101]
[183]
[171]
[7]
[8]
[115]
[147]
[25]
[187]
[11]
[38, 39, 40]
[12]
[44, 45, 46, 47, 48]
[13, 15, 16, 18]
[14]
```

Fig. 13. Results of the Facebook mini dataset

The below graph shows generated graphs for the communities. Colors are used to uniquely represent communities.

Finally, we used the large Facebook dataset. The code executed endlessly to delete multiple edges at a time. It was unsuccessful in showing the final results. Therefore we concluded that Grivan-Newman algorithm is not suitable for large and multi edge dataset.

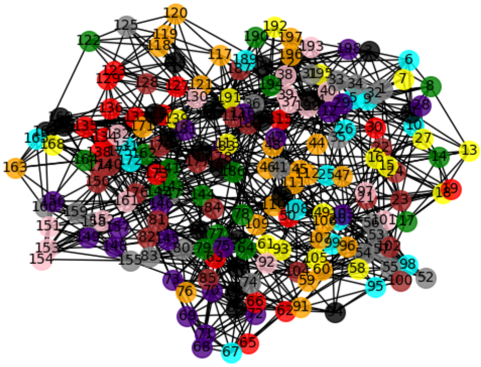


Fig. 14. Graphical Representation of Facebook mini dataset

VIII. Future Work

By improved Girvan-Newman strategy where multi edge evacuation is permitted. It diminishes number of activities by holding similar computational complexities.

At present O (n log n) is the fastest complexity which is utilized to locate an obscure number of networks. For examination of large networks it doesn't ensure that the networks discovered are the most ideal one or not. Different algorithms which are all the more computationally expensive have different benefits, for example, precision or the capacity to recognize ovelapping networks. Questions identified with the best appropriate technique actually stay open and furthermore to look for quicker and more exact strategy further investigation is proposed so more palatable outcomes would come.

IX. Conclusion

Community detection has developed quickly. It isn't abnormal that different community detection strategies have been created. Among all we discovered Girvan-Newman algorithm to be anything but difficult to execute for small networks. In the paper we depicted strategy for edge-betweenness removal and how to assess segments utilizing modularity.Though it is effective for small networks it can at present be improved for complexity and execution time for large networks.

References

[1] M. Girvan and M. E. J.2002. Newman, Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821–7826.

[2] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” Journal of Statistical Mechanics: Theory and Experiment, p. P10008, 2008.

[3] S. Fortunato, “Community detection in graphs,” Physics Reports, vol. 486, no. 3-5, pp. 75–174, 2010.

[4] D. Gibson, J. Kleinberg, and P. Raghavan, Inferring web communities from link topology. In Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, Association of Computing Machinery, New York (1998).

[5] Backstrom L, Huttenlocher D, Kleinberg J, Lan X (2006) Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 44-54

[6] M. Newman, “Finding community structure in networks using the eigenvectors of matrices,” Physical Review E, vol. 74, no. 3, p. 36104, 2006.

[7] M. E. J. Newman, Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality. Phys. Rev. E 64, 016132 (2001).

[8] M. Porter, J. Onnela, and P. Mucha, “Communities in networks,” Notices of the American Mathematical Society, vol. 56, no. 9, pp. 1082–1097, 2009.

[9] Traud A, Kelsic E, Mucha P, Porter M (2011) Comparing community structure to characteristics in online collegiate social networks. SIAM Rev 53:526-546

[10] Duch, J., and Arenas, A. (2005). Community detection in complex networks using extremal optimization. Physical Review E, 72(2), 027104.

[11] A. Lancichinetti and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” Physical Review E, vol. 78, no. 4, p. 046110, 2008.

[12] U. Brandes, “A faster algorithm for betweenness centrality,” Journal of Mathematical Sociology, vol. 25, no. 2, pp. 163– 177, 2001.

[13] S. Wasserman and K. Faust, Social Network Analysis. Cambridge University Press, Cambridge (1994).