Fast unfolding of communities in large networks

Social Network Analysis for Computer Scientists — Course paper

Parthipan Ramakrishnan
s3447014@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Piyush Dash
s3671097@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

In various domains, such as social networks or bio-informatics, detecting communities and be able to detect the patterns efficiently and effectively is of immense importance. The problem of extracting better communities in a network is achieved by using Modularity Optimization in a heuristic approach. Our method results in efficient detection of communities in Large networks and results in better computation time, and at the same time, not trading off with the quality of communities . It results in better inferences/analysis from implementation of an efficient algorithms than classical methods such as Girvan Newman algorithm. In order to show the efficiency of our method, the algorithm will be used in large networks with inferences from our selected datasets .

KEYWORDS

Modularity, Modularity Optimization, large Networks, Community Detection, social network analysis

ACM Reference Format:
Parthipan Ramakrishnan and Piyush Dash. 2022. Fast unfolding of communities in large networks: Social Network Analysis for Computer Scientists — Course paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2022 (SNACS '22).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

In the context of networks, community structure refers to the occurrence of groups of nodes in a network that are more densely connected internally than with the rest of the network. Community structure, or clustering, is one of the most important characteristics of networks that describe actual systems. It refers to how vertices are organized into clusters, with numerous connections connecting vertices in the same cluster and relatively few edges connecting vertices in other clusters.[8] In order to detect communities in the large networks, we need to detect groups/clusters of nodes that are densely connected, and posses sparse connection between the nodes from other group. In real life, similar behaviour is exhibited by people, who tend to associate more with those who posses similar interests as themselves and sparsely with people from other groups. Finding clusters of people with similarity in preferences can be useful for link prediction in market segmentation and recommendation systems . In order to achieve this, decomposing networks into communities, which are collections of nodes with strong connections, is a viable strategy[9].Identification of these communities is essential because it may enable the discovery of functional modules that were previously unknown, such as cyber-communities in social networks or subjects in information networks. Additionally, the resultant meta-network, whose communities serve as its nodes, may be utilized to view the original network topology[2].
The purpose of this paper is to compare and analyze popular existing methods for detecting communities in various data sets. Analyzing the performances of several existing algorithms and doing a comparative study with the method proposed yields interesting observations especially when applied to data where the definition of community is unconventional. For e.g. analyzing purchasing patterns of consumers who buy products from an e-commerce website. The ecosystem of similar products being purchased by a section of consumers here can be described as a community.

2 PROBLEM STATEMENT

In order to solve the challenge of community identification, a network must be divided into communities of tightly linked nodes with only sparse connections between the nodes in other community[9]. In context of large networks with several thousands of nodes, it becomes increasingly important to discover communities fast and efficiently. In order to find communities, one of the prevalent methods is to use optimization methods which are based on the maximization of an objective function[15]. In order to assess the quality of partition produced by such approaches so-called 'modularity' is frequently used. It measures the density of links inside communities as compared to links between communities[14] .

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{1}$$

Here, A_{ij} represents the weight of the edge between i and j, $k_i = \sum_j A_{ij}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, the δ-function $\delta(c_i, c_j)$ is 1 if $c_i = c_j$ or 0 otherwise and m = $\frac{1}{2} \sum_{ij} A_{ij}$ It's used also as an objective function to optimize, however, this optimization problem's precise formulations are known to be computationally challenging, as discussed later[15]. In order to identify decent partitions in a reasonable amount of time, several techniques have been developed. Due to the growing accessibility of big network data sets and the influence of networks on day-to-day life, there is a necessity for algorithms which can generate decent results in shorter period of time.
Another challenge in community detection is that there is no precise definition of community[10].As a result , it is not easy to judge

SNACS '22, Master CS, Fall 2022, Leiden, the Netherlands

the quality of the partitions. In order to solve this, this study proposes a mechanism to balance the size of the community prior to merging, enhancing the algorithm's performance and decreasing calculation time simultaneously.

2.1 Modularity Optimization

Finding graph partition with highest value of modularity is NP-hard problem[15]. Modularity optimization on real life large networks is computationally challenging[9]. The greedy approach repeatedly combines the communities that optimize modularity; as a result, it could provide values for modularity that are lower than those obtained by other techniques. Additionally, this technique can create super-communities that contain the bulk of the network's nodes, which can slow it down and render it useless on big networks[4].

