

Community Search from Multi-Attributed Large Social Graph

by

Riasat Islam Khan
14101156

Sayed Mahmud Khan
16301023

Tanmoy Debnath
16201008

Md. Nazmul Islam
19241026

Muhtasim Ibne Kayes
17201068

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

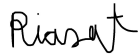
© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing the degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all the main sources of help.

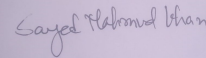
Student's Full Name & Signature:



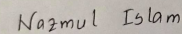
Riasat Islam Khan
14101156



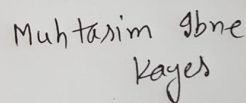
Tanmoy Debnath
16201008



Sayed Mahmud Khan
16301023



Md. Nazmul Islam
19241026



Muhtasim Ibne Kayes
17201068

Approval


The thesis titled “Community Search from Multi-Attributed Large Social Graph” submitted by

1. Riasat Islam Khan (14101156)
2. Sayed Mahmud Khan (16301023)
3. Tanmoy Debnath (16201008)
4. Md. Nazmul Islam (19241026)
5. Muhtasim Ibne Kayes (17201068)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2021.

Examining Committee:

Supervisor:
(Member)



15/01/2021

Mostafijur Rahman Akhond
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)



Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor
Department of Computer Science and Engineering
Brac University

With the constant evolvement of social network structure, complex data, as well as graph structure, has been growing with increasing importance to model the inter-connection of various entities. Community spot is a method of detecting densely connected sub-graph within a large network, for the given set of query vertex in the graph. It has many uses in social networking for instance when a certain user wants to get connected with other people or groups that go with the personality the user possesses. The main purpose of this method is to plot a well-organized mechanism to track the most dominant nodes as well as the corresponding meaningful communities that the vertex belongs to in an online manner.

The multi attributed graph contains the data and statistics as the properties of the nodes as well as the probable relationship among the nodes. These details are used to ensure accuracy and to figure out the target community. The present-day methods of working do not have enough features to allow the attributes or keyword information associated with a given edge for searching for the desired community. We have worked on developing a new multi attributed community search algorithm that takes all the attributes of the edge into account and uses modern weighted search algorithms to find communities for given nodes. These explored nodes are densely connected and share a lot of common features. Our study was conducted in two phases. In the first place, a weight was assigned to each of the attributes matching up their significance. Then an algorithm was applied to the weighted decision matrix to form a single-attributed graph from the initial multi-attributed graph. A sub-graph with the least required weight assigned as the community weight was used to get a strongly connected community that the query vertex belongs to. Our system was built using the tools and built-in libraries of Python programming language. Thus our experimental procedure was used in searching for communities from given data that resembles the real world more closely.

Dedication

To our supporting faculty body, seniors and well-wisher of the department and beyond. Love goes out to our friends and family for giving us the latent energy we always needed to get through this.

Acknowledgement

Throughout our incredible one year long journey, we have received uncountable assistance and contribution from many well-wishers. The journey would have been incomplete and vague without their constant support and priceless contribution. We want to express our wholehearted gratitude to our thesis supervisor Mostafijur Rahman Akhond Sir, his persistent motivation, guidance and expertise fueled our research progress.

Additionally, we are grateful to those students of CSE department, BRAC University who have helped us to collect the data which we used for our research work. Alongside, we are thankful to the people who are trying to make e-Learning easier, better and also fruitful by making different categories of online video tutorials.

Nonetheless, we would like to show appreciation to the Department of Computer Science and Engineering, BRAC University for providing us with all the fundamental help.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Dedication	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vi
1 Introduction	1
1.1 Motivation	2
1.2 Objective	2
2 Literature Review	3
3 Problem Definition	7
3.1 Problem Definition:	7
3.2 Algorithms	9
4 Experiment	14
5 Application	22
6 Future Works and Conclusion	24
Bibliography	26

List of Figures

2.1	A sample graph $G\{a,b,c,d,e,f,g,h,i,j,k,l,m,n\}$	3
3.1	Maximal edges	8
3.2	Support of an edge	8
4.1	A sample graph from youtube	15
4.2	A sample graph from nsa	16
4.3	Youtube density for distance constraint	17
4.4	NSA density for distance constraint	17
4.5	Youtube community size for distance constraint	18
4.6	NSA community size for distance constraint	19
4.7	Youtube nodes visited for distance constraint	19
4.8	NSA nodes visited for distance constraint	20
4.9	Youtube distance constraint executing time	20
4.10	NSA distance constraint executing time	21

Chapter 1

Introduction

Community is a social subdivision with similarity in standards, morals, and conventions in a particular neighboring area or a virtual space over the communal networking platform we use in our day to day life. Community on a different note can be considered as a subdivision of nodes inside a graph such that nodes are connected densely than the other connections with the rest of the network. Community search on a graph targets to discover the densely connected group that accommodating query nodes. Due to the recent developments of gigantic social networks like Flickr, Facebook, and Twitter the topic of finding associated communities using multi-attributed graphs has attracted attention from industry and research communities. Those real-life composite tracteries contain a community of varied structures. As a consequence searching for connected communities is becoming an important systematic task with a growing necessity to be incorporated into a networking system. Our applied research based on searching for a perfect community using a multi-attributed graph has an aim to develop a technique that will have the best accuracy level. To know at length on multi-attributed graphs and the existing technology on community search the existing kinds of literature have been reviewed and the gaps in understanding the areas that have not been addressed in these studies have been figured out. There are some well-defined benchmarks like k-core for the minimum degree, k-truss, and K-edge connected, etc to formulate the cohesiveness of a certain community in network science.

At this moment the question is, how to discover these communities. In our thesis, a community search on a particular multi-attributed graph has been used to find a suitable group according to our exploration need. Our applied research based on searching for a perfect community using a multi-attributed graph has focused on developing a technique with the best accuracy level. Our study is mainly based on the expansion of scientific understanding for seeking a precise conclusion about this searching issue.

Data collection and analysis is an indispensable part of this design. Qualitative research has been conducted to deal with statistics and at the same time with words and meanings. Our assumption has been tested on a large graph and hypotheses have been generalized. This whole job is based on testing the data, measuring the data, ranking, and categorizing the data to seek objectivity. To evaluate the quality of research, to indicate how well a method or testing measure the main focus was

on consistency and accuracy of action. The result has been assessed by checking the accuracy of our result across time and different observations.

1.1 Motivation

To know at length on multi-attributed graphs and the existing technology on community search the existing literature has been reviewed and the gaps in understanding the areas that have not been addressed in these studies have been identified. Preliminary research has been conducted to read and gather information, developing ideas on our current position to complete our research to fix the key issues and controversies that are surrounding these investigations, and finally to come up with the possible solutions to the issues that have not been put into consideration in the past.

In recent days community detection that aims to find the most likely community containing nodes with varied attributes has become an elementary complication in the network field because of the tremendous studies on big data. Practically communities for various vertices have a lot of features that's why we need to add these hallmarks into consideration when putting them in a graph and do the searching methodology locally. Additionally, this searching function to accommodate queries that users may specify in the future has been added. In our searching method, the main focus is on sub-graphs which are more efficient as detecting every possible community from an entire social network is not mostly necessary. For instance, to put recommendations in social media we have focused on those communities people are really interested in rather than the entire community set.

