Performance Evaluation of Clustering Algorithms on a Network of Political Blogs

Kudsi, M; Li, Y; Nguyen, J; Rayalu, V; Stassinopoulos, A; Wang, F

December 5, 2022

1 Abstract

Recent work introduced the vast unfolding of communities in large networks, in which a heuristic methodology not only identifies communities, but also measures the density between nodes in modules that highlight the strength of a subcommunity. It was shown that such methodology can facilitate community detection, and exceed similar community detection algorithms in time complexity. In this paper, we explore 3 popular clustering algorithms, which use simple metrics like the number of common neighbors. We show how a collection of nodes with a large number of common neighbors have a higher probability of being deemed a community. The selected algorithms are the Weighted Threshold, Louvain, and Girvan-Newman. These algorithms will be tested on a political blogs network and their performance will be measured using the max intersection accuracy. The results of the study showed that the Louvain algorithm performed the best, followed by the Weighted Threshold and Girvan-Newman algorithms which had the same max intersection accuracy. Further research is needed to determine the potential applications of these algorithms in the field of network analysis since we only used one network to evaluate them.

2 Introduction

Technological innovations during the past few decades, including the rise of computers, the internet, and social media, have accelerated the size and strength of data networks. When analyzing the data behind various data networks, communities form naturally within them through connections between individual points of data, or nodes. These communities are typically defined by a common variable such as physical location, political alignment, or interest in a public figure. However, as more individual nodes of data are added to the data collection, the number of connections between nodes and the number of communities formed to represent these connections grows exponentially, creating difficult problems to overcome when analyzing the data in a timely manner.

It's important to note that grouping data has always been a problem that we have been trying to solve, and has been done through clustering algorithms, where using multiple attributes for each data entry can be used to find similarities and differences between them to create "clusters". However, the idea of locating and recovering communities is focused specifically on networks as analysis largely relies on a single attribute type - the edge. This is where the planted clique problem is presented: identifying the subset of nodes in a network that have something in common, all determined by edges. The challenge was constructing an algorithm to do so that could perform in efficient time. Methods to achieve this in polynomial time were introduced in

1995 by Luděk Kučera, and improved upon in 1998 by Alon, Krivelevich and Sudakov. Both of which proposed constraints to the size of the planted clique relative to the network, where the planted clique could be found with high probability. More recently, the paper [2] "Computational Lower Bounds for Community Detection on Random Graphs" observes that there are calculations to clearly define three bounds that determine the level of difficulty to retrieve a planted clique: simple, hard, and impossible. This prior research exposes a drawback in graph data, given that some situations cannot be optimized at all.

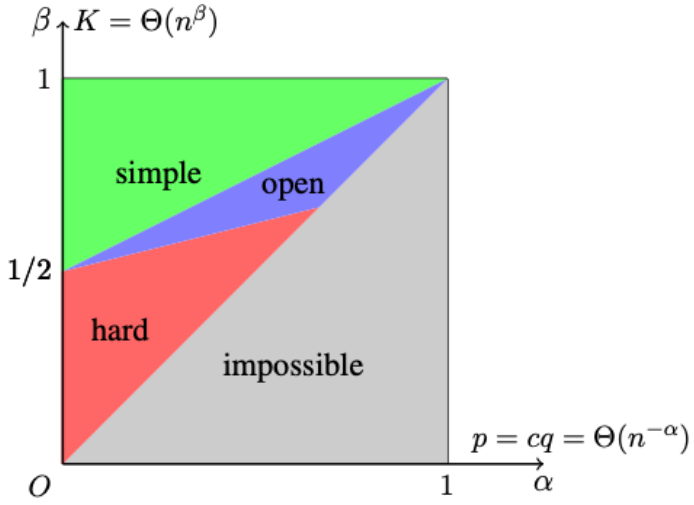

Figure 1: As seen, in [2] "The simple (green), hard (red), impossible (gray) regimes for recovering planted dense subgraphs, and the hardness in the blue regime remains open"

2.1 Definition of Community Detection Algorithms

Community detection, commonly referred to as graph clustering or network clustering, is a task in network analysis which is used to identify groups or communities within a given network. These groups are usually characteristic of dense connections between nodes within the group, as compared to the connections of these nodes to the nodes in the rest of the network.

