Community Detection and their Analyses on the “Amazon Product Co-purchasing” Network

Jui Dakhave, Saurabh Kulkarni, Rajath Warrier
California State University, Chico
CSCI - 651 - Applied Graph Theory
Dr. Richard Carter Tillquist
11th Dec. 2021

INDEX

ABSTRACT

Graph theory is used to study many complex topics related to computer science, biology, politics, physics, and neurology. One of the most important features of networks is community structure. To explore the properties and structure of complex networks, many community detection algorithms have been developed. But it often gets difficult to understand which algorithms are good at accurately detecting the communities. We examine real-time community detection - Greedy Modularity, Louvain Method, Girvan-Newman Algorithm and multiple sampling techniques - Random walk, Snowball on a network with 925,872 edges and 334,863 nodes. We also point out strengths and weaknesses of popular methods, and give directions to their use. To compare community detection methods, we used the LFR benchmark that generates heterogeneous community mixing fractions.

INTRODUCTION

Our dataset is an Undirected Amazon product co-purchasing network with 925,872 edges and 334,863 nodes. Network was collected by crawling the Amazon website. It is based on Customers Who Bought This Item Also Bought feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. Each product category provided by Amazon defines each ground-truth community. Each connected component in a product category has a separate ground-truth community. The minimum community size in our network is Four nodes. Given a huge real graph, There are many known algorithms to compute interesting measures (shortest paths, centrality, betweenness, etc.), but several of them become impractical for large graphs. Thus graph sampling is essential. We consider several sampling methods, random walk and snowball sampling. These sampling strategies do not perform well; We developed our sampling technique which returns a subgraph that preserves our community structure and then using that subgraph, we detected communities using multiple algorithms such as greedy , Louvain and Girvan Newman algorithms. Later for accuracy, we calculated the Normalized Mutual Information by comparing ground truth communities and the communities we found using our algorithms. Evaluating community detection methods is a challenging task. The lack of reliable ground-truth gold-standard communities has made community detection very challenging. We used the LFR benchmark to evaluate community detection algorithms. We first show that the variation in community mixing fractions has different impacts on the performances of different community detection methods that could change the decision to select a particular detecting algorithm. We give a survey of related work in the next section and look further into the characteristics of the algorithm, discussing the applications of the algorithm on different types of networks. Next section gives detailed comparisons between the Louvain, Girvan Newman and fast modularity-optimization algorithms, and explains why we think Louvain is the best. We conclude the paper with future work in our research.

RELATED WORK

In their study, Zhao Yang, René Algesheimer & Claudio J. Tessone compared eight community detection methods using the Lancichinetti-Fortunato-Radicchi benchmark [2]. The eight community detection algorithms being compared are - Fastgreedy, Infomap, Leading eigenvector, Label propagation, Multilevel, Walktrap, Spinglass, Edge betweenness. This study uses the mixing parameter as an indicator of ranges of reliability.

Ian X.Y. Leung, Pan Hui, Pietro Lio', Jon Crowcroft, in their study of real-time community detection, used label propagation to detect the communities [3]. It shows potential implementations, improvements and applications of the algorithm on different types of networks. They also performed detailed comparisons between the label propagation algorithm (LPA) and fast modularity-optimization algorithms.

Jaewon Yang, Jure Leskovec studied a set of 230 large real-world social, collaboration and information networks where nodes state their group memberships [4]. They developed an evaluation methodology for comparing network community detection algorithms based on their accuracy on real data and compared different definitions of network communities and examined their robustness. The data we are using is sourced from this research paper.

METHODS

Sampling Technique

In order to detect communities efficiently, sampling the correct nodes corresponding to the community structure was important. We tried three sampling techniques, Random Walk sampling method, Snowball sampling method, and Community Structure method [2]. The procedure to determine the best technique to retain the community structure, we followed the following steps. First we sampled a fixed number of nodes, used a greedy modularity method [8] to detect the communities and compare them with ground truth communities corresponding to the sampled nodes. The x axis represents the number of communities. The y axis represents the communities sorted according to size of each community. To reduce the processing time, we chose a different dataset to run the sampling. The dataset chosen was generated using email data from a large European research institution. It contains a total of 1005 nodes with a total of 42 predefined ground truth communities.