1.2 Objective

With the rapid expansion of information technologies in the network community, Social media-based applications have many relevant substantial graphs that are densely connected internally. In our algorithm, the foremost purpose is community retrieval to aid the users in their favored ways to be connected. Another main goal is to come up with a well-planned way out for finding highly connected communities from large networks on the online basis. Also to work on various sort of graphs and formulate the communities and perform with the best possible outcomes. Additionally one of the major criteria of the real-life graph is, they evolve frequently and in the global search method sensitivity in updating nodes and rims of the graph is not active enough so in community search we want to handle this formed on query requests.

Chapter 2

Literature Review

Community Detection could be a fundamental problem of any network science that refers to the finding of communities in any graph information. It is not subordinate to a query hub. Community detection is additionally known as community discovery for the most part takes a graph and breaks down it into a few sub-graphs based on certain goodness measures as characterized by the algorithm. In other words, the issue of deconstructing a graph into littler graphs where the nodes in any littler graph are more related to each other than nodes in other sub-graphs. The nature and strength of that relationship are determined by the goodness measure [1]. Community search, on the other hand, could be a smaller field of community detection that depends on a query node and tries to discover the communities that the query node belongs to. It is a more interior out approach compared to community detection. Community search starts with a query node from the graph. It for the most part looks for incident edges and adjacent nodes based. It usually navigates the graph in a recursive way until a certain threshold as characterized by the algorithm is reached and in this way returns a sub-graph which is the community of the query node [2].

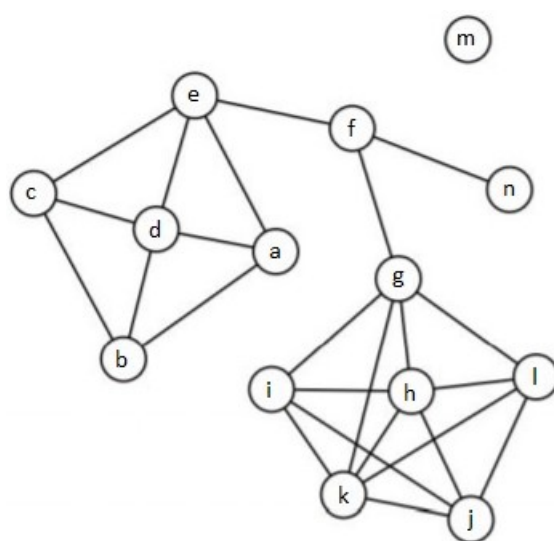


Figure 2.1: A sample graph $G\{a,b,c,d,e,f,g,h,i,j,k,l,m,n\}$

The global search method was proposed by Mauro Sozio and Aristides Gionis in

their popular The Community-search Problem and How to Plan a Effective Cocktail Party paper published in 2010. They were moreover the proponents of the Community Search issue which they characterized as “query-dependent variation of the community detection problem”. The non-specific thought proposed by Gionis and Sozio was to require a query node to emphasize the complete input graph and find the most excellent conceivable sub-graph based on a goodness degree. Their paper centered on least degree as their goodness degree [3]. They came up with two algorithms to solve the issue, GreedyFast, and GreedyDist. GreedyFast was outlined to be productive and it yielded quality while GreedyDist centered on the quality of the subgraph relinquishing effectiveness.

The local search method for community search was introduced by Wanyun Cui, Yanghua Xiao, Haixun Wang, Wei Wang in their paper Local Search of Communities in Large Graphs in 2014. It was defined by them as a search algorithm that searches the neighborhood of the query vertex instead of all the vertices of the whole graph.[3]. To find the most excellent algorithms to conduct a local search the group partitioned the problem into two particular and separate but not irrelevant issues. They called one of these Community Search with Maximality Limitation and the other Community Search with Edge Constraint [3]. The thought behind these two problems stems from two practical uses of community search. In case an algorithm does not require to find the finest conceivable sub-graph of a graph for a particular query vertex but is fulfilled with a specific least degree at that point it saves a lot of computation and time when CST is connected instead of CSM. The goodness measure of a community or a specific graph is the criteria upon which the nature and strength of the connection of that network of different nodes are judged. There are a few methods by which we are able to measure the goodness of our framework or the system. In order to find densely connected communities that contain the query nodes, one must define an appropriate measure of the density of those query nodes. This kind of measure can be the average or the minimum degree of the nodes in the extracted community. we focus on the last-mentioned measure, the minimum degree. We also allow the possibility to exclude nodes that are distant from the query nodes, usually, these nodes are less related to the query nodes than those that are adjacent. Our objective in this paper is to discover compact communities, containing the query nodes, and whose minimum degree is maximized. It is the summation of all their interactions that coalesce in the perception of their relationship along with your brand. More particularly, believe estimation could be a way of giving a result an exact measurement, generally by comparison to some standard or pattern, and usually is done in a quantifiable or numerical way. According to the appearance of node behavior and historical data, adjust and predict the trusted-order of node behavior. We are going to try to establish dynamic trust assessment model-based node behavior characters, and after that, it discusses the trusted measurement strategy which measures the connection and hyperlink for node behavior of the network in trust relationship space. A trust relationship can be communicated with a graph. Within the paper, we think about a quantitative expression of trust relationship within the organized network system by utilizing the strategy of graph theory. At that point, it measures the trusted degree of each node, and it moreover presents the trusted measurement of the connection and hyper connection for node behaviour of the network. The main object of this paper is to build up a dynamic trust evaluation model which will be based on node behaviour characters, Through

the development of the relationship between practical node behaviour characters and on-the-spot model, it sets up some of the mapping models of the trust relationship and draws the skeleton of relationship mapping inversion. Organization search is a significant issue in relational association mining, which has pulled in much consideration lately.. In [4] Zibin Zhenga, Fanghua Yea, Rong-Hua Li, Guohui Lingc, Tan Jind(July,2017) proposed a novel local area model, called weighted k-bracket local area, in view of the idea of k-support.The k-bracket is characterized as the maximal subgraph with the end goal that each edge is contained in at any rate $(k-2)$ triangles in the subgraph . The proposed model in [4] takes the edge weight into thought, accordingly can more readily portray the properties of a local area . In [4] they lead broad trials on six huge genuine organizations, just as a contextual investigation over a co-creation organization. The outcomes exhibit the proficiency and viability of the proposed local area model and calculations.

