

Detection of communities in Dynamic social networks

R.Sai Keerthana(17951A05E6)

Detection of communities in Dynamic social networks

A project Report

*Submitted in partial fulfillment of the
Requirements for the award of the degree of*

**Bachelor of Technology
In
Computer Science and Engineering**

By

R.Sai Keerthana

17951A05E6



Department of Computer Science and Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500043, Telangana

May, 2021

©2021, Sai Keerthana. All rights reserved.

DECLARATION

I certify that

- a. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute in preparing the report.
- d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place:

Date:

Signature of the Student

Sai Keerthana

17951A05E6

CERTIFICATE

This is to certify that the project report entitled **Detection of communities in dynamic social networks** submitted by **Ms. R.Sai keerthana** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree, Bachelor of Technology in **Computer Science and Engineering**, is a bonafide record of work carried out by her under my guidance and supervision. The substances of this report, in full or in parts, have not been submitted to some other Institute for the respect of any Degree.

Supervisor

Dr. M Madhu Bala

Head of the Department

Ms. Dr. G Sucharitha Reddy

Date:

APPROVAL SHEET

This project report entitled **Detection of communities in dynamic social networks** by **Ms.R.Sai Keerthana** is approved for the award of the Degree Bachelor of Technology in **Computer Science and Engineering**.

Examiners

Supervisor(s)

Principal

Date:

Place:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success. I thank our college management and respected Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal for providing me the necessary infrastructure to carry out the project work.

I express my sincere thanks to Dr. L.V. Narasimha Prasad, Professor and Principle who has been a great source of information for my work, and Dr. G Sucharitha Reddy, Professor and Head, Department of CSE, for extending her support to carry on this project work.

I especially thankful to our supervisor Dr. M Madhu Bala, Professor, Department of CSE, for her internal support and professionalism who helped me in shaping the project into successful one.

I take this opportunity to express my thanks to one and all directly or indirectly helped me in bringing this effort to present form.

ABSTRACT

Keywords: Density based clustering, DBSCAN, Community detection

In the present world, online social networks provide huge data that can include the objects information and comments which are analyzed and lead to discovering information and relationship among the networks. Finding community detection is an existing and attracting the researchers where they use different algorithms, one of them is mathematically based that work on connections in the community which has the dynamic networks as humans and also as the key players and the other one is the graph the structure which shows the output and it is similar to the topological structure these are being used from past years but these algorithms and structures have limitations we need to overcome these, in this research the outline of the project that we used is Density based clustering technique DBSCAN to detect communities. Detection and removal of these noisy nodes in the detected communities leads to the improvement of the quality.

\

CONTENTS

Title Page	I
Certificate	II
Approval Sheet	III
Declaration	IV
Acknowledgement	V
Abstract	VI
Contents	VII
List of Figures	IX
List of Abbreviations	XII

Chapter 1	Introduction		2
	1.1	Introduction	2
	1.2	Existing system	3
		1.2.1 Limitations	3
	1.3	Proposed system	3
		1.3.1 Advantages	4
	1.4	Problem Statement	4
		1.4.1 Objectives	4
		1.4.2 Applications	4
		1.4.3 Limitations	5
		1.4.4 Software Requirements	5
		1.4.5 Hardware Requirements	6
	1.5	Overview of the report	6
Chapter 2	Literature Survey		8
Chapter 3	Methodology		14
	3.1	Block Diagram of proposed system	15
	3.2	Network dataset	
	3.3	Network Visualization	

	3.4	Distance matrix / Edge Matrix	
	3.5	DBSCAN	
	3.6	Fast Greedy	
	3.7	Implementation	
		3.7.1	Implementation of DBSCAN
		3.7.2	Implementation of DBSCAN(Node removal) and Fast Greedy
		3.7.3	Implementation of DBSCAN(Edge removal) and Fast Greedy
		3.7.4	Importing Libraries
		3.7.5	Visualizing Network Data
		3.7.6	Distance Matrix
		3.7.7	Performing DBSCAN
		3.7.8	Node Removal
		3.7.9	Edge Removal
		3.7.10	Modularity
		3.7.11	Fast Greedy
		3.7.12	Visualizing the communities formed
	3.8	Community Visualization	
Chapter 4	Results and Discussions		
	4.1	Input data description	
	4.2	Epsilon value	
	4.3	Communities detected using DBSCAN	
	4.4	Communities detected using DBSCAN(Node removal) and Fast Greedy	
	4.5	Communities detected using DBSCAN(Edge removal) and Fast Greedy	
	4.6	Analysis of DBSCAN variants	
Chapter 5	Conclusions and Future Scope		
	5.1	Conclusion	
References			
Appendix			

--	--	--	--

LIST OF FIGURES

Figure	Title	Page
3.1	Block Diagram to detect communities using DBSCAN	15
3.2	DBSCAN clustering (minsamples=6, eps=3)	
3.3	Importing Libraries	
3.4	Network Data Visualization	
3.5	Distance Matrix	
3.6	Applying DBSCAN on data	
3.7	Removing Nodes from the Network	
3.8	Removing Edges from the Network	
3.9	Calculating modularity	
3.10	Applying Fast Greedy on the network	
3.11	Visualizing Communities Detected	
4.1	Distance Matrix Data	
4.2	Edge Matrix	
4.3	Optimal epsilon value for DBSCAN , DBSCAN(Node removal) and Fast greedy	
4.4	Optimal epsilon value for DBSCAN(Edge removal) and Fast greedy	
4.5	Communities Detected Using DBSCAN(eps=10,min samples=8)	
4.6	Communities Detected Using DBSCAN(eps=10,min samples=12)	
4.7	Communities Detected Using DBSCAN(eps=10,min samples=16)	
4.8	Analysis of DBSCAN	
4.9	Communities Detected Using DBSCAN(Node removal) and fast greedy(eps=10,min samples=8)	
4.10	Communities Detected Using DBSCAN(Node removal) and fast greedy(eps=10,min samples=12)	
4.11	Communities Detected Using DBSCAN(Node removal) and fast greedy(eps=10,min samples=16)	
4.12	Analysis of DBSCAN(node removal) and Fast Greedy	
4.13	Communities Detected Using DBSCAN(Edge removal) and fast greedy(eps=6,min samples=6)	
4.14	Communities Detected Using DBSCAN(Edge removal) and fast greedy(eps=6,min samples=13)	