Random Walk sampling

This sampling method takes a random-walk approach. Given a starting point, a neighbor for this starting point is chosen uniformly at random. Again, a neighbor for this node is chosen uniformly at random. This algorithm runs for a set number of choices. The sampled graph contains all the nodes and edges that are visited during the random-walk.

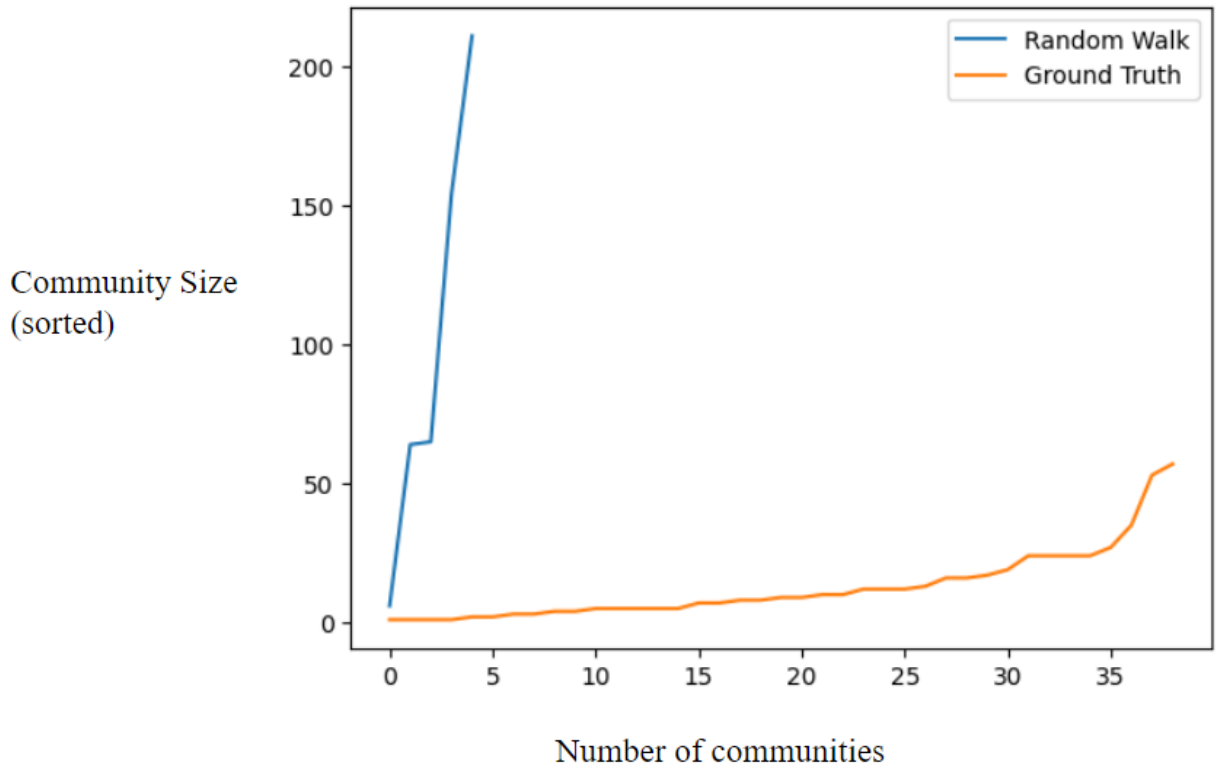


Fig 1. Number of communities obtained using Random walk sampling

After sampling through the random-walk method, the greedy modularity community detection algorithm was able to detect only 5 communities from the sampled set of nodes. The low number indicates that random walk sampling does not provide insights for community structure of the original network.

Snowball sampling

Snowball sampling uses breadth first search (BFS) from a given starting node to discover its neighbors and traverse through the network. At each step, exactly a given number of neighbors are chosen and this procedure continues until the desired number of nodes are sampled or BFS is unable to find a new neighbor. Unlike the random-walk sampling method, a neighbor is chosen only if has not been visited before. The sampled graph contains all the nodes visited during traversing through the network.