Local area location which finds thickly associated structures in an organization has been concentrated a ton. Given an inquiry vertex in a diagram, the issue is to discover important networks that the vertex has a place with in an online way. In [5] Xin Huang†, Hong Cheng†, Lu Qin‡, Wentao Tian†, Jeffrey Xu Yu† proposed a novel community model based on the k-truss concept, which brings nice structural and computational properties. In [5] they design a compact and elegant index structure which supports the efficient search of k-truss communities with a linear cost with respect to the community size. In [6]they explored the ktruss local area search issue in a dynamic chart setting with successive additions and erasures of diagram vertices and edges. Given a diagram G , the k-support of G is the biggest subgraph in which each edge is contained in any event $(k - 2)$ triangles inside the subgraph. On top of the k-bracket, they forced an edge network requirement, that is, any two edges locally either have a place with a similar triangle, or are reachable from one another through a progression of adjoining triangles. Here two triangles are characterized as adjoining on the off chance that they share a typical edge. The edge availability prerequisite guarantees that a found local area is associated and firm. The way to proficient ktruss local area inquiry preparing is to plan a powerful file . In [5] they study k-truss community search in dynamic graphs, where graph vertices and edges can be frequently inserted or deleted. In [5] authors evaluated the quality of the discovered communities on two social networks with ground truth communities and a scientific collaboration network. In [7] Muhammad Abulaish Jahiruddin (2018)introduced a novel weighted distance measure dependent on weighted Euclidean standard which is characterized as a component of both vertex and edge credits, and it very well may be utilized for different chart investigation errands including order and group examination. The diagram could be weighted/unweighted or coordinated/undirected. To display complex information, the idea of multi credited diagram can be utilized wherein every vertex is spoken to by a n-dimensional vector and there might be various weighted edges between each pair of vertices. Bunch decay a given diagram into various strong sub-charts dependent on some basic properties. It is expected to o recognize the basic structure (as groups) of information, which is by and large dependent on some closeness/distance measures between information components. Group partitions an information diagram into various associated parts (sub-charts) with the end goal that intra-segment edges are greatest and between segments edges are least. Each associated segment is known

as a group. The weighted likeness diagram can be broke down utilizing existing arrangement and bunching calculations for shifted purposes. The effectiveness of the proposed distance measure and calculations is tried over the notable Iris informational collection .

In [6] Kifayat Ullah Khan, Tu Nguyen Anh, Mostafijur Rahman Akhond, Waqas Nawaz, Young-Koo Lee given another idea of performing local area search on a dedensified chart. In [8] their point is to sparsify the problem areas to quicken worldwide looking through technique for Greedy to make it appropriate on a huge chart. As of late a diagram dedensification approach has been suggested that adds Compressor Nodes in a chart for dedensification. The issue includes, given some question hubs and a huge chart, discover a thickly associated local area of hubs, including the inquiry hubs. The thickness standards depends on notable least degree measure, where point is to augment the base level of partaking hubs. Ravenous continues by eliminating each hub from the diagram, whose degree is least among rest of the hubs. In [8] the cycle is proceeded with when level of any of the question hub is lesser than a hub to be taken out in current emphasis or the subgraph gets detached . In [8] essayists found a beneficial outcome of problem areas sparsification for speeding up of local area search issue .

The resultant outline diagram is minimized, yet the arrangement has quadratic time unpredictability because of thorough pairwise looking. In this way, a set-based outline approach that totals sets of hubs was presented [9]. The methodology likewise connects with any organization proportions of neighborhood thickness which guarantee to be helpful both in portraying network structures and in looking at networks [10]. To accelerate the persuasive network search calculation, devise a straight space record structure which underpins effective hunt of the top-r k-powerful networks in ideal time [11]. An insatiable calculation is presented that over and over eliminate a vertex with the base weighted degree in leftover graph [12]. The structure of a diagram is construed through assessment of non-zero halfway sanctioned connection between's nodes [13]. In social ties between components can be relied upon to be founded on similitude across numerous ascribes, by far most of work to date on affiliation networks includes attaches characterized as for just a solitary attribute [14]. Analysts have focused especially on a couple of properties that appear to be normal to a huge number the little world property, power-law degree appropriations, and organization transitivity [15]. Existing diagram synopsis strategies are generally statistical.

Chapter 3

Problem Definition

3.1 Problem Definition:

We have considered an undirected, weighted graph $G = (V, E, W)$ with number of total vertices, $n = |V|$ vertices, number of total edges, $m = |E|$ edges and the set of weight of each corresponding weight is W .

Definition 1(Weight of a community): Given an undirected weighted graph $G=(V,E,W)$ and an induced subgraph $H= (V_H, E_H, W_H)$ 4 we define the weight of H as:

$$W(H) = \sum_{e \in E_H} \{w(e)\} \quad (3.1)$$

By Definition 1 the weight of a subgraph $H=0,1,2,6,7$ is the total weight of all of

its edges, $W(H)=1+2+3+2+3=11$.

Definition 2(Edge Density): We define edge density as the ratio of the weight of a community and the maximal number of edges in that community.

$$\rho = W(H)/l_e \quad (3.2)$$

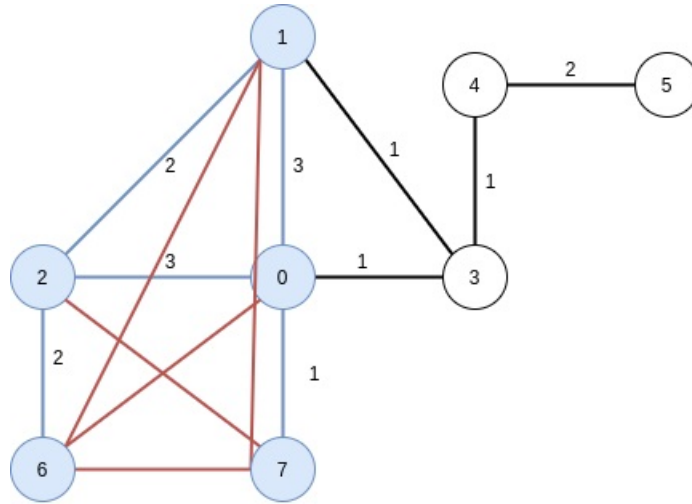


Figure 3.1: Maximal edges

In figure 3.1, the subgraph $H=0,1,2,6,7$ has a weight $W(H)=11$ and maximal edge number of 10. Thus the edge density of H is $11/10=1.1$

In this paper we use edge density as the goodness measure of our graph. Our goal is to increase the edge density of our resultant community compare to our input graph.

Definition 3(Support): The support of an edge $e(u,v)$ is defined by the edges that connect the common neighbors of each of the vertices of the edge.

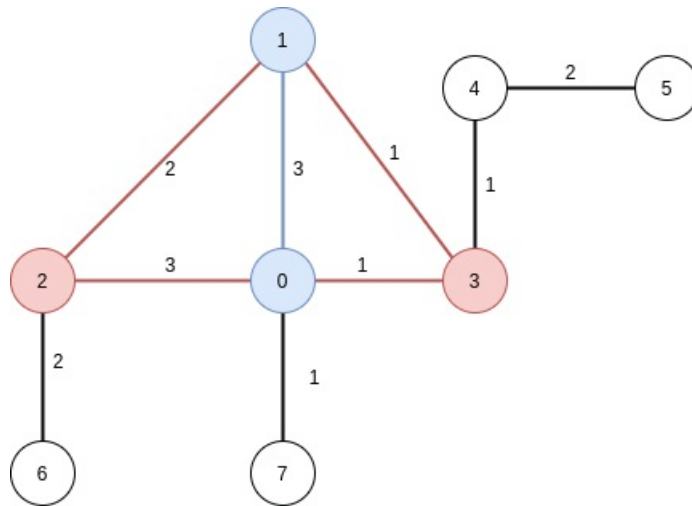


Figure 3.2: Support of an edge

In figure 3.2, the edge $(0,1)$ would be, $\text{sup}(0,1)=(0,2),(1,2),(0,3),(1,3)$

Definition 4(Decision Matrix Analysis):

Decision Matrix Analysis is a method of solving problems involving selection from among a finite number of alternatives. It involves analysis of all probabilities based

on specific criteria and scoring each probability based on those criteria. The next step is to make a decision based on the result of that analysis.