4.15	Communities Detected Using DBSCAN(Edge removal) and fast greedy(eps=6,min samples=18)	
4.16	Analysis of DBSCAN(node removal) and Fast Greedy	
4.17	Analysis of DBSCAN Variants	

ABBREVIATIONS

DBSCAN	Density-based spatial clustering of applications with noise
eps	Epsilon
minSamples	Minimum Number of Points
BFS	Breadth First Search

CHAPTER 1

INTRODUCTION

1.1. Introduction

Social Networks and online communications between people have increased significantly and important part of a social network is its connections and these are some kind of relationship between the users. A Group of users who are more strongly connected to each other with other users in the network forms a community. Detecting such communities are hard and There are many general methods applied to detect communities there are several algorithms and approaches available to detect communities. One of the significant methods for community detection are Grivan newman algorithm also known as Edge Betweenness, Fastgreedy, Lable propagation, Louvain, Walktrap, Infomap.

None of These algorithms are able to identify the noise, As nodes are not the members of any a community so as to address the problem density-based community the detection algorithm is relevant since they provide the best to leave specious connected nodes such as noise, Out of the detected community.

DBSCAN algorithm has been taken for detecting the outliers and this detects the communities in a social network, and outliers are also known as “noisy nodes”, those are removed from the network graph. The main algorithm in the paper, Focuses on the detection and remove those noisy nodes in the detected communities which lead to the improvement of the quality.

The graph has well-established DBSCAN Algorithm where it is a density-based approach that could be applied to community detection. Similarly to DBSCAN, and has two parameters, as the density-based level ϵ and a lower bound Min Pts for the number of nodes that forms a community.

1.2. Existing System

There are many community detection algorithms that are appeared in the past, but only few of them are large scale algorithms that are applicable in social media graphs.

The Girvan–Newman algorithm detects communities by removing edges from the original network. These connected components of the remaining network are the communities. We need to construct a measure that tells us that the edges are the central communities, this algorithm mainly has edges that are likely "between" communities. Which has the edges that define "edge betweenness" of the edge which has the number of shortest paths between the pairs of nodes that will run in it. Fast greedy algorithm the problem-solving which make it as optimal choice at each stage. It has many problems and greedy strategy which does not produce optimal result. Label Propagation algorithm is which that allocate labels to the unlabelled nodes by propagating labels through the different kind of datasets.

The Louvain method for community detection is to extract communities from large social networks. This is an unsupervised algorithm and it does not require the input of the number of communities or size before execution. Walktrap is a ordered clustering algorithm. which has an idea of this method which has the short distance walk and likely will stay in the same community. Starting from a non-clustered partition the distances between the adjacent nodes are computed. Infomap algorithm reduce the cost that is based on the flow that was created by the pattern of connections in a given network.

1.2.1. Limitations

In the Existing systems we have observed the following deviations:

- Girvan Newman is not suitable for larger data sets.
- Fast greedy algorithm fail to get the optimal solution as they do not consider all the data.
- Label uses its information as clustering constraints.
- Communities may also be internally disconnected.

1.3. Proposed System

In our project, we have used DBSCAN algorithm as it is a best-unsupervised algorithm that is done to accentuate community detection in the social networks. The results specify that the large bias members by core, less bias by border and members with no influence in the groups represented by outliers. By removing the outliers the dataset will be noise-free.

Density-based clustering algorithm, has ability to extract the clusters without the prior

knowledge on number of clusters, also in the case where there is noise. The clustering is based on two parameters ϵ and minSamples , which are by the density level ϵ and a lower bound and the number of points in a minSamples .

DBSCAN (edge removal) and fast greedy, DBSCAN(Node removal) and fast greedy are the proposed algorithms these two algorithms are used to Detect communities efficiently.

Facebook data was collected from survey participants using the Facebook app. Where the dataset has node, circles, and network. Facebook data has been inspect by replacing the Facebook-internal identities for each user with a new value and while vectors from the dataset have been provided by the exposition. The data set consists of 4038 nodes and 88234 edges.

1.3.1. Advantages

In DBSCAN approach these are the advantages:

- DBSCAN is great while removing the clusters with high density vs clusters with low density in the dataset.
- Best while handling the outliers in the dataset.
- It does not require to specify the number of clusters in the data that opposed to k-means.
- It has a conception of noise, and is sturdy to outliers.
- Parameters like minSamples and ϵ can be done by the expert, if the data is known.

1.4. Problem Statement

Community detection is the important term in identify the compound structure of social networks, and it has great importance in sciences where systems are often shown as graphs. The graph is a representation of the real world network with the nodes shown as the entities, the relations between these entities are represented by edges. Community detection involves group of similar users into the clusters, where users in a group are strongly bonded with one another than the other members in the network.

1.4.1. Objectives

The objective of the proposed solution are as follows:

- To understand the facebook network data and preprocess the data.
- To detect communities in large networks using DBSCAN
- To visualize the communities detected using DBSCAN variants.

1.4.2. Applications

- Recommendation Systems:
The main task of this is community detection which is a kind of seperation of people which is like mind.
- public Health:
In community detection it is generally used to come accross the dynamics of groups and to suspect the epidemic disease.
- Politics:
It is used for the observation of political ideas and discrete politicians on some of the social groups.
- Criminology:
It is used to identify the criminal user group
- Community Evolution Prediction:
It is used for the prediction of future in a community that given its past and present in the community events.

1.4.3. Limitations

In the Existing systems we have observed the following deviations :

- Data sets with alternating densities are hard.
- Sensitive to clustering parameters minSamples and eps.
- Cannot find identify cluster in density varies and if dataset is too large.
- sampling affects the density measures.