2.2 High Computation Time

There are several community detection algorithms discussed later, among which Modularity based approaches are well known and commonly used. When dealing with networks with millions of nodes and links, we want to be able to comprehend the community structure fast. As remarked earlier, finding the best partition of the graph in a relatively short time is a difficult task. The fastest approximation algorithm for optimizing modularity on large networks was proposed by Clauset et al[4], however it has a tendency to generate values of modularity that are significantly low. Hence, we require a heuristic method which uses approximation to achieve the task of optimizing modularity in a shorter run time.

3 RELATED WORK

3.1 Girvan-Newman

One of the most widely used algorithm to evaluate the quality of a partition of a network is the Girvan-Newman algorithm. It uses edge betweenness (equation. 2) to find and remove central edges that connect communities within a larger graph[11]. After removing an edge, the Girvan-Newman algorithm calculates the modularity (Q) of the graph, which is a value in the range [-1/2,1].

$$B(e) = \sum_{u,v \in V(G)} \frac{\sigma_{u,v}(e)}{\sigma_{u,v}} \tag{2}$$

Where, $\sigma_{u,v}$ is the number of shortest paths between two distinct vertices and $\sigma_{u,v}(e)$ is the corresponding number of shortest paths containing a particular edge. A higher modularity value suggests more significant community structure. Therefore, we can identify communities by maximizing modularity. This process of removing an edge and calculating the modularity is iterated repeatedly. The algorithm will stop when there cannot be any further improvement in the modularity.
It is difficult to find smaller communities using this algorithm, due to the modularity optimization, the algorithm fails to detect "modules smaller than a scale" which depends on the total size of the network. This often tends to be large in real life networks[1].
As a result, while using this algorithm, one should not take the result at face value and detect larger community structures and then examine the detected communities for sub communities. Despite Girvan-Newman's popularity and quality of community detection, it has a high time complexity, increasing up to $O(m^2n)$ on a sparse graph having m edges and n nodes. Hence, Girvan-Newman is generally used on networks with node count in few thousands or lesser[1].

3.2 Clauset, Newman and Moore

Because of the caveats and limitations of Girvan-Newman, there exist greedy algorithms for detecting communities ,in order to reduce the run-time. CNM algorithm will recurrently merge communities that optimize the modularity but at the same time compromising the accuracy of the results. Moreover, this greedy approach can produce modularity values less than actual values, produce super-communities with large fraction of nodes[4]. This slows down the algorithm making it inapplicable for large networks as the time complexity of this algorithm is considerably higher.

3.3 Wakita and Tsurumi

Wakita and Tsurumi introduced 3 heuristics in order to balance the size of communities being merged, thereby speeding up the run time and making it possible to run on large networks[18].
1st Heuristics - Measures the community size in terms of its degree
2nd heuristics - Choice of the pair with largest change in modularity is two staged
3rd heuristics - Measures the size of community in terms of the number of its members
Though the addition of the 3 heuristics made it possible to run on large networks, the algorithm comes with the caveat that may sometimes result in bad modularity value, which could be the result of the heuristic which creates balanced communities.

4 APPROACH

In most real life large networks such as Mobile Network service providers or social Media Networking sites like Twitter with millions of users, there are several natural organization levels communities divide themselves into sub-communities and it is thus desirable to obtain community detection methods that reveal this hierarchical structure. To unfold communities faster in a network, we introduce an algorithm, which consists of two phases.
The phases are :

- **Phase 1 :- Modularity Optimization :** In the first phase, we begin with an N-node weighted network. Initially each node is assigned to a unique community i.e.number of nodes is the same as the number of communities. Now, we begin the optimization of modularity of node i by taking a look around its neighbors j . We assess if there can be a gain in modularity value by transferring i from its community and into the community of j. If there is a tie, the breaking rule is used, and if the gain is positive, the node i is then assigned to the neighbor community j. If no gain is attainable, i remains in its native community. Modularity gain ΔQ obtained by moving an isolated node i into a community C can be computed by :

$$\Delta Q = \left[\frac{\sum_{in}+2k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{3}$$

where $\sum_{in}$ is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and m is the sum of the weights of all the links in the network. The first phase of our algorithm is said to be finished when no further improvement can be achieved, after repetition and sequential application to all nodes. This first phase stops when a local maxima of the modularity is attained, i.e., when no individual move can improve the modularity.