Community detection in a network is important and interesting because it can provide useful insights to the structural organization of a network that can be applied to many diverse real-world networks. Since there is a tremendous amount of information stored in each network, if we could detect communities in each network it would provide us with important information and allow the study of the network easier. Furthermore, it could help us improve efficiency for processing and analyzing network data. For example, in social media each user is a node, and the users' friends whom they interact with form a connection and thus become a network. Social media companies could use community detection algorithms to keep people with common friends,common interests, and background tightly connected, so they could better personalize and establish a more efficient recommendation system and advertisements. By analyzing the existence of communities, we can also learn about the processes of how a network is spreading in various settings. Another useful and important application of community detection is the prediction of missing links and identifying false links in a network because of errors. By applying a community detection algorithm it would allow users to assign and fix these links.

There are many different community detection algorithms, each with their own pros and cons. In this paper, we aim to explore and analyze the performance of 3 algorithms used for community detection on a single dataset. The algorithms we have chosen are:

- **Weighted Threshold algorithm**[6], which uses a weight threshold to detect high-quality communities in networks with spread out edge weights
- **Louvain (Modularity) algorithm**[4], which works by recursively optimizing a modularity measure to identify communities.
- **Girvan-Newman algorithm**[5], which uses 'edge betweenness' to identify communities with clear boundaries in networks.

These algorithms were chosen since they represent a range of different approaches to community detection and have also been extensively used in research. We will apply each of these algorithms on the same dataset, and use a fixed metric which measures the ability of each algorithm to identify communities accurately. The results of our analysis will be reported in the paper, along with a discussion of why a certain algorithm may have performed better. We hope that our work will provide insights into the strengths and weaknesses of the selected algorithms, and will help guide future research on community detection.

2.2 Overview of Dataset

The dataset[1] we chose contains a directed network of hyperlinks between weblogs on US politics from the year 2005, created by Lada A. Adamic and Natalie Glance. These hyperlinks were taken from top blog websites on politics which contained links to other blogs. The dataset consists of a set of nodes which represent the individual blogs, and a set of directed edges representing the links between the blogs. We chose this dataset to assess how the different selected algorithms perform at identifying communities from a real world dataset. The network contains a total of 1490 nodes and 19090 edges. The node values indicate political leaning as follows:

Label	Significance
0	Left-wing or Liberal
1	Right-wing or Conservative

Table 1: Dataset value labels

The actual number of nodes in the left and right wing parties are 758 and 732, respectively. A paper written by the creators of this dataset (Adamic and Glance) highlights how one of the main differences they found between the two communities or parties is how the right wing blogs had significantly more links to each other than the left wing. After performing some exploratory analysis on the dataset, we can see that this is true since the in-group degree of the conservative (right wing) blogs is far larger than the in-group degree for the liberal blogs. The in-group degree is a measure of the connections a particular node has with other nodes in the same community. The red points on Figure 1 indicate nodes that belong to the left wing community, and the purple nodes belong to the right wing community.

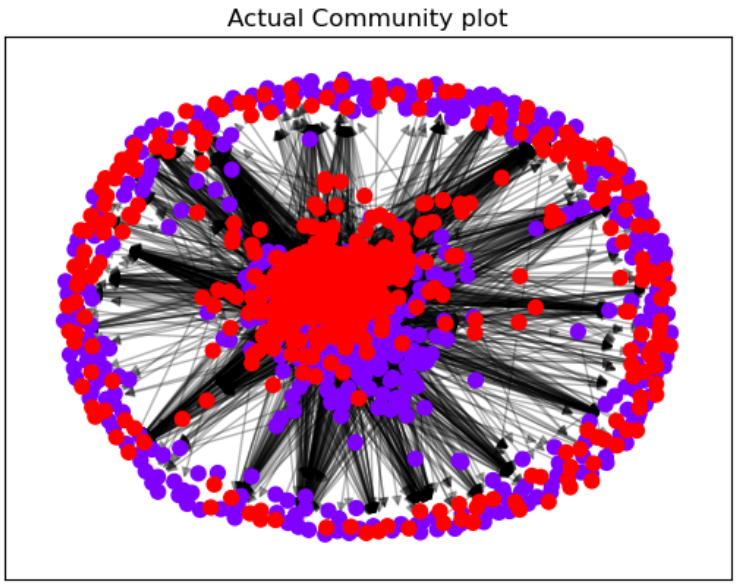

Figure 2: A visualization of the graph.