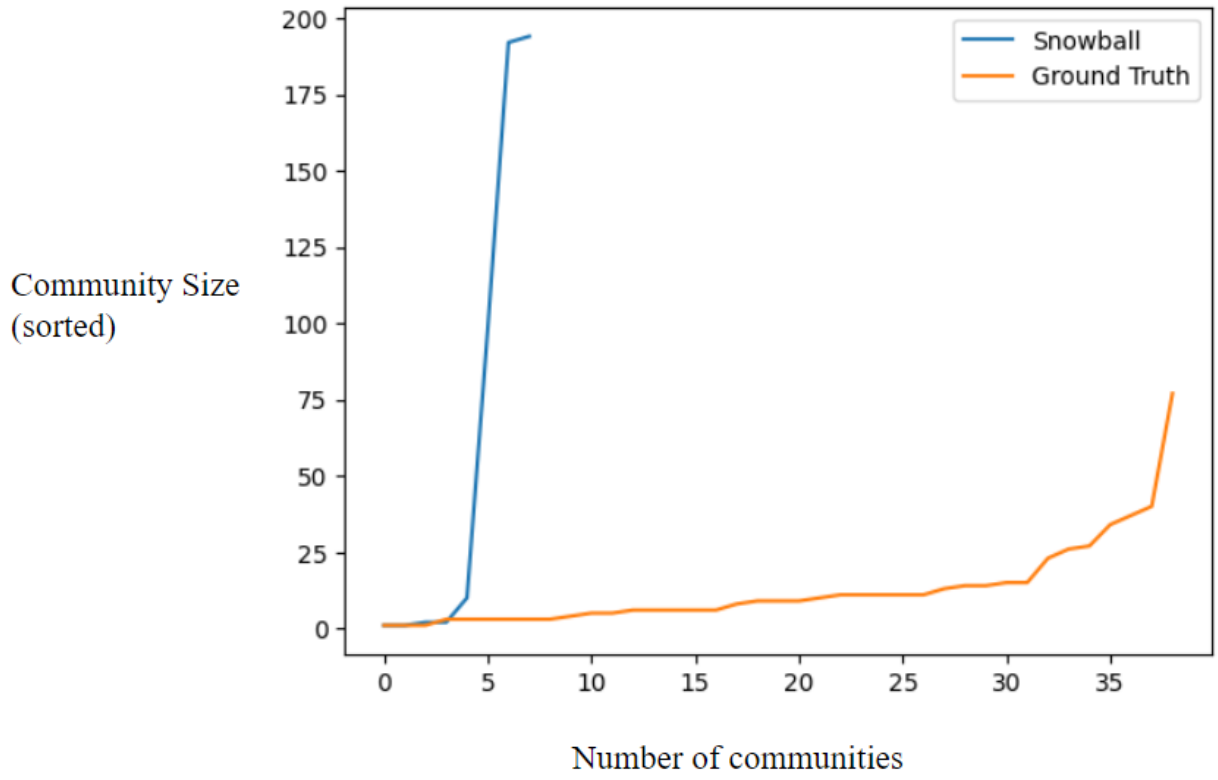


Fig 2. Number of communities obtained using Snowball sampling

Greedy modularity was able to detect 7 communities from the subgraph sampled using Snowball sampling. The low number as compared to ground truth communities indicates that Snowball sampling is not well suited for our requirements. This is likely because no factors other than the BFS are taken into consideration while sampling the network.

Expander sampling

This sampling method uses the properties of expander graphs. Expander graphs are highly connected yet sparse graphs. The method is based on the fact that samples with good expansion properties tend to be more representative of community structure in the original network than samples with worse expansion [1]. Starting from a given node, we find neighbors of neighbors and calculate the expansion factor for each of the sets. The nodes contributing to the maximum expansion factor are taken as a sample.