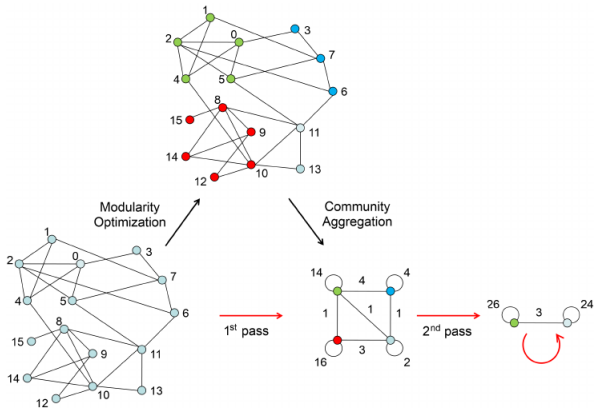

Figure 1: 2 Phase method

- **Phase 2 :- Community Aggregation :**The algorithm's second phase entails creating a new network with the communities discovered in the first phase as its nodes. To do this, the weights of the links between the new nodes are determined by adding the weights of the links between nodes in the relevant two communities. Links between nodes belonging to the same community in the new network cause self-loops for this community. It is then conceivable to repeat the algorithm's first phase on the weighted network produced after this second phase is finished.

Let us use the word "pass" to describe the union of these two stages. By design, there are fewer meta-communities in each run, thus the first pass consumes the majority of the computation time. The passes are repeated until the maximum level of modularity is reached and there are no longer any modifications. As communities of communities are formed during the process, the algorithm is suggestive of the self-similar structure of large networks and naturally integrates a sense of hierarchy. The number of passes determines the hierarchy's height, which is typically only a small number(less than 8 in most cases).This straightforward approach comes with some benefits. Firstly, the process is easy to comprehend and put into practice, and the results are unattended. The technique is also significantly faster, and computer simulations on big ad-hoc modular networks indicate that its complexity is linear on typical and sparse data. This is because our approach method makes it simple to calculate any changes in modularity and since there are fewer communities after just a few rounds, taking the majority of the run-time. The initial iterations receive most of the attention as the part of detection of communities by initializing nodes into its neighbor's community is done here . The inherent multi-level character the approach also appears to avoid the so-called resolution limit problem which arises when using modularity. The time complexity of this algorithm is $O(Nlog(N))$

5 DATA

For the purpose of achieving better inferences from the application of our method, we intend to use it on real world datasets which have large number of nodes and edges. The datasets we choose are all of various sizes to show the scalability of the fast unfolding algorithm.

- **Zachary's karate club :** is a social network of a university karate club [20]. It is a well known dataset for community structure. It has 34 nodes and 78 edges.
- **US. Rap Album :** a network of famous Rap albums and artist from US over the years. It has 815 nodes and 831 edges.We got the dataset from Kaggle[5].
  Data Processing:
  - We have removed some unwanted data points like url, and other artist attributes.
  - We found the cosine similarity between 2 albums and created a link between the 2 albums.
- **Bike Travel :** a network of bike travel between places in UK over the course of 2021 - 2022*. It represent a direction connection of a bike travel between two places in UK. We got the dataset from Transport for London [6]. It has the bike id, start and end destination. The dataset has 800 nodes and 277,852 edges.
  Data Processing:
  - We have removed some unwanted data points like rental Id from the dataset.
  - We used the duration of the travel as weights for our edges. Duration was in seconds we converted that to hours.
- **Protein-Protein Interaction :** PPI is a network of different physical interactions between proteins. We got the dataset from SNAP [13]. It has 8,152 nodes and 156,642 edges.
- **Board Directors :** It is a Bipartite graph of directors and companies beyond the Fortune 500 companies. We got the dataset from networks.skewed.de[7] After processing we got 259,977 nodes and 266,098 edges. The nodes and Edges are similar because the dataset has both Company as well as person records. A link is between a company and a person (board member).
  Data Processing:
  - We have removed the non-relevant details like, number of employees, and Null values from the dataset.
  - We used the company and the board member as the link.
  - Each edge will have a company, name , person(target)

- **Amazon Meta :** it contains product metadata and review about 548,552 different products (Books, music CDs, DVDs and VHS video tapes) on Amazon website. We got the data from SNAP[13]. After processing the data, it has 479,749 nodes and 881,736 edges.
  Data Processing:
  - We have parse the meta data to get nodes and edges from the dataset, we removed the non relevant data points like ID, reviews and kept the ASIN (product id) as ID.
  - We used the product and the similar product as the link.
  - Each edge will have a ASIN(source), Title , group, and similar product(target)
- **Reddit User Post :** It is a network of reddit user's post on different topic over a years. It has 15,122 nodes and 4,535,176 edges. The link is between the user and the topic on which he posts. We got the dataset from SNAP[13].
- **Google WebGraph :** It is a network of web pages and hyperlinks between them. It has 875,713 nodes and 5,105,039 edges.We got the dataset from SNAP[13]. A link is between a web page(source) and hyperlink(target) in the web pages.