To find out a connected and cohesive community, we tried to improve the searching better and developed a novel algorithm to search for communities for specific query nodes in a large multi-attributed social graph based on certain goodness measures and such that our algorithm maintains efficiency based on time complexity.

3.2 Algorithms

Multi Attributed Decision Making

Multi Attributed Decision Making(MADM) is a method of solving problems involving selection from among a finite number of alternatives based on each alternative having multiple attributes. In the case of a multi-attributed graph managing the multiple weights of each edge is one such problem. Each attribute of the edge should be put through a custom weight adjustment algorithm that is fit for the graph and the circumstances. The summation of the adjusted weight values could be considered the new weight of the edge.

$$W_e = \sum(f_1(w_1), f_2(w_2), \dots, f_n(w_n)) \quad (3.3)$$

An override is also added at the top summation function to tweak the returned adjusted weights without having to directly modify the function if necessary. Thus the final adjusted weight can be found using the following algorithm. [16]

$$W_e = \sum(a_1f_1(w_1), a_2f_2(w_2), \dots, a_nf_n(w_n)) \quad (3.4)$$

Let e be an Edge of the graph $G = (V, E)$ such that $e \in E(G)$ with three attributes.

The weights of e are w_1, w_2, w_3 be weights associated with those attributes respectively. The adjusted weight values of these edges can be found using the functions f_1, f_2, f_3 respectively. After that weight overrides of a_1, a_2, a_3 can be added to and sum them to get the final weight w_e . This process is a globally accepted way to value each branch in a decision tree. If the incoming or outgoing (in a directed graph) or adjacent (in an undirected graph) edges from a vertex is considered as branches with different attributes with differing weights the use of the Decision Matrix Weight Generator can be seen in reducing the number of attributes in an edge while still not completely removing its value.

Scoring Algorithm

To make a decision on whether a neighbor of a query node can be considered, a scoring algorithm based on a selected goodness measure function must be implemented. Since edge density has been selected as our goodness measure our scoring algorithm will be as follows:

Algorithm 1

input : $G(V,E),e$

output: edgeDensityScore

begin

$s,t=e;$

 sourceNeighbors = $N(s)$

 targetNeighbors = $N(t)$

$H(V_H, E_H) = sourceNeighbors + targetNeighbors$

 edgeDensityScore = $W(H)/l_e$

 return edgeDensityScore

end

A graph $G(V,E)$ is taken and an edge e as input and the edge is divided into its two nodes s and t . A subgraph H is constructed from the neighbors of s , $N(S)$ and t , $N(t)$. The edge density of H , using Equation 3.2 is found and returned.

Although this algorithm does return the edge density of a subgraph of all the neighboring nodes of the two vertices of an edge it does not take into consideration the goodness of connectivity in the resulting subgraph. To solve that problem the concept of support is introduced into our algorithm. The optimized algorithm is as follows:

Algorithm 2

input : $G(V,E),e$

output: edgeDensityScore

begin

$H(V_H, E_H) = sup(e)$

 edgeDensityScore = $W(H)/l_e$

 return edgeDensityScore

end

Rather than creating a subgraph from all the neighboring nodes of an edge only nodes that make sure that all the edges in the new graph at least have two support-

ing edges are chosen. It is done so by only selecting the nodes that are common between the neighbors of the two vertices. This is an accepted way to ensure connectivity.

Filtering Algorithm

A recursive algorithm that takes a Graph $G(V, E)$ and a query node q and finds a community that node belongs to by analyzing and filtering the node's neighbors based on Algorithm 2.

Algorithm 3

input : $G(V, E), v_q$

output: subgraph H containing v_q

```

begin
  winner.push( $v_q$ )
  for  $v_n \in N(v_q)$  do
     $e = v_q, v_n$ ;
    scores.push( $edge = e, score = Score(G, e)$ );
  end
  medianScore=median(scores);
  for  $s \in scores.score$  do
    if  $s < medianScore$  then
      G.remove(scores.edge);
    end
  end
  for  $e \in scores.edge$  do
     $v_q, v_n = e$ ;
    if  $v_n$  not in winner then
      goto Step 1 with  $v_n$ ;
    end
  end
  H= winner;
  return H
end

```

All the vertices required for the final subgraph are added to a list, winner. Since the query node is going to be part of the subgraph it is added to the list. Then the edges connected to v_q are iterated through and find their scores using Algorithm 5. The edge and its score is pushed to a list for future processing. After all of the query vertex's edges are iterated through the median of the scores are found and remove the destination nodes of edges lower than the median from the graph. Median is chosen over average to account for any skewing of data where one edge might have a significantly higher weight compared to other edges. Generally, if the distribution of data is skewed to the left, the mean is less than the median, which is often less than the mode. If the distribution of data is skewed to the right, the mode is often

less than the median, which is less than the mean[17]. The next node is selected to be put through the same process. To do so any node that has already been selected as a winner is ignored. The process runs until all nodes have either been selected to be winners or disconnected. A subgraph with all the winner nodes is formed and returns it as the resultant community.

Although this algorithm serves our purpose it has however an issue since it does not keep track of the edges it visits. If a neighbor of a query node is input into the algorithm it will consider the shared edge of the nodes twice and analyze them both times. This will significantly increase the algorithms run time as it will iterate over the same edges multiple times. To solve this problem the algorithm is modified in the following manner:

Algorithm 4

```

input :  $G(V,E),v_q$ 
output: subgraph H containing  $v_q$ 
begin
  winner.push( $v_q$ )
  for  $v_n \in N(v_q)$  do
     $e = v_q, v_n$ ;
    if  $e$  not in visited then
      scores.push(edge= $e$ ,score=Score( $G,e$ ));
      visited.add( $e$ );
    end
  end
  medianScore=median(scores);
  for  $s \in scores.score$  do
    if  $s < medianScore$  then
       $G.remove(scores.edge)$ ;
    end
  end
  for  $e \in scores.edge$  do
     $v_q, v_n = e$ ;
    if  $v_n$  not in winner then
      goto Step 1 with  $v_n$ ;
    end
  end
  H= winner;
  return H
end

```

In steps 4 and 6 we introduce a check to stop the analysis of duplicate edges.

Filtering Algorithm with Distance Constraint

Although Algorithm 4 filters out many edges and vertices based on the scoring mechanism we still end up with strongly connected graphs where certain nodes can be too far away from the query node to have any meaningful impact in the real world. That is why we introduce the concept of a distance constraint into our algorithm. This not only makes the resultant community more impactful but also cuts down execution time by acting as another filter layer. The modified algorithm is as follows:

Algorithm 5

input : Graph $G(V,E)$, query node v_q , *distanceconstraint* dc

output: subgraph H containing v_q

```

begin
  originalQueryNode  $v_q$ 
  winner.push( $v_q$ )
  for  $v_n \in N(v_q)$  do
     $e = v_q, v_n$ ;
    if  $e$  not in visited then
      scores.push(edge= $e$ , score=Score( $G, e$ ));
      visited.add( $e$ );
    end
  end
  medianScore = median(scores);
  for  $s \in$  scores.score do
    if  $s <$  medianScore or distance( $v_q, scores.edge[v_n]$ )  $>$   $dc$  then
       $G.remove(scores.edge[v_n])$ ;
    end
  end
  for  $e \in$  scores.edge do
     $v_q, v_n = e$ ;
    if  $v_n$  not in winner then
      goto Step 2 with  $v_n$ ;
    end
  end
   $H =$  winner;
  return  $H$ 
end

```

In Step 10 of Algorithm 8 whether the distance between the query node and the destination of the node that has been analyzed is greater than the input distance constraint is checked. If the check passes it is removed from the graph.