1.4.4. Software requirements

- Programming Language:
 - Python 3.7
- Integrated Development Environment(IDE):
 - Kaggle
 - Jupyter Notebook
 - Python IDLE

- Packages required:
 - Sklearn
 - Scikit-learn is a ML library consists of different supervised and unsupervised algorithms.
 - Numpy
 - Numpy provides 50 times faster processing of arrays when compared to traditional array objects.
 - Matplotlib
 - Matplotlib is used for data visualization and graphically plotting the data.
 - Igraph
 - It is on the Python Index which is pre-compiled for most Python platforms, so in most cases it can simply be installed.
 - Networkx
 - It is a Python library used to study graphs and their networks.
 - Pandas
 - It is a software library used for the Python programming and for data manipulation and analysis.

1.4.5. Hardware Requirements

- Processor :
 - Intel core i5 processor
 - 8Gb DRAM
- Disk space :
 - 2 -3 Gb
- Operating system :
 - Windows 10

1.5. Overview of the Report

In Chapter 1, the report gives the brief description of the existing and the proposed system. Along with the description it provides the limitations of the previous models. Then it provides the information of the software and hardware requirements for the model to run.

In Chapter 2, we discuss about the journals studied which are relevant to the proposed model.

In Chapter 3, we start with the block diagram of the model and provide some description about it. Then the information regarding the algorithm used in the model is provided and they are described with the help of architecture of the model and figures. In addition, we provide the algorithm implementation with the flow and screenshots of code.

In Chapter 4 we discuss about results acquired by using all the implemented methods.

Finally in Chapter 5 we discuss about the future scope of the application and how it can be used in various fields.

Chapter 2

Literature Survey

We are going to represent literature overview of few papers that we have studied for choosing the topic as follows:

2.1. Community structure in networks using Girvan-Newman algorithm.

In this paper [1] the authors Ljiljana Despalatovic, Tanja Vojkovic & Damir Vukicevic provides the community detection by Girvan–Newman algorithm detects communities by removing edges from the original network. These connected components of the remaining network are the communities. We need to construct a measure that tells us that the edges are the central communities, this algorithm mainly has edges that are likely "between" communities. Which has the edges that define "edge betweenness" of the edge which has the number of shortest paths between the pairs of nodes that will run in the network.

2.2. Community detection using Fast Greedy algorithm

In this paper [2] the authors Bakillah, M., Li, R.-Y., & Liang, S. H. L. provides problem-solving which make it as optimal choice at each stage. It has many problems and greedy strategy which does not produce optimal result. It has a set in which a solution is generated and After the selection function it has the best candidate to be added into the solution, Then after the feasibility function it is used to determine either a candidate can be used to have a solution or not objective function assign values to the result. Result function will indicate when we get the entire solution. But for many other problems the Fast greedy algorithms fail to produce the best result, and may also produce the worst result.

2.3. Community detection using Label Propagation

In this paper the author Garza, S. E., & Schaeffer provides the allocation of the labels to the unlabelled nodes by propagating labels through the different kind of datasets. The

edge connecting two nodes has few similarities with the connection between other algorithms. Label propagation can have different community structures that have starting condition. The solutions are reduced when some nodes are given with preceding labels and while others are unlabelled. And these unlabelled nodes will be more likely to adopt the labelled ones. This algorithm has the labels of the already labelled nodes as their ground and they try to predict the labels of the unlabelled nodes.

2.4. Community Detection with the Louvain Algorithm

In this paper the author Que, X., Checconi, F., Petrini, F., & Gunnels, J. A. provides community detection is to extract communities from large social networks. This is an unsupervised algorithm and it does not require the input of the number of communities or size before execution and it is divided into two phases: Modularity Optimization, Community Aggregation. After the first step is done then the second follows later both will be executed until there are no more changes in the network and then the maximum modularity is achieved.

2.5. Walktrap algorithm for overlapping communities

In this paper the author Hu, F., Zhu, Y., Shi, Y., Cai, J., Chen, L., & Shen, S. provides Walktrap is a ordered clustering algorithm. which has an idea of this method which has the short distance walk and likely will stay in the same community. Starting from a non-clustered partition the distances between the adjacent nodes are computed.

2.6. Community Detection using Infomap

In this paper the author Yu-Liang, L., Jie, T., Hao, G., & Yu, W provides Infomap algorithm reduce the cost that is based on the flow that was created by the pattern of connections in a given network. Another way to choose the same path in a more incisive way is by Huffman coding approach. This approach also shows that the community finding algorithms can be also used to solve the compression problems and this approach also shows that the community finding algorithm can be also used to solve compression problems.

2.7. Detection of communities by Edge Betweenness

In this paper the author Seunghyeon Moon, Jae-Gil Lee, & Minseo Kang. provides the betweenness of the connected pairs are calculated for all edges in the network. The edges with high betweenness and low similarity between the connected pairs are known then these edges are removed from the network and the betweenness of the remaining edges are recalculated. This procedure is repeated until there is no more edge the proposed algorithm is validated in both synthetic and real-world network

CHAPTER 3

METHODOLOGY

Stanford Facebook survey network dataset is used in this paper. Initially the network is visualized. The edge data is converted into distance matrix and edge matrix. DBSCAN is performed on the matrix. The outliers are removed from the network. The algorithm Fast Greedy is applied on the network after removing outliers. Finally the communities formed are visualized. The communities are evaluated using modularity score.