Dataset	Nodes	Edges	Avg. Clustering Coefficient	Density	Type
Karate Club	34	78	0.57064	0.13903	Undirected
US. Rap Album	815	831	0.04936	0.00125	Undirected
Bike Travel	800	277852	0.65061	0.44128	Directed
Protein-Protein	8,152	156,642	0.13060	0.00471	Directed
Board Directors	259,977	266,098	0.10300	2.9634e-06	Directed
Amazon Meta	560,728	1,763,471	1.3698e-06	3.8309e-06	Directed
Reddit User Post	15,122	4,535,176	0.40980	0.01983	Directed
WebGraph-Google	875,713	5,105,039	0.36983	6.6569e-06	Directed

Table 1: This table shows the different datasets used and the characteristics like nodes, edges, average clustering coefficients, Density and type of datasets.

6 EXPERIMENTS

For the comparative performance analysis of fast unfolding algorithm over the other algorithms, we created a performance testing model to calculated the number of communities found, modularity of the partition, and computation time taken by the algorithm. We ran the performance testing model against the real world datasets of various sizes as mentioned in the data section to analysis the performance of the fast unfolding algorithm with other community detection algorithms like Greedy Modularity (CNM algorithm), Walktrap, Surprise, Girvan-Newman and Naive Greedy Algorithm explained below.

- **Walktrap** - It is hierarchical clustering algorithm proposed by Pons & Latapy[16]. It is based on the idea that short distance random walk have the tendency to form clusters. The distance between the adjacent nodes are calculated from a totally non-clustered partition. We then merge the two adjacent clusters and update the distance between the two. This step is repeated for (N - 1) times. So, the time complexity of Walktrap is $O(MN^2)$.
- **Naive Greedy Modularity** - The naive greedy modularity algorithm is a heuristic method for identifying communities or clusters within a network. It is a simple and efficient approach that can be used to identify highly modular structures within a network quickly. The algorithm works by starting with each node in the network as its own community, and then iteratively merging communities based on the modularity score. The modularity score is a measure of the density of connections within a community compared to the density of connections between communities. The time complexity of this algorithm is $O(N^2)$
- **Greedy modularity** - The greedy modularity algorithm[4] is a heuristic method for identifying communities or clusters within a network. It is a more advanced version of the naive greedy modularity algorithm, which uses a different heuristic for identifying communities within the network. The algorithm works by starting with each node in the network as its own community, and then iteratively merging communities based on the modularity score. The modularity score is a measure of the density of connections within a community compared to the density of connections between communities. The time complexity of this algorithm is $O(Mlog(N))$.
- **Girvan-Newman** - The Girvan-Newman algorithm[12] is a heuristic method for identifying communities or clusters within a network. It is a divisive hierarchical clustering algorithm, which means that it starts with the entire network as a single community and then iteratively splits it into smaller communities. The algorithm works by repeatedly removing the edge with the highest betweenness centrality from the network. Betweenness centrality is a measure of how many shortest paths pass through a given edge. Removing an edge with high betweenness centrality is likely to split the network into two or more smaller communities. The time complexity of this algorithm is $O(N^3)$
- **Surprise** - Surprise algorithm was proposed by V. A. Traag, R. Aldecoa, and J.-C. Delvenne[17]. Surprise community detection algorithms work by analyzing the connections between individuals in a network and identifying groups of individuals who are more strongly connected to one another than they are to the rest of the population. This can help identify communities within a larger population and can be useful for a variety of applications, such as identifying subgroups within a social network or understanding the structure of a population.
- **Fast unfolding** - The fast unfolding algorithm proposed by Vincent et. al [2]. The two phase method with modularity optimization and community aggregation we discussed. The time complexity of this algorithm is $O(Nlog(N))$.

We used NetworkX and Community packages of python for the performance testing model. For the visualization we used Gephi, Matplotlib, Seaborn and NetworkX. Hardware used for the experimentation is Intel i5-12500H (16 CPUs) with 16Gb ram.