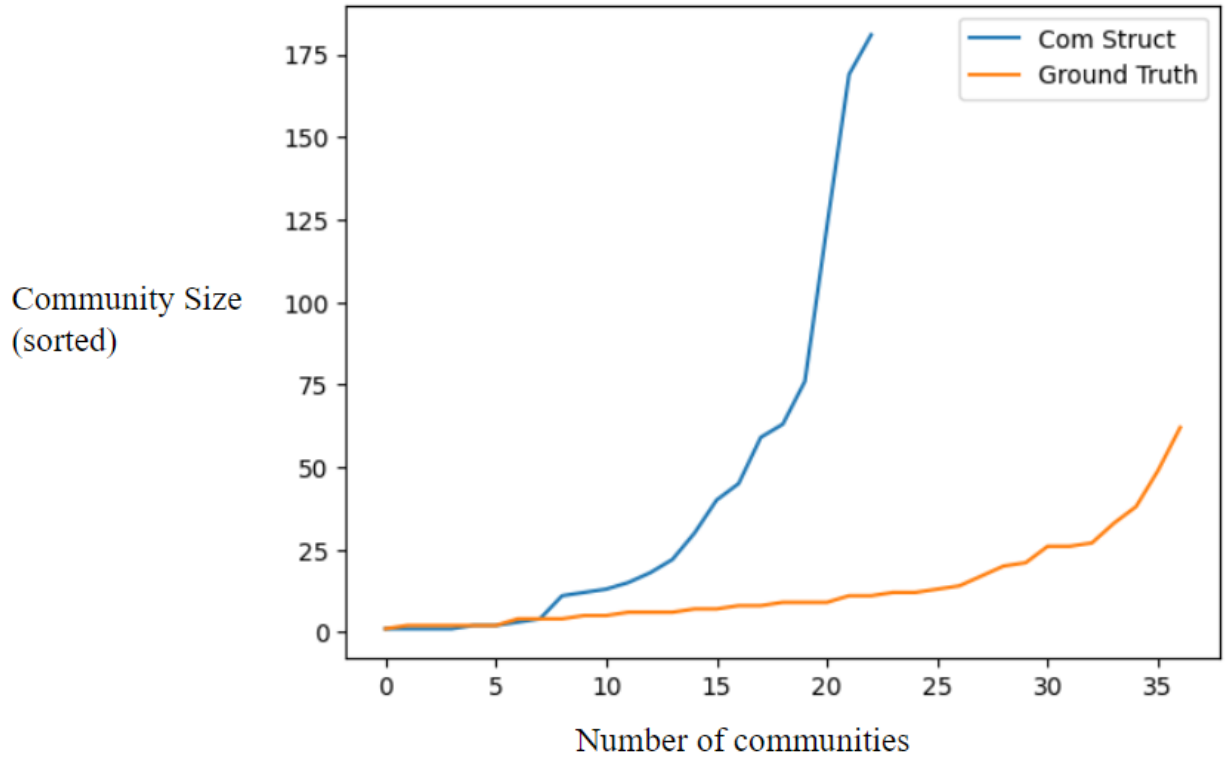


Fig 3. Number of communities obtained using Expander Sampling

The greedy modularity community detection algorithm was able to find a total of 22 communities. This shows that Expansion sampling is better at retaining the community structure from the original network. Hence we decided to use expansion sampling on our “Amazon Product Co-purchasing” network to reduce the number of nodes while making sure that community structure remains similar to that of the original network.

Community Detection

Networks with ground-truth communities: The network we consider is the Amazon product co-purchasing network. Here the product represents the node of the network and co-purchased products are linked using edges. Each product (i.e., node) belongs to one or more product categories and products from the same category define a group which we view as a ground-truth community. Note that here the definition of ground-truth is somewhat different. The lack of reliable ground-truth labels and presence of overlapping communities has made the accuracy calculation very challenging. Also overlapping communities doesn't work on some benchmark techniques. So to overcome this overlapping issue, for each node we assigned it to the largest community it is a part of. This may have impacted our overall analysis. Due to absence of reliable ground truth, we used normalized_mutual_info_score from sklearn. This function is useful to measure the agreement of two independent label assignment strategies on the same dataset when the real ground truth is not known.