3.1 Block diagram of proposed system

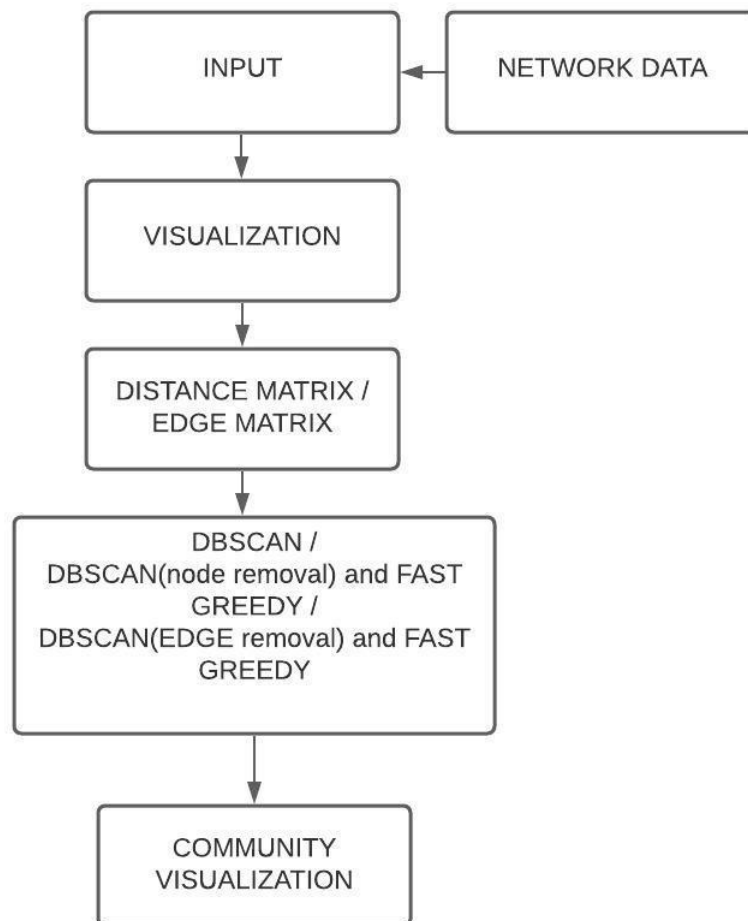


Fig.3.1.Block diagram for community detection

3.2 Network dataset

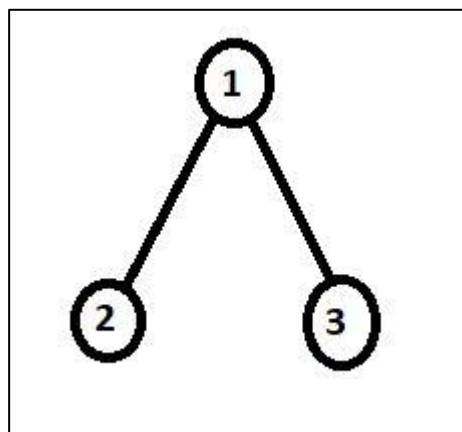
Facebook data was collected from survey participants using the Facebook app .This dataset consists of 4038 nodes and 88234 edges and dataset includes node features , circles and ego-networks, facebook-Ids have been replaced with a new value for each user.

3.3 Network Visualization

The network dataset is visualized as a graph. Visualization is done for better understanding of the data. This dataset has been visualized using igraph module.

3.4 Distance matrix / Edge matrix

The distance matrix is calculated using the BFS tree of the network. Every node in the network is considered as the starting node and BFS tree has been generated. Distances are calculated from each node to every other node.



Distance matrix of the above network-[[0,1,1],[1,0,1],[1,1,0]]

Edge Matrix - Edge matrix is calculated by loading the edge data into pandas data frame

3.5 DBSCAN

DBSCAN stands for Density-based spatial clustering of applications with noise (DBSCAN). A node belongs to a community if it is close to many nodes from that community. There are two key parameters of DBSCAN which help in identifying communities. Epsilon and minSamples. eps is the radius of the circle that specifies the neighbour. The condition to be considered to be the neighbour. The distance between them is less than or equal to

Epsilon. minSamples is the minimum number of data points to define a community. Based on where the points lie they are classified as core point, border point and outlier/noise.

Core point has atleast minSamples in its surrounding with radius eps. A Border Point has fewer than minSamples within its eps-neighbourhood , but it lies within the neighbourhood of another core point. Outlier is a point that it is not a core point and not reached by any other core points or border points.

Steps of DBSCAN Algorithm:

- The algorithm starts with an arbitrary Node or Edge which hasn't been visited then its neighbourhood information is rescue from the eps-parameter.
- If it contains minSamples within eps-neighbourhood, community formation starts. Otherwise the aim is labeled as noise.
- The above process continues until the density-connected cluster is totally found

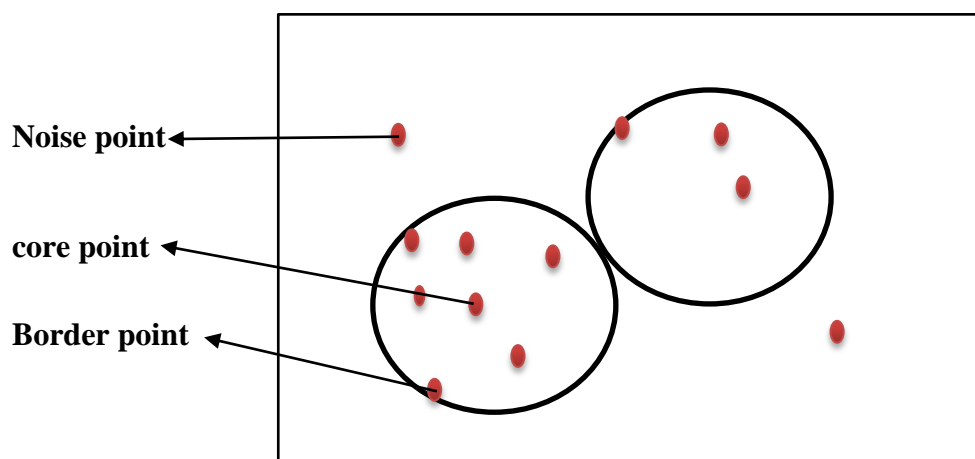


Fig3.2. DBSCAN clustering (minSamples=6, eps=3)

3.6 FAST GREEDY

Fast greedy algorithm the problem-solving which make it as optimal choice at each stage. It has many problems and greedy strategy which does not produce optimal result, Greedy may have optimal solutions that has best solution in a maximum amount of time. Fast

Fast Greedy algorithm:

- It has a set in which a solution is generated
- After the selection function it has the best candidate to be added into the solution

- Then after the feasibility function it is used to determine either a candidate can be used to have a solution or not
- objective function assign values to the result.
- Result function will indicate when we get the entire solution. But for many other problems the Fast greedy algorithms fail to produce the best result , and may also produce the worst result.