From the image 2 we can see the communities found using various algorithms on the Karate club network, our algorithm found communities faster and the quality measure i.e., modularity is higher than all the other algorithms used for comparison refer table 2.

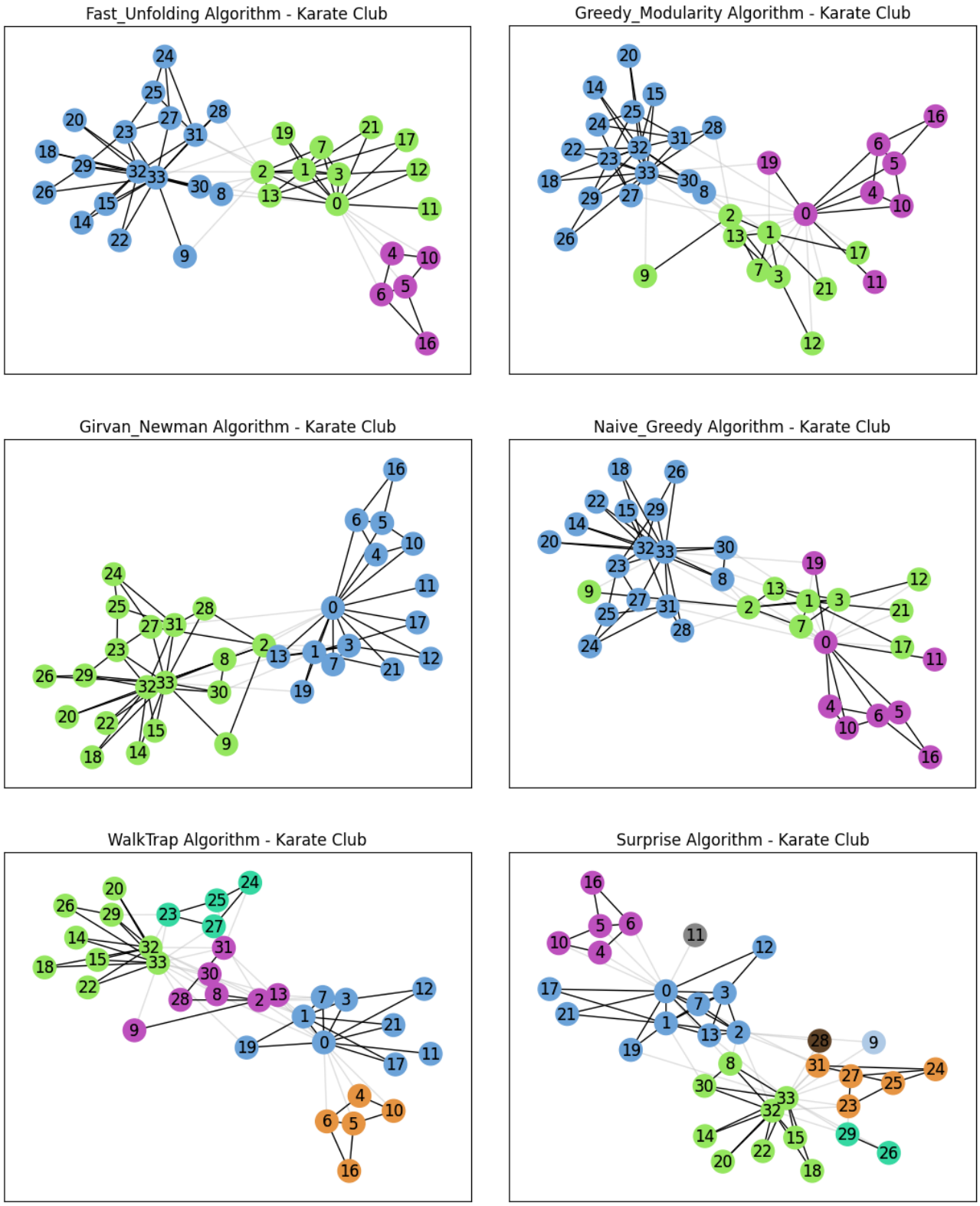

Figure 2: Community detection on Karate Club network by fast unfolding, greedy modularity, Girven-Newman, Naive Greedy, Walktrap and Surprise algorithms

Table 2: Summary of experimentation. This table gives the performances of the algorithm of Fast Unfolding, Greedy Modularity, Naive Greedy, Walktrap, Girven-Newman and Surprise algorithm for community detection in networks of various sizes. For each method/network, the table displays the communities found, modularity that is achieved and the computation time. Empty cells correspond to a computation time over 24 h. Fast unfolding method clearly performs better in terms of computer time and modularity then the other algorithms.