Label	Number of nodes	In-group degree	In-group probability	Out-group degree	Out-group probability
0	758	16816	0.029306	1688	0.003042
1	732	17988	0.033617	1688	0.003042

Table 2: Descriptive graph statistics of the blog dataset

3 Algorithm Descriptions

3.1 Weighted Threshold

The weighted threshold algorithm synthesizes both Depth First Search (DFS) and common neighbor counting to detect algorithms. Determining whether a pair of nodes belongs in the same community depends on a weighted threshold calculated from their degrees. The threshold is determined beforehand as a parameter input by the user, then it is used to decide if the number of common neighbors between two nodes is sufficient enough to categorize the two as being in the same community. This process is repeated for each pair of nodes until all nodes have been categorized into a community.

3.2 Louvain

The Louvain community detection algorithm is based upon a modularity approach. The algorithm compares the actual number of edges in a community to the expected number of edges in a community. The algorithm uses a recursive format to achieve maximum accuracy of community detection.

The two step process assigns nodes to communities, then iteratively using a modularity approach to evaluate whether moving a node to another community has a positive increase in accuracy. If the algorithm detects a higher accuracy ("gain"), then the node is kept in its new community and the algorithm moves onto the next iterative node. However, if the algorithm detects a lower accuracy ("gain"), then the node is kept in its current community.

This process repeats recursively until the accuracy ("gain") of the model is no longer improved per recursive iteration.

3.3 Girvan-Newman

The Girvan-Newman algorithm is a heuristic algorithm used to detect communities. It works by finding and removing the edges between the most central nodes in a network, such that the network forms smaller broken down communities. The central nodes in a network would be the nodes through which most other nodes need to pass through in order to find the shortest path to different nodes. This process of removing the edges between the most central nodes is continued until the desired number of communities is reached.

The algorithm uses a measure called edge betweenness which is how often an edge in a network lies on the shortest path between two nodes. The Girvan-Newman algorithm first calculates the edge betweenness of each node in the network. Nodes with a higher edge betweenness are considered to be more important to connect different parts or clusters of the network.

The algorithm then removes the edge of the node with the highest edge betweenness and breaks the network into two communities. The algorithm then recalculates the edge betweenness for the remaining nodes and edges and removes the next edge iteratively until the network is divided into the required number of communities.

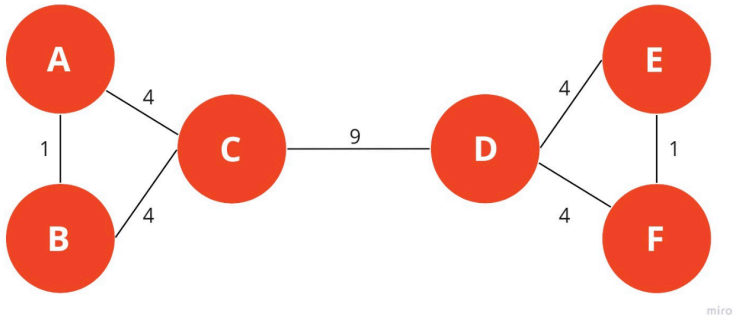

Figure 3: A visualization[3] of a sample graph to explain the Girvan-Newman algorithm.

In this figure, the numbers next to each edge are the edge betweenness scores. So at this point in the algorithm (after the scores are calculated), the edge between the nodes C and D would be removed to form two communities on the left and right sides.