3.7 Implementation

3.7.1. Implementation of DBSCAN

1. Import numpy, pandas, sklearn, networkx, igraph, matplotlib packages.
2. load the dataset which consists of edge data of a graph.
3. Visualise the network using igraph package
4. Calculate the distance matrix of the network
5. Fit the distance matrix to DBSCAN algorithm
6. Visualise the detected communities using igraph

3.7.2. Implementation of DBSCAN (Node removal) and Fast Greedy

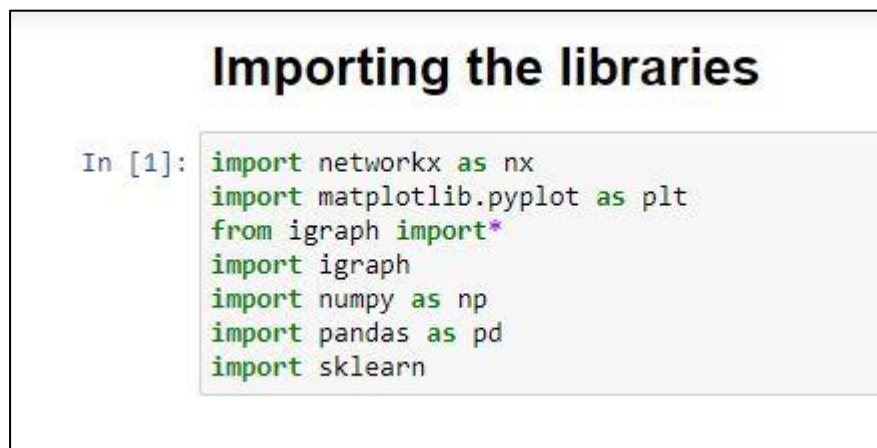
1. Import numpy, pandas, sklearn , networkx, igraph, matplotlib packages.
2. load the dataset which consists of edge data of a graph.
3. Visualize the network using igraph package
4. Calculate the distance matrix of the network
5. Fit the distance matrix to DBSCAN algorithm
6. Remove the nodes from the original network which are marked as noise by DBSCAN
7. Apply fastgreedy algorithm on the new network
8. Visualise the detected communities using igraph

3.7.3. Implementation of DBSCAN (Edge removal) and Fast Greedy

1. Import numpy, pandas, sklearn , networkx, igraph, matplotlib packages.
2. load the dataset which consists of edge data of a graph. Visualise the network using igraph package
3. Convert the dataset into pandas DataFrame
4. Fit the edge to DBSCAN algorithm
5. Remove the edges from the original network which are marked as noise by DBSCAN
6. Apply fastgreedy algorithm on the new network
7. Visualise the detected communities using igraph.

3.7.4 Importing Libraries

Here all the libraries required are imported. The Libraries used are numpy, pandas, networkx, igraph, sklearn, matplotlib.



```

In [1]: import networkx as nx
import matplotlib.pyplot as plt
from igraph import *
import igraph
import numpy as np
import pandas as pd
import sklearn

```

Fig3.3. Importing Libraries

3.7.5 Visualizing Network Data

Using Igraph python module for visualizing the initial network.



```

In [2]: f=Graph.Read_Edgelist(r"C:\Users\manik\Desktop\facebook_combined.txt",directed=None)
visual_style = {}
visual_style["vertex_size"] = 5
visual_style["margin"] = 17
visual_style["bbox"] = (250,250)
my_layout = f.layout_fruchterman_reingold()
visual_style["layout"] = my_layout
igraph.plot(f,r"C:\Users\manik\Desktop\graph plots\initial_network.png",**visual_style)

```

Fig.3.4 Network Data Visualization

3.7.6 Distance Matrix

Calculating BFS of the network from each node gives us the distance matrix

Calculating Distance Matrix

```
In [3]: G=nx.read_gml(r'C:\Users\manik\Desktop\football.gml')
dist_mat=[]
for i in G.nodes():
    t=nx.bfs_tree(G,i)
    k=nx.shortest_path_length(G,i)
    dist_mat.append(list(k.values()))
```

Fig.3.5 Distance Matrix

3.7.7 Performing DBSCAN

DBSCAN is applied on the given data

```
In [3]: a=[]
b=[]
for i in f.get_edgelist():
    a.append(i[0])
    b.append(i[1])
```

```
In [4]: x=pd.DataFrame()
```

```
In [5]: x['A']=a
x['B']=b
```

```
In [6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Using KNN to find out the optimal epsilon value. The optimal value for epsilon will be found at the point of maximum curvature.

```
In [7]: from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(x)
distances, indices = nbrs.kneighbors(x)
```

```
In [8]: distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.figure(figsize=(3,3))
```

Fig.3.6 Applying DBSCAN on data

3.7.8 Node Removal

In this step the nodes are removed from the original network

Removing Nodes

```
In [54]: import networkx as nx
g=nx.read_edgelist(r"C:\Users\manik\Desktop\facebook_combined.txt")
g1=nx.read_edgelist(r"C:\Users\manik\Desktop\facebook_combined.txt")
c=0
k=[]
for i in g1.nodes():
    if labels[c]==-1:
        g.remove_node(i)
        k.append(i)
    c+=1
a=[]
b=[]
for i in g.edges():
    a.append(i[0])
    b.append(i[1])
y=0
for i in g1.edges():
    if i[0] in k or i[1] in k:
        y+=1
```

Fig.3.7 Removing Nodes from the Network

3.7.9 Edge Removal

In this step the edges are removed from the original network

Removing all the outliers i.e; edges in the network

```
In [74]: j=0
for node in f.get_edgelist():
    if labels[j]==-1:
        f.delete_edges(f.get_eid(node[0],node[1]))
    j+=1

In [91]: new_labels = np.delete(labels, np.where(labels == -1))

In [92]: len(f.get_edgelist())

In [93]: a=[]
b=[]
for i in f.get_edgelist():
    a.append(i[0])
    b.append(i[1])
print(len(f.get_edgelist()))
Y=pd.DataFrame()

In [48]: Y['A']=a
Y['B']=b
```

Fig.3.8 Removing Edges in the network

3.7.10 Modularity