Chapter 4

Experiment

In this section our proposed algorithm is tested with different distance constraints on two datasets on Google Colaboratory with an Intel(R) Xeon(R) CPU @ 2.20GHz processor with 13GB memory. The edge density of the returned community was recorded at different distance constraints and analyzed. So was the community size and execution time of the algorithm at different distance constraints.

Dataset

For our work, We need A dataset from where we can gather data which can be measured, recorded and visualized using graphs after processing all the data is needed. Firstly, data has been collected using a social site and that is "YouTube". Now our proposed algorithm will be applied on the data collected from "YouTube". In our research, an efficient and iterative algorithm has been presented . As "YouTube" is a social network site wherein a specific network lots of people can connect through their videos with each other, they can share their ideas and thinking through videos. We can considered this graph as a multi-attributed graph. In dataset of "YouTube" have been found some data in a specific group there like source, target, shares, likes which can be addressed as edges and attributes. Edges usually are in dense concentration among the neighboring members of the community in high tech real-life networks. These nodes represent the whole dataset of the site and it represents how many sources and targets there are as well as the number of likes and shares it got. Additionally, the edge is the connection between some specific nodes of a graph. As a multi-attributed undirected graph is being used so all the edges here are bidirectional. Also, as the graph is a weighted graph, so each of the edges of our graph has some fixed value like connected numerical value, and those values are called edge weight. Moreover, Edge weight is referred to as the "cost" of a particular edge. These weights are sometimes measured by distance or length. Additionally, From the research, it can be seen that there is no exceptional case in the YouTube dataset. The CSV file contains all the data and values of all nodes.

Additionally, another one we used is the NSA. One of the ways of NSA's collecting data is phone, email, meeting. They calculate how many emails and meetings were conducted within 1 year.

Our collected dataset is split up into edges and attributes. Here the connections between nodes are considered as Edges and Phone Calls, Emails and Meetings as attributes.

Table 4.3: Dataset list collected from NSA

Edges		Attributes		
Source	Target	Phone calls	Emails	Meetings(in hours)
690506	395507	992	1239	2712.8
302593	439921	427	13	2460.60
439921	395507	52	664	395.10
582820	911456	5	20	514.2
911456	302593	18	30	4865.0
395507	176506	21	3711	2480.35
302593	176506	131	25	3920.14
176506	439921	10	18	2712.8
754336	819606	20	4210	587.4
582820	439921	199	31	938.61

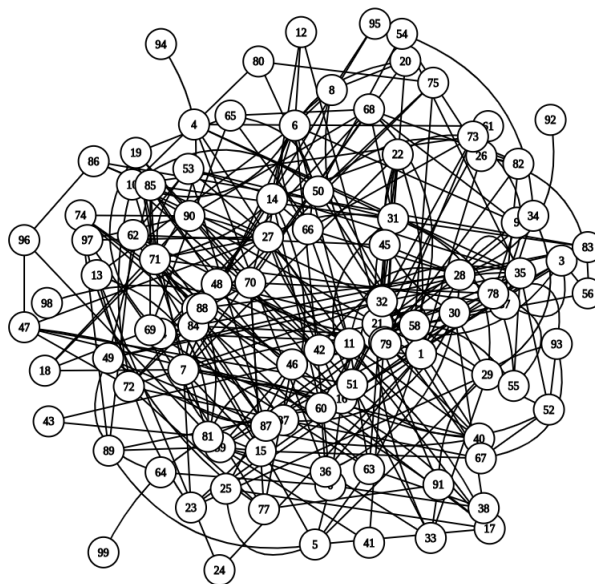


Figure 4.2: A sample graph from nsa

Result Analysis

Goodness Function Performance

The algorithm was run on two datasets with varying distance constraints and no constraint to analyze the constraint's effects on the edge density of the resultant community.

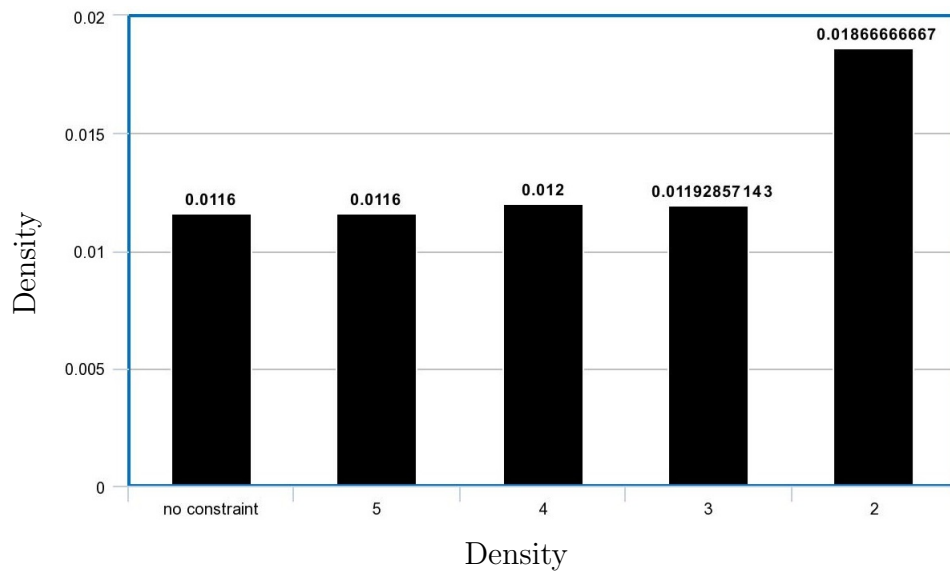


Figure 4.3: Youtube density for distance constraint

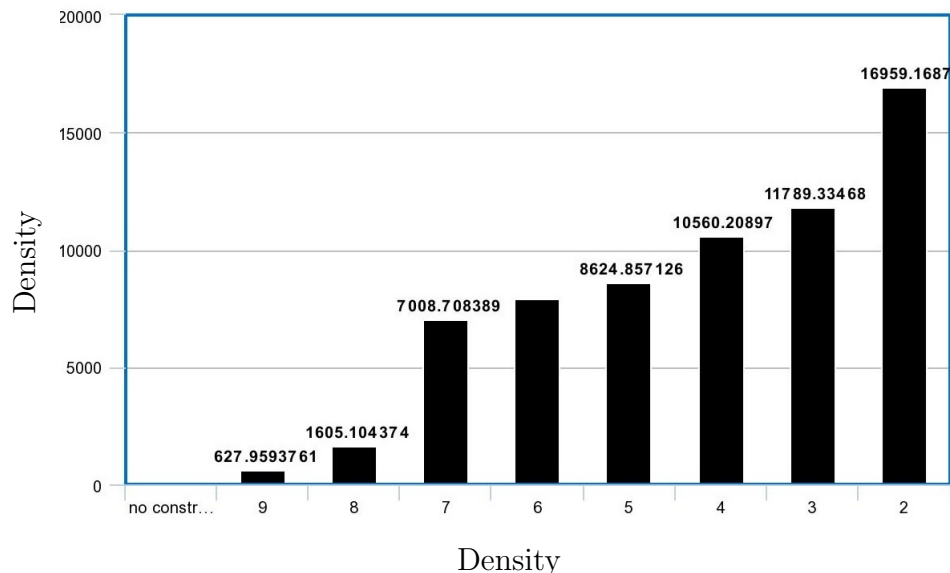


Figure 4.4: NSA density for distance constraint

In Figure 4.3 and Figure 4.4 it can be seen that increasing edge density as distance constraint decreases. This means that distance constraint has a positive effect on our goodness measure. This mainly happens due to a decrease in maximal edges as

the community size decreases as is evident in Figure 4.5 and Figure 4.6.

