Evaluation and Finding Similarity of Community Detection Algorithms Under Real-world Networks

Ping-Chang Lee
Candidate Number: 10001

May 1, 2022

Contents

1 Description of Topic

1.1 Overview

In recent decades, the importance of community detection has drawn the attention of scientists due to the rise of social network analysis. Additionally, more and more researchers in other fields such as physics and medical field start to apply graph structure to their experiments. For instance, a group of medical researchers has illustrated how proteins structure and accumulate to form a larger compound by introducing community detection concept (13). In this proposal, I will introduce two types of community detection algorithms, approaches to evaluate the result of communities computed by the algorithms, and finally discuss how to find the similarities between algorithms qualitatively.

1.2 Motivation

Rather than data points separated in dimensional space, that is, let the collection of real-world data points be S, where $S \subseteq R^n$ and $n \in N$, the network we will focus on is the a set of vertices and a set of edges. The structure of network is basically identical to the structure of a graph. Therefore, I will use the word network and graph interchangeably. Given a graph (or a network) $G = \{V, E\}$, where V is the vertex set and E is the edge set, we would like to find out if we can partition the graph into a collection of communities. There is no rigorous and universal definition of community in graph theory; however, informally, a community is a subset of $V(G)$ such that the connection between the vertices in the same community is closer to other vertices outside the community. The data structure of a network is beneficial for graph analysis since the researchers can solely work on the relation of vertices and edges without considering the whole space. Additionally, as for the network, I will run the community detection algorithms on both real-world networks and randomly generated networks in the research. However, to focus on the primary goal of the research, the application of randomly generated networks will discourse in the miscellaneous section.

Throughout the study, I plan to apply the community detection algorithms to real-world networks with ground truth labels. With the computed result, I will be able to compare the difference between the outcome of the algorithms and the ground truth. As for the difference mentioned previously, the greater the difference does not automatically imply the worse the algorithm is; it is not more than suggesting that such an algorithm may not be suitable for the network.(2) Therefore, I would like to investigate how a result of an algorithm differs under multiple standards. The investigation is beneficial for future researchers to prevent misinterpreting the outcome of an algorithm due to a lack of comprehensive evaluations.

1.3 Research Aims and Objectives

The research aims to choose the community detection algorithms that are popular in industry or in academia , apply them to a group of real-world networks, and evaluate them based on the goodness of community partitioning or the betweenness of vertices in the same community.

The first aim can be achieved by studying papers (3) (2) that discuss and analyze the algorithms in depth. By completing the first aim, I expect to understand the concepts of the community detection algorithms chosen to be presented in the research and to apply the algorithms to the real-world network. As for the real-world network, I will choose the data from the Stanford Network Analysis Project(SNAP) . Their networks are up-to-date from various sources such as Facebook, Amazon, road maps, etc. By conducting algorithms on these networks, the computed result can be more approachable and insightful than working on a fully randomized network. The final aim is to learn the approaches to evaluate the quality of community partitioning; this can be achieved similarly to the previous aim by reading related papers and websites (4) (5).

2 Literature Search

This section is about how I conducted the literature search, the searching history while looking for useful papers for the study, and how I evaluate the source found in various databases. Before conducting the literature search, I read several papers provided by my supervisor, Professor Luca Zanetti. These papers mainly discuss network analysis, semi-supervised learning (SSN), and graphical neural network(GNN). After a short period, I decided to focus on studying the community detection algorithms under the topic of network analysis. The Paper (3) provides various approaches to detecting communities, and it is not hard to learn the concepts. Also, the referenced papers are related to my study (1; 2; 9); this helped me dramatically while doing literature search. Other than reading papers referenced from the paper mentioned, I also used IEEE Xplore, arXiv,and Google Scholar with keywords that are associated with my research interest. The following section is the result of my literature search.

2.1 Sources and Databases

Name of Source	Search Statement(s)	Returns Found	Notes
IEEE	Graph & Community	4301	Too much to look at.
IEEE	Network Analysis	408129	Too much to look at.
IEEE	Community Detection Algorithms	389	Found (7) sorted by relevance.
IEEE	Evaluation & Community Detection Algorithms	301	Found lots of relevant papers, can be used for future study.
IEEE	Girvan & Community Detection	33	Most papers proposed variant versions of Girvan-Newman's algorithm on all kinds of real-world networks, for instance, social network and protein structure.
IEEE	Quality Function & Community Detection	119	Bad searching criteria. Most related papers based on the searching statements are irrelevant to my study interests.
IEEE	Louvain Algorithm	842	The search result is immense. However, quite a few papers mention the idea of parallel Louvain-based algorithms. Found (8) which talks about the improvement of quality function in Louvain algorithm.
IEEE	Similarities & Community Detection Algorithms	373	The result is mostly about considering similarities between communities into community assignment.
Arxiv	Community Detection Algorithms & Evaluation	409	The papers mostly focus on its application and the evaluation of their experiments.
Arxiv	Community Detection & Modularity	366	Most papers are published after 2015.
Arxiv	Modularity & Network Analysis	45	Fewer than the previous search.
Arxiv	Betweenness & Network Analysis	5	Found papers introduce betweenness centrality as quality function as part of the community detection algorithms.
Arxiv	Benchmark & Community Detection & Network Analysis	68	Most papers are focus on the application of community detection algorithms to their very specific dataset.