Benchmarking

In order to understand which community detection algorithms perform better, we used LFR Benchmarking to observe the trends of the Normalized Mutual Information (NMI) with respect to the Mixing Parameter μ. NMI is a measure widely used in Information Theory. On a high-level, NMI captures the similarity between 2 partitions. One partition will come from the ground-truth communities that we have for the network and the other partition will be the communities generated from the LFR Benchmark structure. We used the LFR Benchmarking function available in NetworkX. The parameters it takes are number of nodes (n), power law exponent for the degree distribution of graph (τ_1), power law exponent for the community size distribution (τ_2), mixing parameter (μ), average degree, maximum degree of network, minimum size of communities, and maximum size of communities.

RESULTS AND DISCUSSION

Community Detection

Greedy Modularity approach

This approach uses greedy modularity maximization. Greedy modularity maximization begins with each node in its own community and joins the pair of communities that most increases modularity until no such pair exists. This approach does not work well if the average size of the community is very small. We used the optimized greedy modularity function available in NetworkX. Here we got a hairball like structure and the accuracy we got is 0.09 using normalized_mutual_info_score from sklearn. In Greedy, we can expect lower accuracy given the average size of the community for this data structure is very less. We only have 100 nodes per community. We can conclude that the Greedy Modularity approach does not work for such data structures.

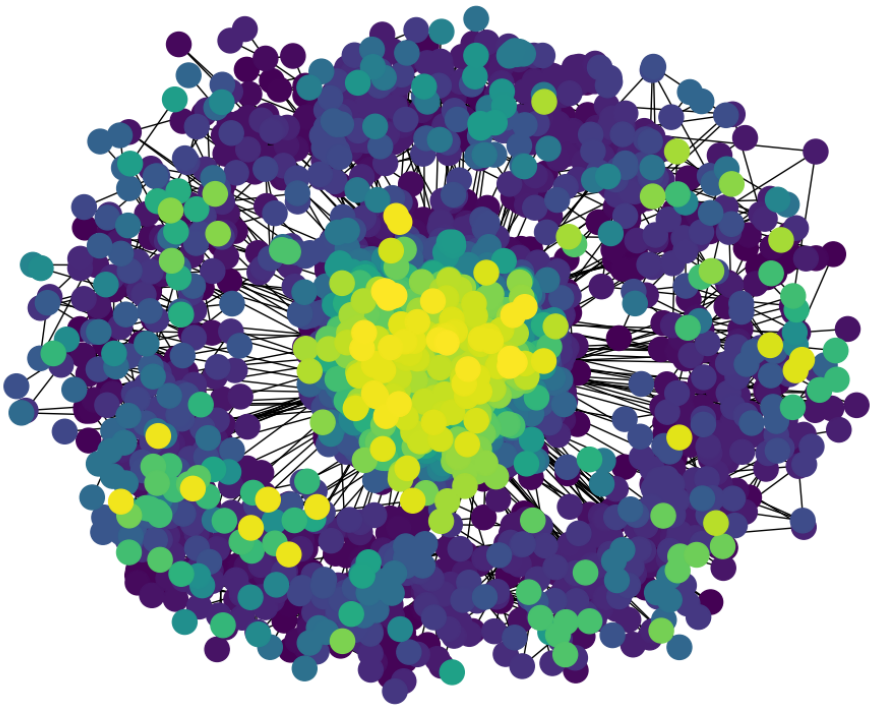


Fig 1: Greedy Modularity approach on 30,000 nodes

Louvain Community Detection

The Louvain algorithm is a hierarchical clustering algorithm that recursively merges communities into a single node and executes the modularity clustering on the condensed graphs [6]. The method allows zooming within communities to discover sub-communities, sub-sub-communities. The Louvain method is a simple, efficient and easy-to-implement method for identifying communities in large networks. The method has been used with success for networks of many different types and for sizes up to 100 million nodes and billions of links. The analysis of a typical network of 2 million nodes takes 2 minutes on a standard PC [7]. Hence, It is one of the most widely used methods for detecting communities in large networks. We used the louvain function available in NetworkX. We got 0.7 accuracy using the normalized_mutual_info_score function which is highest of all.