As the youtube dataset is considerably smaller the farthest node from the query node is only 5 connections away. Thus a relatively smaller increase in density is seen as distance constraint is reduced.

But the NSA dataset is comparatively larger and the farthest node without constraint is approximately 200 connections away. This means the link between the furthest node and query node is weaker. With a much smaller distance constraint imposed on the algorithm lessens the size of community considerably. This means the difference between the actual edges and maximal edges is much smaller compared to without constraint. Thus the edge density increases as the farthest node comes nearer.

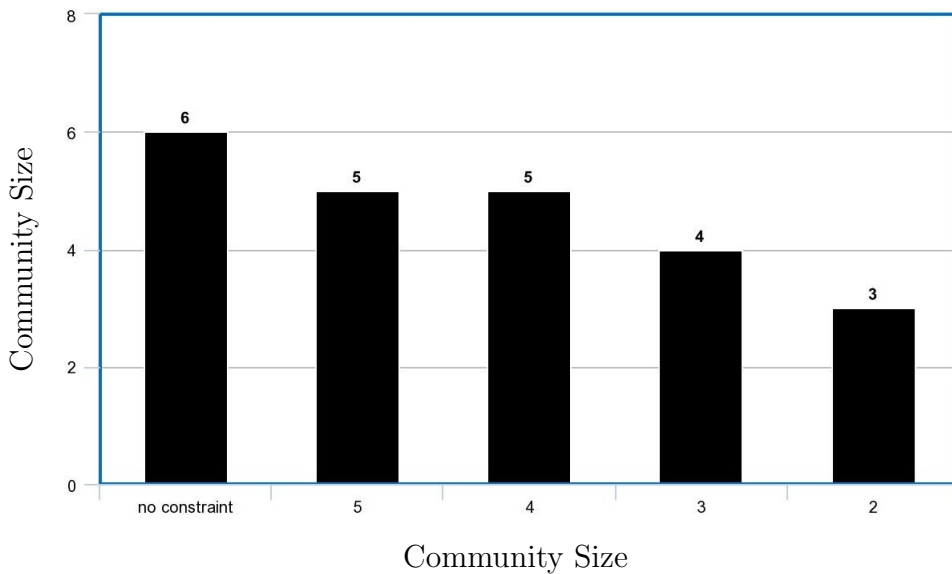


Figure 4.5: Youtube community size for distance constraint

As the distance constraint is lower the community is bounded by only the closest of neighbors which results in a smaller and denser community.

Iteration Reduction

Implementing distance constraint on Algorithm 8 also reduces the number of recursive iterations it has to perform. It is evident in Figure 4.8 and Figure 4.9 that the number of nodes visited reduces steadily.

Modern programming languages have e.g. python has a limitation. Python imposes a (system-dependent) limit on the size of the call stack [17]. This is done to ensure that the memory stack is not overloaded and to reduce stress on the compiler. To solve this problem it is essential to reduce the number of nodes visited as each node would need to be fed through the algorithm in a recursive manner. Imposing distance constraint serves that purpose. From Figure 4.7 it can be seen that the reduction in visited nodes is not that high. This will happen when the graph size is