4 Results

Using the Political Blog dataset, we were able to get the following results:

Algorithm	Accuracy
Girvan-Newman	0.6067
Louvain	0.9584
Weighted Threshold	0.6067

Table 3: The results of our analysis

4.1 Performance Metric for the Algorithms

We decided to use a metric called the **maximum intersection accuracy** which is a measure of how well a clustering algorithm has identified the correct number of clusters and correctly assigned nodes to their actual communities. A high max intersection accuracy means that the algorithm has performed well in identifying the structure of a network, while a low max intersection accuracy indicates that the algorithm may have failed to accurately identify the communities. The max intersection accuracy can be used to effectively compare and evaluate the performance of the three selected algorithms.

4.2 Performance of the Weighted Threshold Algorithm

The algorithm yielded a performance of about 60.6% for max intersection accuracy. This is no surprise, considering the fact that the dataset has an inner group probability of 0.0315, meaning there are less connections between nodes of the same cluster. Inevitably though, this is on par with the Girvan-Newman algorithm but not as efficient as the Louvain algorithm.

4.3 Performance of Louvain Algorithm

The Louvain algorithm yielded a performance of 95.9% for max intersection accuracy. This is significantly higher than both the weighted threshold algorithm and the Girvan-Newman algorithm. Therefore, we can conclude that for this particular network that we tested the algorithms on, the Louvain algorithm is the most effective.

4.4 Performance of Girvan-Newman Algorithm

The Girvan-Newman algorithm resulted in a max intersection accuracy of 60.6%. The max intersection accuracy was the same as the performance of the weighted threshold algorithm which could imply that both the algorithms are similar. It is possible that both these algorithms had similar performance since both algorithms use some form of a score related to the connectedness of nodes in a network. In the case of the Girvan-Newman algorithm, this score is the betweenness centrality of the edges, while the weighted threshold algorithm uses a score based on the weights of the edges.

5 Discussion

5.1 Strengths and Weaknesses of Each Algorithm

5.1.1 Weighted Threshold

The weighted threshold algorithm depends on a high probability of connections within clusters because of its reliance on counting common neighbors. If the probability of a connection between nodes of the same cluster is low, that will inevitably result in the cluster also having a lower chance of having common neighbors between nodes. The weighted threshold's relatively low accuracy is likely due to the fact that the political blogs network has an extremely low intra probability of 0.0315. If the in group probability were higher, the weighted threshold algorithm would likely perform better.

5.1.2 Louvain

The Louvain algorithm's largest benefit is the ease of implementation and time efficiency of the algorithm to run. There are limited tweaks to be made to the model, which allows a user to easily implement it and optimize its runtime.

On the other hand, the Louvain algorithm's downfall is its high usage of system storage throughout the running of the model. This is due to the need to store communities and duplicate nodes between communities while running the recursive algorithm and placement of nodes.

5.1.3 Girvan-Newman

The main benefit of using the Girvan-Newman algorithm is its simplicity. The algorithm only requires the calculation of the edge betweenness for each node and the removal of the edge with the highest edge betweenness. The formula to calculate the edge betweenness is also extremely trivial and easy to understand. This makes the algorithm particularly easy to implement and understand, even for people with little to no knowledge of graph theory.

Another strength of this algorithm is its versatility and ability to be applied to a wide range of network types and sizes. It is a popular and well documented algorithm which has been widely used in research for the same reason.

Despite its simplicity, running the Girvan-Newman algorithm on extremely large datasets would be computationally intense and hence, not ideal. Even though the calculation for the edge betweenness is simple, calculating this for every single node in a network would take time and resources. This can make the algorithm impractical to use on large datasets. Since the selected dataset had only 1490 nodes, the results of running the algorithm were almost immediate.

Another limitation of the Girvan-Newman algorithm is its sensitivity to noise. The algorithm solely relies on the calculation of the edge betweenness, which can severely be affected by the presence of outliers or noisy data. This can lead to the formation of false communities and ultimately result in a worse accuracy.

5.2 Comparison of Results

Both algorithms, Girvan-Newman Algorithm and Weighted Threshold, use these scores to identify and remove edges that are important for connecting different parts of the network, leading to the formation of clusters or communities. This could be why they both have the same max intersection accuracy in this particular case. However, it is important to note that the performance of these algorithms can vary depending on the specific dataset and the parameters used. Therefore, it is not always accurate to conclude that the algorithms are similar just based on their max intersection accuracy.