```

In [56]: import scipy as sp
         A = nx.adjacency_matrix(g)
         res = np.delete(labels, np.where(labels == -1))

In [57]: #modularity in dbscan
         from sknetwork.clustering import modularity
         from sknetwork.data import house
         import numpy as np
         np.round(modularity(A,res),2)

```

Fig.3.9 Calculating modularity

3.7.11 Fast Greedy

Fast greedy algorithm is applied on the network data after removing edges and nodes

Applying fast greedy to detect communities

```

In [77]: clusters=f.community_fastgreedy()
         print(clusters)

Dendrogram, 4039 elements, 1842 merges

In [78]: clusters=clusters.as_clustering()

In [90]: print(clusters)

In [89]: clusters.modularity

In [88]: clusters.membership

In [83]: pal=igraph.drawing.colors.ClusterColoringPalette(len(clusters))

In [84]: f.vs['color']=pal.get_many(clusters.membership)

```

Fig.3.10 Applying Fast Greedy on the network

3.7.12 Visualizing the communities formed.

detected clusters

```

In [85]: visual_style = {}
         visual_style["vertex_size"] = 5
         visual_style["margin"] = 17
         visual_style["bbox"] = (250,250)
         my_layout =f.layout_fruchterman_reingold()
         visual_style["layout"] = my_layout