	Karate club	US. Rap Album	Bike Travel	Protein-Protein	Board Directors	Amazon Meta	Reddit User	Google WebGraph
Node/Edges	34/78	815/831	800/277K	8k/156K	356K/376K	560K/1.76M	15K/4.5M	875K/5.1M
Fast Unfolding	3 / **0.444**/ 0.0s	91 / 0.925 / 0.04s	3 / 0.207 / 2.38s	20 / **0.291** / 5.14s	9606 / **0.965** / 104.20s	5152 / **0.886** / 600.84s	1512 / **0.900** / 104.47s	3149 / **0.97946** / 594.83s
Greedy Modularity	3 / 0.411 / 0.0s	88 / **0.926** / 0.04s	3 / **0.208** / 6.64s	45 / 0.243 / 290.82s	10059 / 0.965 / 195.93s	–/–/–	–/–/–	–/–/–
Girvan Newman	2 / 0.348 / 0.0s	76 / 0.720 / 1.55s	3 / 0.000 / 51.75s	12 / 0.001 / 89632.30s	–/–/–	–/–/–	–/–/–	–/–/–
Naive Greedy	3 / 0.411 / 1.87s	86 / 0.828 / 540s	–/–/–	–/–/–	–/–/–	–/–/–	–/–/–	–/–/–
Walktrap	5 / 0.323 / 0.0s	98 / 0.690 / 1.03s	5 / 0.170 / 1.53s	620 / 0.182 / 11.89s	–/–/–	–/–/–	–/–/–	–/–/–
Surprise	8 / 0.418 / 0.0s	151 / 0.860 / 0.01s	793 / 0.002 / 2.36s	1031 / 0.157 / 0.91s	–/–/–	–/–/–	–/–/–	–/–/–

To verify the speed and quality of the fast unfolding algorithm, we applied it to the datasets mentioned in the data section and we have compared it with 5 other community detection algorithms. The table 2 shows the number of communities found, modularity achieved and the computation time taken for each dataset by all the mentioned community detection algorithm. Even for the largest dataset i.e., Google Webgraph it has 875K nodes and 5.1M edges, it took less than 10 minutes foe the fast unfolding algorithm to detect the communities. From the table 2 we can see that the fast unfolding algorithm found better modularity then the other community detection algorithm and also the computation time is better then the other algorithms. We mentioned earlier that the modularity helps us in measuring the quality of partition, with the fast unfolding algorithm we got better modularity which means high quality communities that too faster compared to other algorithms, also the community detection algorithms like Greedy Modularity, Girven Newman, Naive Greedy, Walktrap and Surprise algorithms cannot perform well on large networks over Million edges, even for the protein-protein interaction network with 156K links Girvan-Newman took around 90,000 seconds which is more than 24hrs, whereas the highest time the fast unfolding took was 600 seconds that too for a large network with 5M links. Overall, the fast unfolding algorithm found better quality communities in a small time then the other community detection algorithms we compared.

7 APPLICATION TO LARGE NETWORK :

To validate the fast unfolding algorithm, we applied it to Amazon Metadata dataset. The dataset is based on co-purchasing network i.e., Customers Who Bought This Item Also Bought. The network is composed of 764K nodes and 1.76M links. A link is basically between two co-purchased products. The dataset consists of ASIN - which is the product ID, Title of the product, Type of the product, Salesrank and Co-purchased product. This dataset is exceptional because we can infer details like how people are purchasing similar products, connection between 2 or more products and also the purchasing behaviour of people. Also we can use this to recommend products to the customers as well.

We applied the fast unfolding algorithm to the Amazon Metadata dataset, the algorithm found 5152 communities with a modularity of 0.886 within 10 minutes. The algorithm took more time because we used text data instead of IDs, to get more info about the data. Out of the 5152 communities, 31 communities has a size of more than 5,000 products, 11 communities has a size of more than 10,000 products and 2 communities has a size of more than 65,000 products. These communities comprises of more than 70% of the entire products in the dataset.

7.1 Observations :

We have found some interesting observations from the detected communities,

1) The table 3 shows one of the communities the fast unfolding algorithm found, looking into the data we can see that all the nodes in the community are of product type 'book' ,particularly religious books referring to Christianity.