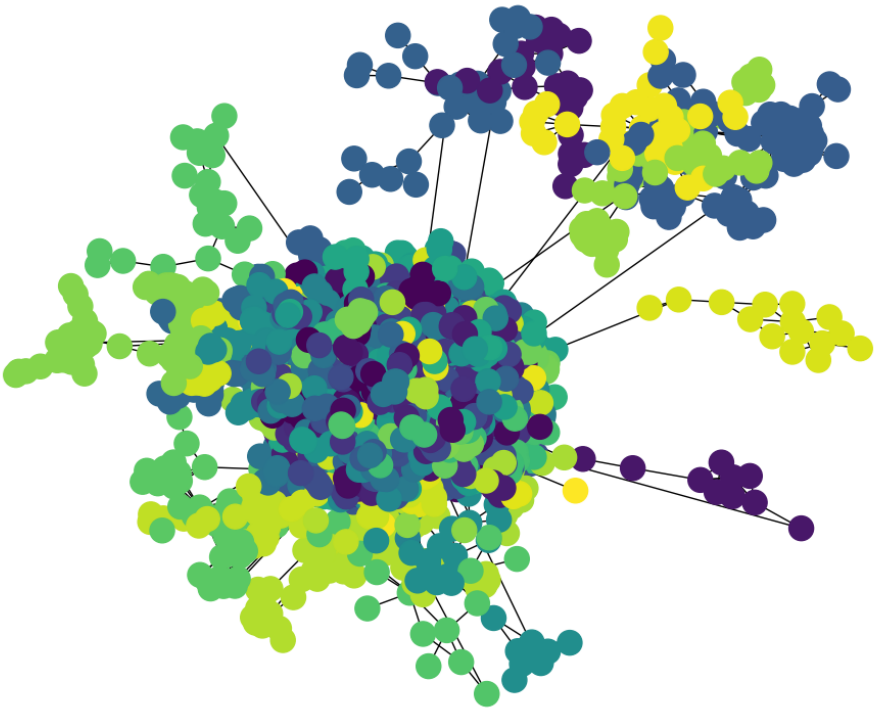


Fig 2. Louvain algorithm on 30,000 nodes

Girvan-Newman Algorithm

Our major challenge was the size of the dataset and the lack of sufficient hardware. Since we have only been running our algorithms on a local computer, we were unable to perform girvan Newman community detection algorithms on our sampled network. We again sampled our network to 10,000 nodes. Here's what it looks like for 10,000 nodes. This algorithm relies on the iterative elimination of edges that have the highest number of shortest paths between nodes passing through them. It calculates the edge betweenness centrality for every edge in the graph and then it removes the edge with the highest betweenness centrality. It again calculates the betweenness centrality for every remaining edge and it continues to remove the edges until there are no more edges left. That is why it takes a lot of time to detect the communities. This algorithm should give us the highest accuracy but considering the time it takes to detect the communities, we decided not to go ahead with the girvan newman approach. The Girvan-Newman algorithm is the slowest on each network, in line with its predicted high computational complexity. For example, the algorithm failed to find communities in the scientific collaboration network in seven days.