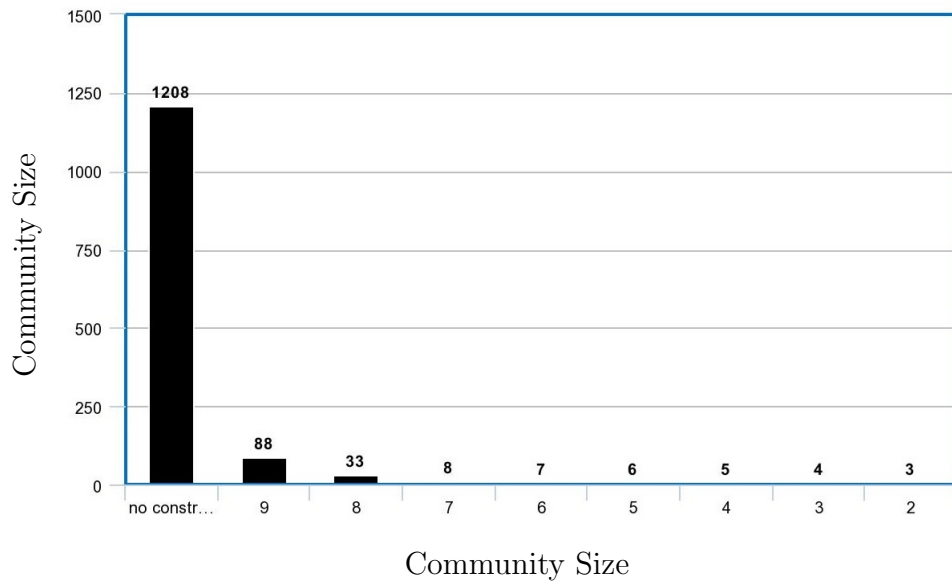


Figure 4.6: NSA community size for distance constraint

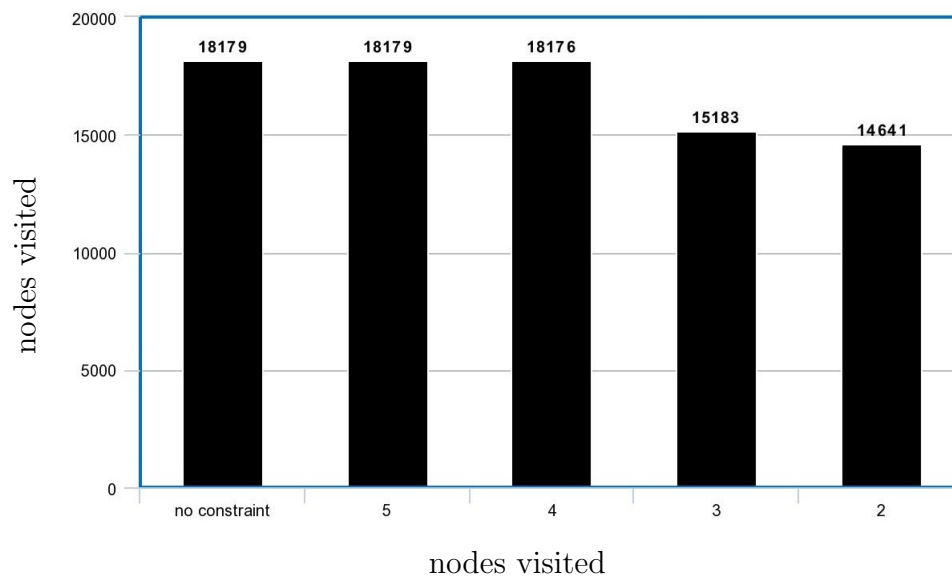


Figure 4.7: Youtube nodes visited for distance constraint

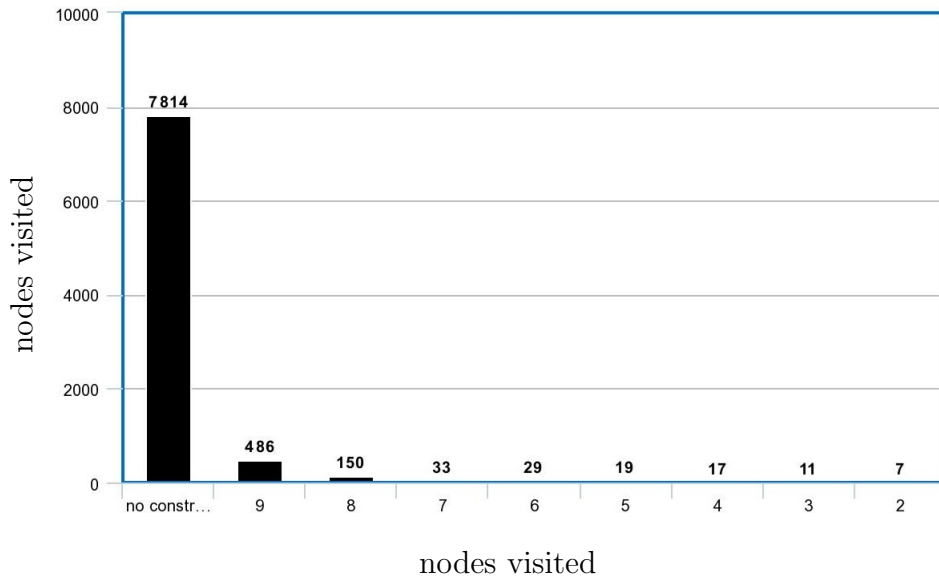


Figure 4.8: NSA nodes visited for distance constraint

small and is well connected. Distance constraint is unable to reject nodes as they are already very close to the query node. But in a loosely connected graph like the NSA dataset it can be seen from Figure 4.8 that there is a drastic reduction in iterations going from an infinite constraint to a distance constraint of nine. From 9 to 8 a good reduction was seen. But as the constraints got lower the reduction also got lower. This means that the algorithm reached the closet of nodes and was unable to filter out nodes from further iterations.

Time Performance

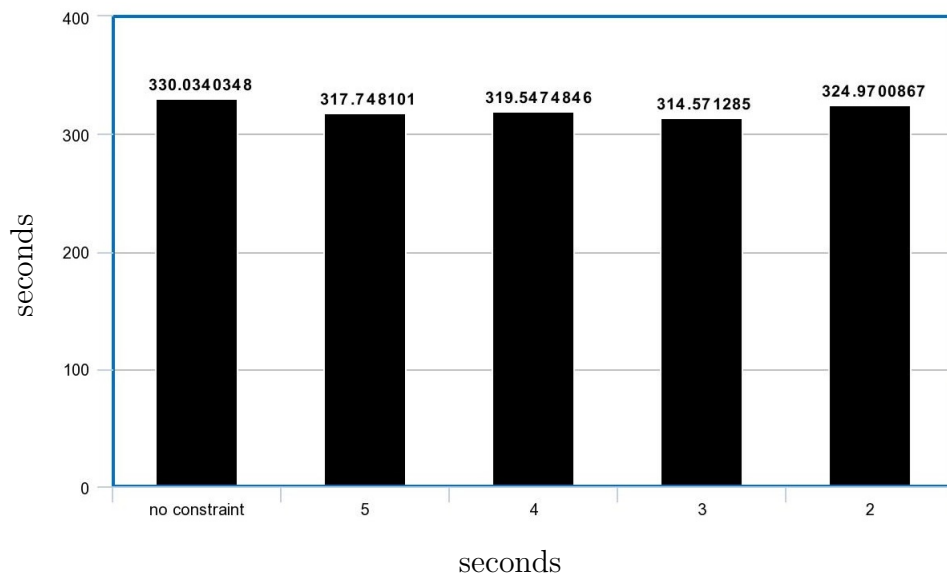


Figure 4.9: Youtube distance constraint executing time

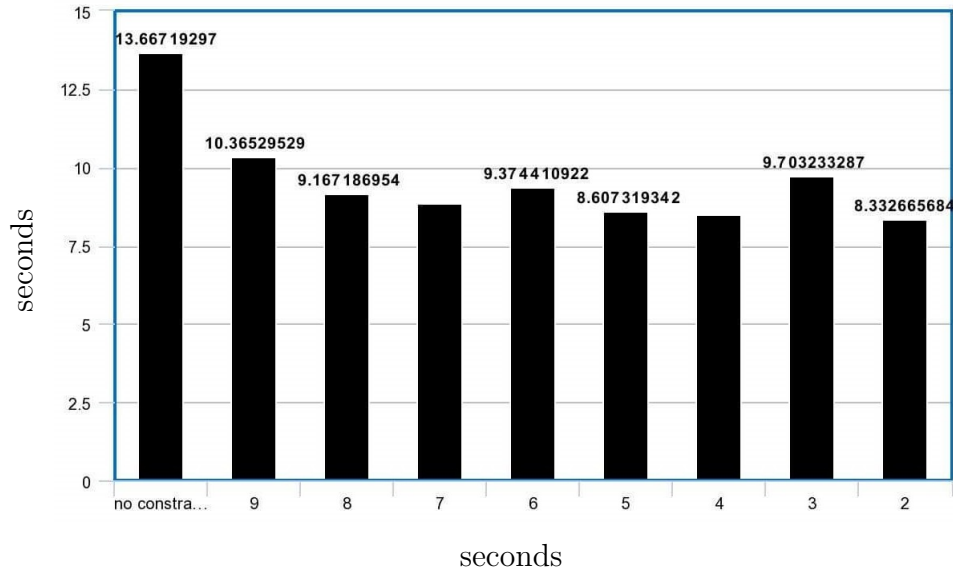


Figure 4.10: NSA distance constraint executing time

The results indicate that although distance constraint does have some impact in reducing execution time it might not be that significant. This coupled with the fact that distance constraint greatly reduces nodes visited it spends more time in analyzing nodes and making decisions than processing them. From Figure 4.9 and Figure 4.7 it can be seen that although a distance constraint of 2 reduces the number of nodes visited by approximately 4000 it still increases the execution time by 7 seconds.

In Figure 4.10 a 23% reduction in execution time is observed in the NSA dataset as it goes from infinite constraint to a distance constraint of 9. But the execution time becomes irregular after that. This requires further study into what is consuming the most time in a single iteration. It is possible that if that part is optimized in a later study execution can also be reduced in the same pattern as iterations were reduced.