In [87]: graph.plot(f,r"C:\Users\manik\Desktop\graph plots\edge removal minsamples=18.png",**visual_style)

```

Fig.3.11 Visualizing Communities Detected

3.8 Community Visualization

After applying DBSCAN on the network data, communities are visualized. Visualization shows which DBSCAN variant was able to detect communities efficiently.

CHAPTER 4

RESULTS

4.1. Input data description

Facebook data was collected from survey participants using the Facebook app .This dataset consists of 4038 nodes and 88234 edges and dataset includes node features , circles and ego-networks, facebook-Ids have been replaced with a new value for each user.

	Unnamed: 0	0	1	2	3	4	5	6	7	8	...	4029	4030	4031	4032	4033	4034	4035	4036	4037	4038
0	0	0	1	1	1	1	1	1	1	1	...	6	6	6	6	6	6	6	6	6	6
1	1	0	1	1	1	1	1	1	1	1	...	7	7	7	7	7	7	7	7	7	7
2	2	0	1	1	1	1	1	1	1	1	...	7	7	7	7	7	7	7	7	7	7
3	3	0	1	1	1	1	1	1	1	1	...	7	7	7	7	7	7	7	7	7	7
4	4	0	1	1	1	1	1	1	1	1	...	7	7	7	7	7	7	7	7	7	7

Fig.4.1 Distance Matrix

node	edges
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
0	10
0	11
0	12
0	13

Fig.4.2 Edge Matrix

4.2. Epsilon Value

The optimal epsilon value will be found at the maximum point of curvature

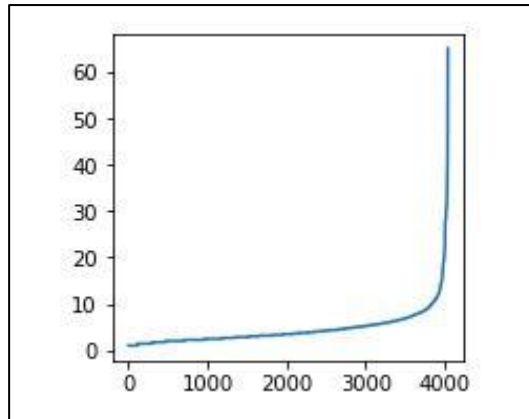


Fig.4.3 Optimal epsilon value for DBSCAN , DBSCAN(Node removal) and Fast greedy

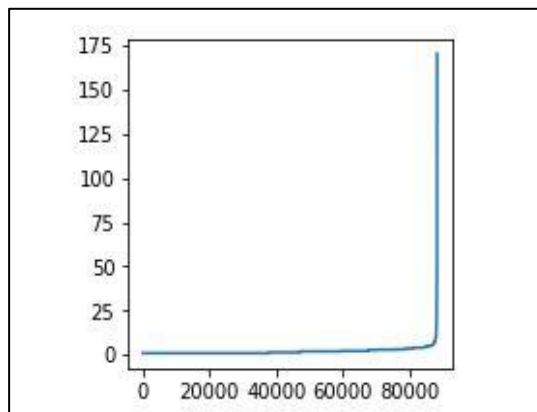


Fig.4.4 Optimal epsilon value for DBSCAN(Edge removal) and Fast greedy

4.3. Communities detected using DBSCAN

The red colour nodes are marked as Noise by DBSCAN. The data was evaluated for different Minsamples. These are the communities detected by DBSCAN

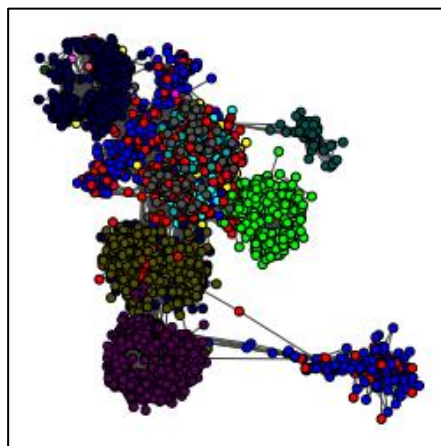


Fig.4.5 Communities Detected Using DBSCAN(eps=10,minSamples=8)

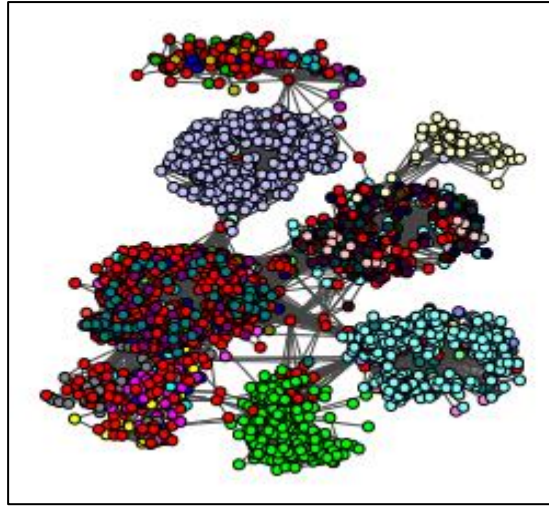


Fig.4.6 Communities Detected Using DBSCAN($\text{eps}=10, \text{minSamples}=12$)

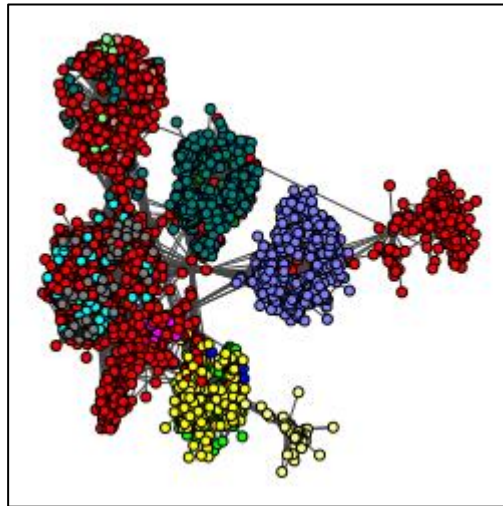


Fig.4.7 Communities Detected Using DBSCAN($\text{eps}=10, \text{minSamples}=16$)

DBSCAN eps=10	Clusters	Modularity	Noise Points/out liers	Edges Removed	Nodes Removed
minSamples=8)	26	0.58	603	0	0
minSamples=12	45	0.47	1105	0	0
minSamples=16	17	0.46	2144	0	0

Fig.4.8 Analysis of DBSCAN

4.4. Communities detected using DBSCAN (Node removal) and Fast Greedy

In this approach the communities detected are very large although they have a good modularity score

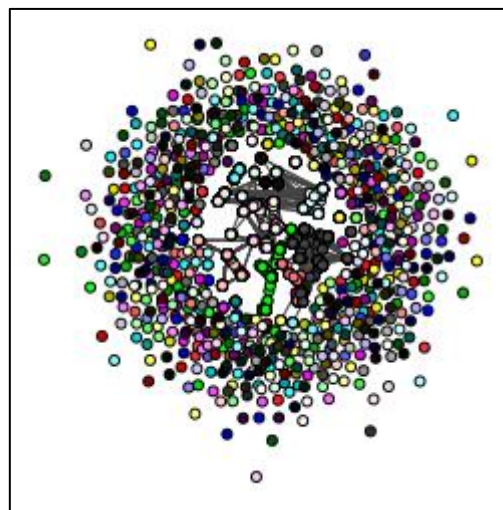


Fig.4.9 Communities Detected Using DBSCAN(Node removal) and fast greedy (eps=10,minSamples=8)

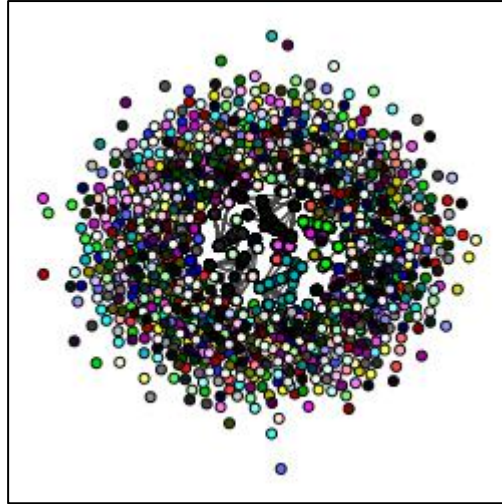


Fig.4.10 Communities Detected Using DBSCAN(Node removal) and fast greedy (eps=10,minSamples=12)

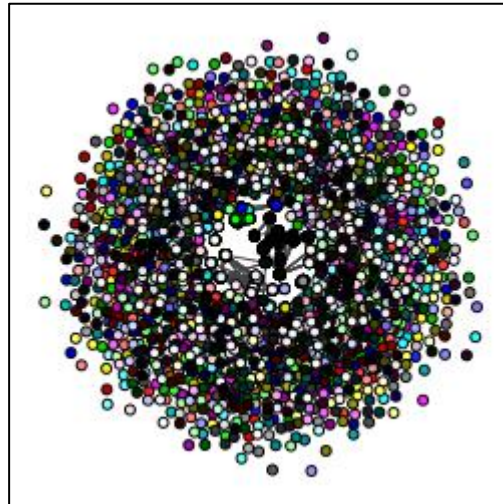


Fig.4.11 Communities Detected Using DBSCAN(Node removal) and fast greedy (eps=10,minSamples=16)

Dbscan(Node Removal) and fast greedy eps=10	Clusters	Modularity	Noise Points/outliers	Edges Removed	Nodes Removed
minSamples=8	757	0.83	603	33,308	603
minSamples=12	1243	0.83	1105	53240	1105
minSamples=16	2284	0.77	2144	68845	2144

Fig.4.12 Analysis of DBSCAN(node removal) and Fast Greedy

4.5. Communities detected using DBSCAN (Edge removal) and Fast Greedy

The communities detected are clearly visible in this approach also the formed communities have a good modularity score

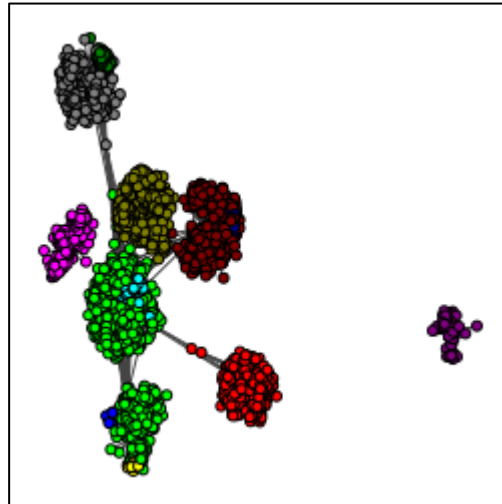


Fig.4.13 Communities Detected Using DBSCAN(Edge removal) and fast greedy (eps=6,minSamples=6)

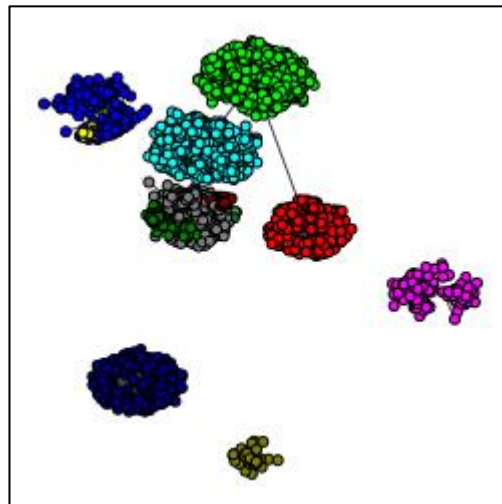


Fig.4.14 Communities Detected Using DBSCAN(Edge removal) and fast greedy (eps=6,minSamples=13)

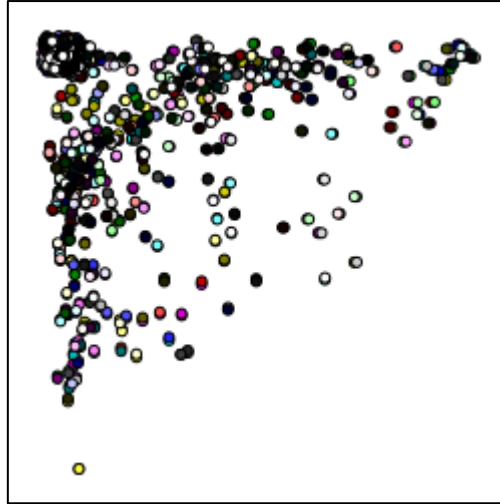


Fig.4.15 Communities Detected Using DBSCAN(Edge removal) and fast greedy (eps=6,minSamples=18)

DBSCAN(edge removal) and Fast Greedy eps=6	Clusters	Modularity	Noise Points/outliers	Edges Removed	Nodes Removed
minSamples=6	12	0.80	8,507	8507	0
minsamples=13	11	0.76	43,315	43,315	0
minSamples=18	2202	0.68	62194	62194	0

Fig.4.16 Analysis of DBSCAN (Edge Removal)and Fast Greedy

4.6. Analysis of DBSCAN variants