A Community of Amazon Metadata		
ASIN	**Title**	**Type**
0842310444	The Handbook of Bible Application	Book
0842328130	Ephesians	Book
0842328327	Life Application Bible Commentary: 1 and 2	Book
084232853X	1 & 2 Corinthians	Book
0842328572	1, 2, & 3 John	Book
0842328610	Acts	Book
0842328629	1 & 2 Thessalonians	Book
0842328742	Life Application Bible Commentary: Revelation	Book
0842328904	Romans	Book
0842330313	1 Peter 2 Peter Jude	Book

Table 3: All the products in the community are books and related to the Christianity.

2) The table 4 shows another community our algorithm found, all the products in the community are of type 'DVD' and all belongs to same category of product in this case Marvel Avengers series.

A Community of Amazon Metadata		
ASIN	**Title**	**Type**
0767015533	Avengers '67 - Set 2, Vols. 3 & 4	DVD
0767018664	Avengers '66 - Set 1, Vols. 1 & 2	DVD
0767018699	Avengers '66 - Set 2, Vols. 3 & 4	DVD
6305299951	Avengers '67 - Set 3, Vols. 5 & 6	DVD
B00000IC8Z	Avengers '67 - Set 1, Vols. 1 & 2	DVD
B00000JMQJ	Avengers '65 - Set 1, Vols. 1 & 2	DVD
B00000JMQR	Avengers '65 - Set 2, Vols. 3 & 4	DVD

Table 4: All the products in the community are DVDs of Marvel Comics, particularly the Avengers series.

3) The another community (table 5) the algorithm found is an interesting one because this community has mixture of many product types like "Book", "DVD" and "Music", even-though the products are of different types they are refer to a particular music band called **ABBA** [19], which is a Swedish supergroup formed in Stockholm in 1972 by Agnetha Fältskog, Björn Ulvaeus, Benny Andersson, and Anni-Frid Lyngstad. The group name ABBA is the acronym of their first letters. This community shows a mix of all the type of products and shows how good the communities found by fast unfolding algorithm.

A Community of Amazon Metadata		
ASIN	**Title**	**Type**
0711983895	Bright Lights, Dark Shadows : Story Of ABBA	Book
B000001E6T	Thank You for the Music	Music
B00006AUG0	Abba - The Definitive Collection	DVD
B0002NIB6U	Abba - The Last Video	DVD
B00065TZE6	Abba - Super Troupers	DVD

Table 5: Products in this community are of different types like DVDs, Books, and Music, but focused on a Swedish musical supergroup called ABBA

These are the interesting communities we found in the Amazon metadata set, we visualized the Amazon Metadata datasest in Gephi with fast folding algorithm to detect communities in the dataset, the image 3 shows the visualization of communities found in the Amazon metadata using the fast unfolding algorithm.

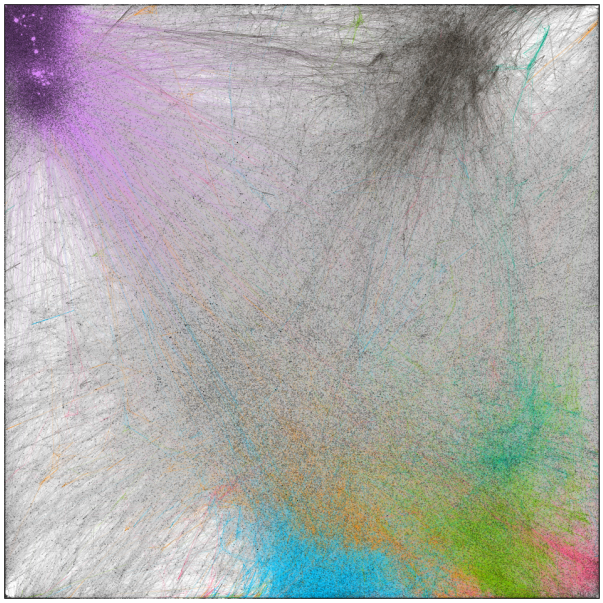

Figure 3: Visualization of communities found in the Amazon Metadata dataset using the fast unfolding algorithm

In the the visualization of communities in Amazon Metadata dataset image 3, the purple coloured region in the top left corner are of communities with product type Music, the black one on the top right are DVDs, the blue, greenish yellow and Pink coloured regions on the bottom right are books community focused on science, religions and literature respectively.