Chapter 5

Application

As an application example, NSA is the one with whom our research can be related. NSA a national-level intelligence agency whose responsibilities are global monitoring, collecting, and processing information for both foreign and domestic intelligence and safety purposes. Monitoring is done by NSA through financial records, e-mails, and also through some social networks like Facebook. Moreover, sometimes extensive surveillance can be performed by NSA on phone call, texts also. In our research the sample data have been used are the volume of phone calls, emails, and time spent in meetings are some examples of data NSA would collect from the population. Each node is considered a person and the edges are the communications taking place between two people. Our proposed algorithm can be used to filter out what community the target belongs to. This will be helpful in sifting out connections e.g terrorist links and identify previously unknown suspicious individuals. The call detail records program has been using to preserve the technique of contact-chaining which means the checking of patterns between whom the contacts of a surveillance target call and receives calls from. This means with a call detail records order, the NSA acquires all metadata about the target's incoming and outgoing calls (the "first hop"), and all phone records of the people who contacted the target (the "second hop"). Thus, under a single order, the NSA collects all the phone records of target and "first hop" individuals, and some of the phone records of "second hop" individuals.[18]

Each call detail record contains, the time and date of the call as well as the duration of the call. Further, the records show what cell towers were used by a phone during the phone call.[17]

Ever since Edward Snowden leaked the secrets of the NSA, we've learned that the covert organization has tabs on a lot more civilian activity than we originally suspected including copies of our emails. The NSA has been tracking civilian emails since the Bush administration. Even so, it's safe to assume that the average business-or person-doesn't have any emails dating back as far as 2002.[19]

Defense Secretary Bob Gates signed a document that OK'd the collection and mining of "the information appearing on the 'to,' 'from' or 'bcc' lines of a standard email or other electronic communication" from, well, you and your friends and maybe some terrorists.[20]

In the sample data used in our work the volume of phone calls, emails, and time spent in meetings are some examples of data NSA would collect from the population. Each node is considered a person and the edges are the communications taking place between two people. Our proposed algorithm can be used to filter out what community the target belongs to. This will be helpful in sifting out connections e.g terrorist links and identify previously unknown suspicious individuals.

Additionally, as another application example related with our research, a large number of data from YouTube have been used which is currently the largest host to streaming video content on the Internet. To collect popular YouTube users, at first the list of YouTube user list need to be downloaded from any of the popular websites like socialblade.com, vidstatsx.com etc. which provide the list of the most subscribed users.

Chapter 6

Future Works and Conclusion

While good progress has been made, research on community search is still in its infancy, and there are many opportunities for further research. We have spotted some of the encouraging directions.

In this recent time, we all know a global pandemic happening all around the world that is Corona Virus. Many humans are continuously being attacked by this virus and dying. So it is needed to find out in which areas of the world are affected mostly by this virus. By using the community search algorithm in a largely attributed graph we can find out in which area this virus is affecting mostly so that we can prevent that.

Also, it is needed to give the vaccine of COVID 19 to the people of the most affected area. By using the community search algorithm we can calculate and find out in which area vaccines are most needed and how many vaccines we need in a particular area.

In the future, we aim to team up with some specialist teams and built this project so that it can be added to the existing search methodology.

To conclude, Community detection is a fundamental problem of network science that refers to the finding of communities in any graph data. It is not dependent on a query node. Community search, on the other hand, is a narrower field of community detection that depends on a query node and tries to find the communities that the query node belongs to. So our purpose was to find a better search than the other existing search algorithm. So, we have worked for a search method which is a novel algorithm of online search by using a technique of keyword search over the attributed graph. As we know that the node or social networks are strongly connected. So by using this it is possible to get a better result as the keyword search technology makes it possible to search the community of nodes in an online and computationally efficient manner.

Bibliography

- [1] P. Attachment, “Chaos, complexity, and inference (36-462),” 2009.
- [2] W. Cui, Y. Xiao, H. Wang, and W. Wang, “Local search of communities in large graphs,” in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014, pp. 991–1002.
- [3] M. Sozio and A. Gionis, “The community-search problem and how to plan a successful cocktail party,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 939–948.
- [4] Z. Zheng, F. Ye, R.-H. Li, G. Ling, and T. Jin, “Finding weighted k-truss communities in large networks,” *Information Sciences*, vol. 417, pp. 344–360, 2017.
- [5] X. Huang, H. Cheng, L. Qin, W. Tian, and J. X. Yu, “Querying k-truss community in large and dynamic graphs,” in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014, pp. 1311–1322.
- [6] K. U. Khan, T. N. Anh, M. R. Akhond, W. Nawaz, and Y.-K. Lee, “Accelerating community-search problem through faster graph dedensification,” in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, 2017, pp. 340–347.
- [7] M. Abulaish, “A novel weighted distance measure for multi-attributed graph,” in *Proceedings of the 10th Annual ACM India Compute Conference*, 2017, pp. 39–47.
- [8] R. V. Rao, *Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods*. Springer Science & Business Media, 2007.
- [9] K. U. Khan, W. Nawaz, and Y.-K. Lee, “Set-based approximate approach for lossless graph summarization,” *Computing*, vol. 97, no. 12, pp. 1185–1207, 2015.
- [10] S. B. Seidman, “Network structure and minimum degree,” *Social networks*, vol. 5, no. 3, pp. 269–287, 1983.
- [11] R.-H. Li, L. Qin, J. X. Yu, and R. Mao, “Influential community search in large networks,” *Proceedings of the VLDB Endowment*, vol. 8, no. 5, pp. 509–520, 2015.
- [12] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, “Greedy finding a dense subgraph,” *Journal of Algorithms*, vol. 34, no. 2, pp. 203–221, 2000.

- [13] M. Kolar, H. Liu, and E. P. Xing, “Graph estimation from multi-attribute data,” *Journal of Machine Learning Research*, 2014.
- [14] N. Katenka, E. D. Kolaczyk, *et al.*, “Inference and characterization of multi-attribute networks with application to computational biology,” *The Annals of Applied Statistics*, vol. 6, no. 3, pp. 1068–1094, 2012.
- [15] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proc. Natl. Acad. Sci. USA*, vol. 99, no. cond-mat/0112110, pp. 8271–8276, 2001.
- [16] R. K. Ronhovde and Z. Nussinov, “An edge density definition of overlapping and weighted graph communities,” *arXiv preprint arXiv:1301.3120*, 2013.
- [17] *Levitan, ben*. “*use of meta-data by the nsa.*” *nsa meta-data information by ben levitan*, 2007, <https://www.benlevitan.com/tracklocation.html>, (Accessed on 01/09/2021).
- [18] *Laperruque, jake*. “*the history and future of mass metadata surveillance.*” <https://www.pogo.org>, 2019, www.pogo.org/analysis/2019/06/the-history-and-future-of-mass-metadata-surveillance, (Accessed on 01/09/2021).
- [19] *Qureshi, azam*. “*email archiving: Nsa has copies of all your emails, why don’t you?*” *intradyn*, 15 may 2020, <https://www.intradyn.com/email-archiving-nsa-has-copies-of-all-your-emails-why-dont-you>, (Accessed on 01/09/2021).
- [20] R. H. Sloan and R. Warner, “When is an algorithm fair? errors, proxies, and predictions in algorithmic decision making,” *Errors, Proxies, and Predictions in Algorithmic Decision Making (April 16, 2018)*, 2018.