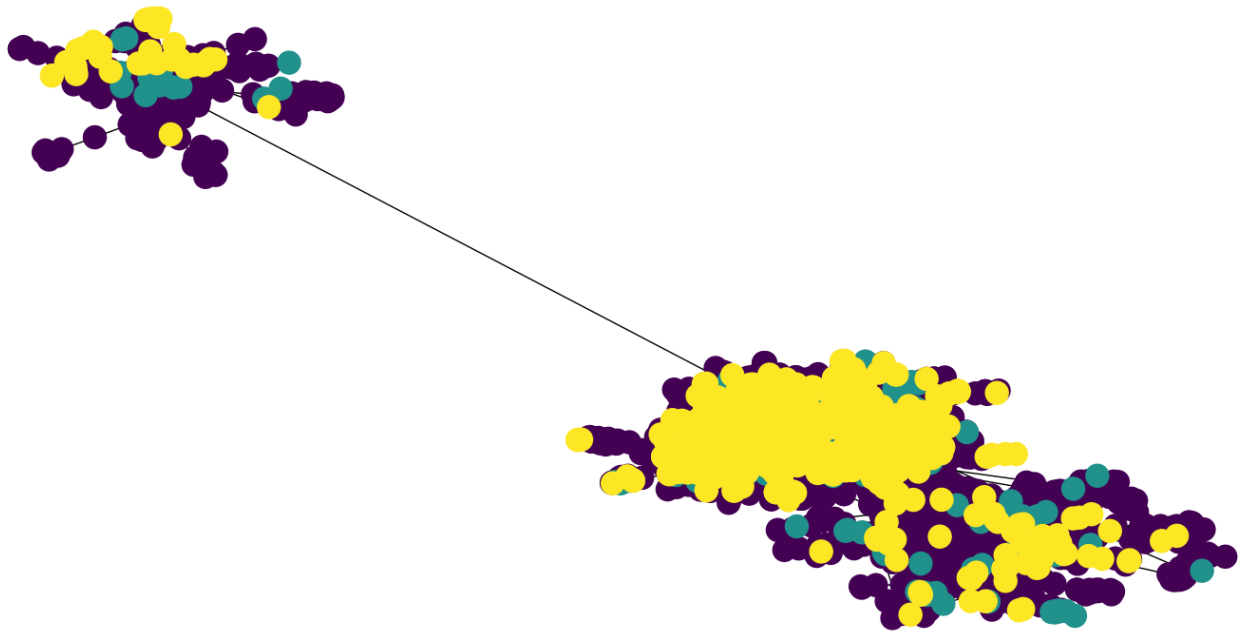


Fig 3. Girvan Newman Algorithm on 5000 nodes

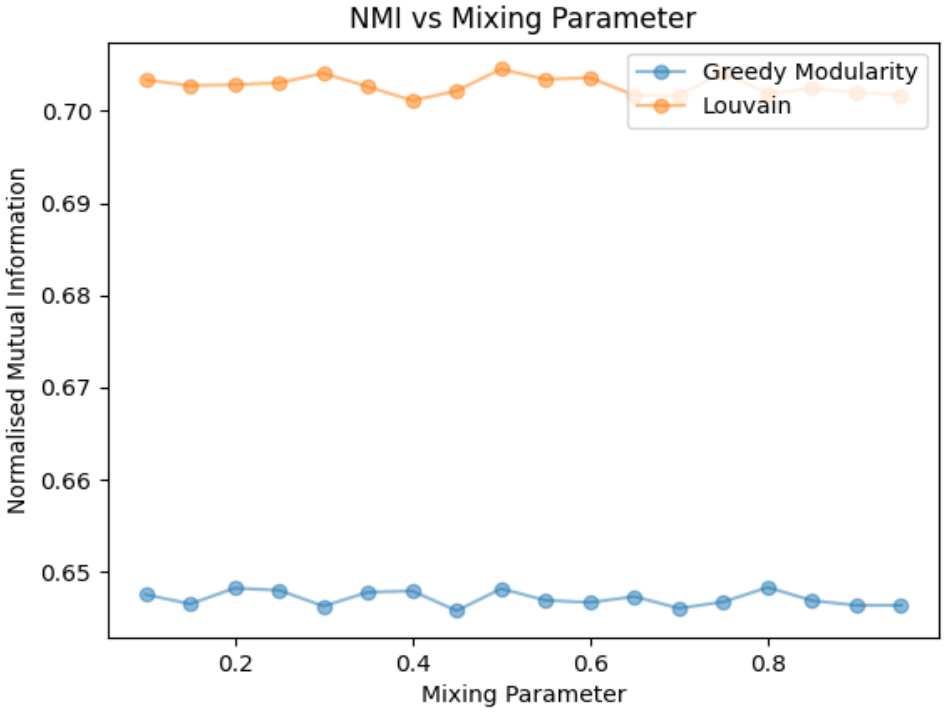
Results of Community Detection on our Network

Greedy modularity approach didn't work well on our dataset because the average size of our communities is very small. Due to the large dataset and insufficient hardwares, we could not run the Girvan Newman algorithm on our network. but results could have been better with this algorithm. We found the Louvain approach is best. This algorithm is simple and very easy to implement. It took 10 seconds to do the analysis of our network and the accuracy we got is highest of all.