The issue with the weighted threshold algorithm is, as stated before, that it relies on a high inner probability value to best take advantage of common neighbor counting. Because this dataset has a small intraprobability of 0.0315, the algorithm does not perform as well as one might expect, though still satisfactory, as it matches the performance of the Girvan-Newman algorithm.

Problem with Girvan-Newman is that it is not ideal if we do not know the required number of communities, this can result in poor performance. After all, it is simple, but the lack of dependence on further information can yield an incorrect number of communities without that constraint.

6 Conclusion

6.1 Summary of Findings

After testing different community detection algorithms on one dataset, the following findings were observed:

1. The Louvain algorithm was found to be the most effective in identifying and dividing the network into clusters, with a max intersection accuracy of 95.9%
2. The weighted threshold algorithm performed average, with a max intersection accuracy of 60.6%.
3. The Girvan-Newman algorithm had the same accuracy as the weighted threshold algorithm.

Overall, these findings suggest that different community detection algorithms can have varying performance depending on the specific dataset and parameters used. It is important to carefully evaluate the performance of different algorithms and select the one that is most appropriate for the specific application and dataset. As stated before, for this particular network of weblogs and hyperlinks, the Louvain algorithm was the most well-suited. Further testing and analysis are needed to determine its overall effectiveness in different scenarios.

6.2 Recommendations and Implications for Further Research

Looking forward to future research, it's important to take what we have learned regarding the accuracy of each model and put it into practice and testing with larger datasets. Furthermore, it is recommended to run a comparison test using more than the two comparison communities (left wing vs right wing) that we utilized in this paper. These models and their accuracy are tested and maximized on comparing two communities, in which we found the Louvain algorithm to have the highest accuracy in predicting communities. However, with an increase in communities the other models may perform better or worse. Therefore the number of communities is one factor to test in future research.

Furthermore, it would be interesting to include new models in comparison. Although we identified Louvain had a high accuracy, there may be other models that perform better that weren't taken into consideration in this paper, such as the "Surprise Community Detection" model in the same NetworkX package as the three models researched in this paper.

Taking a different approach, it may be interesting to take the Louvain algorithm and look at large datasets with multiple communities, and use the algorithm to identify unique communities. This is especially interesting in today's world of social media applications and the data they collect. For example, one could find it interesting to utilize Louvain's algorithm to identify communities in Tiktok to explore existing and possibly new minute communities. We have only scratched the surface into community detection algorithms, it is up to the scientific community to take our learnings to both leverage and improve them.

References

[1] Adamic, L. A., & Glance, N. (2005). The political blogosphere and the 2004 U.S. election. Proceedings of the 3rd International Workshop on Link Discovery - LinkKDD '05. https://doi.org/10.1145/1134271.1134277

[2] Hajek, B., Wu, Y. & Xu, J.. (2015). Computational Lower Bounds for Community Detection on Random Graphs. Proceedings of The 28th Conference on Learning Theory, in Proceedings of Machine Learning Research 40:899-928 Available from https://proceedings.mlr.press/v40/Hajek15.html.

[3] Jayawickrama, T. D. (2021, February 1). Community detection algorithms. Medium. Retrieved December 4, 2022, from https://towardsdatascience.com/community-detection-algorithms-9bd8951e7dae

[4] Louvain. Neo4j Graph Data Platform. (n.d.). Retrieved December 4, 2022, from https://neo4j.com/docs/graph-data-science/current/algorithms/louvain/

[5] Memgraph. (n.d.). Girvan-Newman algorithm. NetworkX Guide. Retrieved December 4, 2022, from https://networkx.guide/algorithms/community-detection/girvan-newman/

[6] Yan, X., Jeub, L. G., Flammini, A., Radicchi, F., & Fortunato, S. (2018). Weight Thresholding on complex networks. Physical Review E, 98(4). https://doi.org/10.1103/physreve.98.042304