8 CONCLUSION

Finding communities in a network has a immense importance in the modern world that runs on data. This can be useful for a variety of purposes, in a social network understanding the structure of social networks or identifying potential collaboration opportunities within a group of individuals. And can be useful in many fields like bio-informatics, marketing, computer science etc., Overall, the use of community detection can help to better understand the structure and relationships within a network, and can provide valuable insights that can inform a wide range of decision-making and analysis tasks. In order to find communities within a network, we need community detection algorithms to run on huge amount of data, in the modern era data is growing exponentially[3] this means we need algorithm that runs faster and finds better quality communities on large networks. From the performance experimentation on the large networks, we can clearly see that the fast unfolding algorithm ran faster and found better communities then the other algorithms. The fast unfolding algorithm addresses the modularity optimization problem by introducing an optimization formula thus solves the hard computation time problem. The modulairty optimization technique, the quality of detected communities and the run time to find them makes the fast unfolding algorithm a better algorithm to run on large networks with more than Millions of nodes then the other community detection algorithms.

ACKNOWLEDGMENTS

REFERENCES

[1] [n. d.]. ([n. d.]). https://medium.com/smucs/girvan-newman-and-louvain-algorithms-for-community-detection-f3feb7c31908

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (oct 2008), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

[3] Tawfik Borgi, Nesrine Zoghlami, Mourad Abed, and Naceur Mohamed Saber. 2017. Big Data for Operational Efficiency of Transport and Logistics: A Review. 113–120. https://doi.org/10.1109/ICAdLT.2017.8547029

[4] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70 (Dec 2004), 066111. Issue 6. https://doi.org/10.1103/PhysRevE.70.066111

[5] Nolan Conaway. 2016. *18,393 Pitchfork Reviews*. Pitchfork reviews from Jan 5, 1999 to Jan 8, 2017.

[6] cycling.data.tfl.gov.uk. 2021. *Cycle Counters*.

[7] Anna Evtushenko and Michael T. Gastner. 2019. *Data set discussed in "Beyond Fortune 500: Women in a Global Network of Directors"*. https://doi.org/10.5281/zenodo.3553442 A. Evtushenko and M. T. Gastner, Beyond Fortune 500: Women in a global network of directors. In H. Cherifi et al. (Eds.), Complex Networks and Their Applications VIII, Proc. 8th Int. Conf. Complex Networks and Their Applications, Volume 1, pp. 586-598 (Springer, Cham, 2020), DOI: 10.1007/978-3-030-36683-4_47. M. T. G. was supported by the Singapore Ministry of Education and a Yale-NUS College start-up grant (R-607-263-043-121)..

[8] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3-5 (feb 2010), 75–174. https://doi.org/10.1016/j.physrep.2009.11.002

[9] Santo Fortunato and Marc Barthé lemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104, 1 (jan 2007), 36–41. https://doi.org/10.1073/pnas.0605965104

[10] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *Physics Reports* 659 (nov 2016), 1–44. https://doi.org/10.1016/j.physrep.2016.09.002

[11] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (jun 2002), 7821–7826. https://doi.org/10.1073/pnas.122653799

[12] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826. https://doi.org/10.1073/pnas.122653799 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.122653799

[13] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[14] M. E. J. Newman. 2004. Analysis of weighted networks. *Physical Review E* 70, 5 (nov 2004). https://doi.org/10.1103/physreve.70.056131

[15] M. E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* 74, 3 (sep 2006). https://doi.org/10.1103/physreve.74.036104

[16] Pascal Pons and Matthieu Latapy. 2005. Computing Communities in Large Networks Using Random Walks. In *Computer and Information Sciences - ISCIS 2005*, pInar Yolum, Tunga Güngör, Fikret Gürgen, and Can Özturan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 284–293.

[17] V. A. Traag, R. Aldecoa, and J.-C. Delvenne. 2015. Detecting communities using asymptotical surprise. *Phys. Rev. E* 92 (Aug 2015), 022816. Issue 2. https://doi.org/10.1103/PhysRevE.92.022816

[18] Ken Wakita and Toshiyuki Tsurumi. 2007. Finding Community Structure in Mega-scale Social Networks. https://doi.org/10.48550/ARXIV.CS/0702048

[19] Wikipedia. 2022. ABBA — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=ABBA&oldid=1124302785. [Online; accessed 10-December-2022].

[20] Wikipedia contributors. 2022. Zachary's karate club — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Zachary%27s_karate_club&oldid=1111053967.