In terms of speed, the Louvain community detection algorithm beat the others with the fastest completion time on our sampled network of 30,000 nodes. The Greedy modularity algorithm was fast, but not as fast as the Louvain algorithm. The worst performing of all the algorithms is the Girvann-Newman community detection algorithm. For our sampled network, this algorithm didn't finish running even after 4 days.

Results of LFR Benchmarking

Once the detection of communities were completed, our next milestone was to measure the accuracy of the community detection algorithms we used using the models generated from LFR Benchmarks. To measure the accuracy, we plot the Normalized Mutual Information (I_n) against the mixing parameter μ. The results we achieved were inconclusive and incorrect. [5]



The Normalized Mutual Information must show a sudden drop in its values as the mixing parameter increases, a sharp drop is usually observed after $\mu = 0.5$. But we did not observe any drop and the values of I_n remained almost constant. An increase in μ suggests that the number of communities start to disappear since the number of edges of a node connecting itself to nodes outside its own community increases. We believe that the plots were incorrect because of many

limitations that hindered our ability to perform the analyses on the entire network. But from the results the only observation we could make was that the Louvain community detection algorithm is more accurate at detecting communities than the Greedy modularity community detection algorithm. Since the Girvan-Newman algorithm never finished running, there was no information available to compare it with the other algorithms.

CONCLUSION

There are many community detection algorithms that have been developed that employ different procedures to detect communities. Because each community detection algorithm varies so much in its implementation, their performance also varies. It is not easy to answer the question of which algorithm is the best. Some algorithms may be good for some applications, while others may not be suitable for those applications. However, considering just the accuracy and running time of the algorithms as metrics some analyses can be made. Through our work, we realized that the Louvain community detection algorithm gave us the highest accuracy and the fastest finishing time. Though there were many limitations under which we had to operate, the Louvain community detection algorithm consistently performed better than the others.

This topic can be taken into very interesting future directions. For example, we would love to extend our analyses to some more community detection algorithms like Ravasz, Label Propagation, Multilevel, Walktrap, etc. Another interesting area to explore would be to develop a better benchmarking algorithm or system to overcome the limitations and the errors of the existing benchmarking algorithms.

CONTRIBUTIONS

The entire project was divided into 3 major parts and a few secondary parts. There was no explicit division of work but the major parts of the project had 1 major contributor and 2 secondary contributors each. Sampling, Community Detection and LFR Benchmarking were the major parts of the project.

The division and contributions are roughly as follows -

1. Idea Generation -Jui Dakhave (33%), Saurabh Kulkarni (33%), Rajath W (33%)
2. Sampling - Saurabh Kulkarni (70%), Jui Dakhave(15%), Rajath W (15%)
3. Community Detection and Accuracy - Jui Dakhave (70%), Saurabh Kulkarni (15%), Rajath W (15%)
4. Community Layout - Jui Dakhave (33%), Saurabh Kulkarni (33%), Rajath W (33%)
5. LFR Benchmarking -Rajath W (70%), Jui Dakhave(15%), Saurabh Kulkarni (15%)
6. Report - Jui Dakhave (33%), Saurabh Kulkarni (33%), Rajath W (33%)

REFERENCES

1. Sampling Community Structure, Arun S. Maiya, Tanya Y. Berger-Wolf http://arun.maiya.net/papers/maiya_etal-sampcomm.pdf
2. Towards real-time community detection in large networks, X.Y. Leung, Pan Hui, Pietro Lio', Jon Crowcroft https://arxiv.org/abs/0808.2633
3. A Comparative Analysis of Community Detection Algorithms on Artificial Networks https://arxiv.org/abs/1608.00763
4. Defining and Evaluating Network Communities based on Ground-truth https://arxiv.org/abs/1205.6233
5. Network Science by Albert-Laszlo Barabasi http://networksciencebook.com/
6. Neo4J Docs - Louvain https://neo4j.com/docs/graph-data-science/current/algorithms/louvain/
7. The Louvain method for community detection in large networks https://perso.uclouvain.be/vincent.blondel/research/louvain.html
8. NetworkX: https://networkx.org/