There is no ground truth data available for the dataset. We used modularity score to evaluate how well the clusters are formed. Modularity is a measure of the structure of networks or graphs which measures the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules.

Algorithm	Clusters	Modularity	Noise Points/outliers	Edges Removed	Nodes Removed
Dbscan(eps=10,minSamples=8)	26	0.58	603	0	0
Dbscan(Node Removal) and Fast Greedy (eps=10,minSamples=8) (Proposed)	757	0.83	603	33,308	603
Dbscan(edge removal) and Fast Greedy (eps=6,minSamples=13) (proposed)	11	0.76	43,315	43,315	0

Fig.4.17 Analysis of DBSCAN variants

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion and Future Scope:

This project proposed a novel community detection using different DBSCAN approaches with Fast Greedy. Community detection of facebook-ego network is successfully performed using different DBSCAN approaches with Fast Greedy. Our approach was able to detect communities quickly and efficiently. We were able to detect communities in the network with a good modularity value. The proposed models were able to detect communities in complex networks efficiently.

Further optimizations in the code can achieve better results. We expect that our model can achieve better results in sparse networks and networks which do not have dense connections.

References

- [1] Despalatovic, L., Vojkovic, T., & Vukicevic, D. (2014). *Community structure in networks: Girvan-Newman algorithm improvement*. 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). doi:10.1109/mipro.2014.6859714
- [2] Bakillah, M., Li, R.-Y., & Liang, S. H. L. (2014). *Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan*. *International Journal of Geographical Information Science*, 29(2), 258–279. doi:10.1080/13658816.2014.964247
- S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, “Community detection in social media,” *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.
- [3] Garza, S. E., & Schaeffer, S. E. (2019). *Community detection with the Label Propagation Algorithm: A survey*. *Physica A: Statistical Mechanics and Its Applications*, 122058. doi:10.1016/j.physa.2019.122058
- [4] Que, X., Checconi, F., Petrini, F., & Gunnels, J. A. (2015). *Scalable Community Detection with the Louvain Algorithm*. 2015 IEEE International Parallel and Distributed Processing Symposium. doi:10.1109/ipdps.2015.59
- [5] Seunghyeon Moon, Jae-Gil Lee, & Minseo Kang. (2014). Scalable community detection from networks by computing edge betweenness on MapReduce. 2014 International Conference on Big Data and Smart Computing (BIGCOMP). doi:10.1109/bigcomp.2014.6741425
- [6] Yu-Liang, L., Jie, T., Hao, G., & Yu, W. (2012). Infomap Based Community Detection in Weibo Following Graph. 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control. doi:10.1109/imccc.2012.286.
- [7] Hu, F., Zhu, Y., Shi, Y., Cai, J., Chen, L., & Shen, S. (2017). An algorithm Walktrap-SPM for detecting overlapping community structure. *International Journal of Modern Physics B*, 31(15), 1750121. doi:10.1142/s0217979217501211

Appendix

1	Name of the student	R.Sai Keerthana			
2	Email ID and Phone Number	saikeerthanasonu2000@gmail.com			
3	Roll Number	17951A05E6			
4	Date of submission				
5	Name of the Guide	Madhubala			
6	Title of the project work	Detection of Communities in Dynamic social networks			
7	Department	Computer Science and Engineering			
8	Details of the payment				
9	No. of times submitted	First / Second / Third (First time–Free; Second time–Rs200/-; Third–Rs500/-; There after multiple of third)			
10	Similarity Content (%) (up to 25% acceptable)	1st	2nd	3rd	4th
For R & D Centre Use					
Date of Verification					
Similarity report percentage					
R & D Staff Name and Signature					
<p>I / We hereby declare that, the above mentioned research work is original & it doesn't contain any plagiarized contents. The similarity index of this research work is Justification for similarity index:</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>					
Signature of Student		Signature of the Guide			