Google Scholar	Community Detection Algorithms	over 2,000,000	The amount of related work is too board.
Google Scholar	Community Detection Algorithms & Evaluation & Modularity & Ground-Truth Data	495	Found this paper (11) which is helpful for future algorithm evaluation.
Google Scholar	Community Detection Algorithms & Similarity	230,000	Very broad searching result. Found (12) for its qualitative comparison of community detection algorithms.

2.2 Findings

To determine the level of relevance of a paper to my study, I first checked the number of cited and the publisher. By doing so, we can ensure the credibility of the paper. Afterward, I read through the topic, abstract, and content, respectively, searching for keywords such as community detection, modularity, betweenness, and network analysis. At this point, if a paper meets the requirement listed above, I then screen the body of the paper and observe if the context is relevant to any of my research interests.

Same as mentioned in the overview section, community detection is not extensive studied until the recent decades. According to the result of literature search, most of the papers are published after 2015. Also, Louvain algorithm and Girvan-Newman's algorithm are frequently applied in their experiments, showing that the above two algorithms are one of the most popular community detection nowadays. There are also a lot of papers mentioning the idea of "speeding up" the computation time of the above algorithms, showing the sign that the algorithm can be time consuming and lack of optimization structural-wise. In addition, learned from the literature search, most of papers mentioned that the quality functions, known as the evaluation functions, can often be misleading despite the fact that these functions are able to successfully determine if a community is "good" in majority cases. (2)

As for finding the similarities of community algorithms, due to the immense outcome of related paper according to the literature search, I could not review the work done by the researchers thoroughly. However, one paper was found discussing the similarities between community detection algorithms by imposing a community detection algorithm on the result of the evaluations done by the community detection algorithms(10). Furthermore, the paper (10) exhibits the association between each community detection algorithm in a network, suggesting that any algorithm in the same communities may share something in common and hence potentially disclose the latent information behind the algorithms that seem different and yet they are similar in certain aspects.

3 Literature Review

In this section, I will introduce two different types of algorithms; they are Girvan-Newman's algorithm and Louvain algorithm. Additionally, several ways of evaluation will be performed according to the result of literature search. More categories of algorithms will be introduced to the research as the study continues .

3.1 Divisive Algorithm: Girvan-Newman's Algorithm

Among other community detection algorithms, Girvan-Newman's method (2) is one of the most popular algorithms. The Girvan-Newman's method is a iterative algorithms, the procedure is as follows:

1. Calculate the edge betweenness of every edge in the network.
2. Remove the edge with the highest edge betweenness.
3. Repeat step 1 and step 2 until each node in the network is a unique community.

Given a weighted and undirected graph G = {V, E}, where V is the vertex set and E is the edge set. The Girvan-Newman's method provides a new approach to evaluate the importance of an edge; if an edge is important, or with high betweenness centrality, then such edge has a great chance to be part of the shortest path between node in different community. The definition of betweenness centrality applied in Girvan-Newman's paper(2) is as follows:

$$B(v) = \sum_{s \neq v \neq d \in V} \frac{\sigma_{sd}(v)}{\sigma_{sd}},$$

where $B(v)$ is the betweenness of vertex v, s is the start vertex of the path, d is the destination vertex o f the path, σ_{sd} is the count of shortest path found from vertex s to vertex d. Intuitively, the higher betweenness value of an vertex v, the higher the chance that the v also connects to other community; removing v can therefore lower the possibility that vertices in one community be assigned to other irrelevant communities.

Due to the nature of the algorithm, the algorithm treats the network as a whole community initially, and divide them into unique communities after numerous iterations; such technique is also called divisive algorithm(3). To determine which iteration has the best community division, a quality function defined in the paper (3) is as follows:

$$Q = \sum_i (e_{ii} - a_i^2),$$

where e is a k by k symmetric matrix, k indicates the number of the community in the current iteration, e_{ij} is the fraction of the count of vertices in community i to the count of vertices in community j (3), and $a_i = \sum_j e_{ij}$.

With the betweenness centrality function and the quality function(or so-called modularity), the researchers can therefore be able to determine which outcome of the iteration of the algorithm has the best modularity. However, the average time complexity of the algorithm is $O(mn(n+m))$. The worst time complexity of the algorithm is $O(n^3)$ when applied to a sparse graph(a graph with the number of edges that is far less than the number of vertices). With such an expensive time cost, we are not advised to conduct the algorithms on networks that have a lot of vertices(3).

3.2 Aggregative Algorithm: Louvain Algorithm

Unlike Girvan-Newman's algorithm initially sets the entire network as a community and divides them into smaller communities iteratively, the Louvain algorithm treats every vertex in the network as a community and aggregates them based on the change of modularity. There are various modularity function for different networks, assume our network is undirected and unweighted, then the modularity function is

$$M = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] C_{ij},$$

where $m = |E|$, A is the adjacency matrix, k_i is the degree of vertex i, and C is the community matrix; when vertex i and vertex j are in the same community, $C_{ij} = 1$, $C_{ij} = 0$ otherwise. The purpose of the modularity function is to determine of the current structure of the network is consisted of well-partitioned communities. Normally, $M \in [-0.5, 1]$, the higher the better. A good real-world networks community partitioning will normally have around 0.7 modularity indicator(12).

The procedure of Louvain algorithms is as follows:

1. Let every vertex in the network be in its own communities.

2. For every vertex i, iterate through every communities in the network, say c. Try to find a c that maximises the change of the modularity, that is , ΔM. ΔM can be calculated using the following equation:

$$\Delta M_{i \to c} = [\frac{k_{i,in}}{2m} - \frac{\sum_{tot}^{c} k_i}{2m^2}],$$

where $\Delta M_{i \to c}$ is the change of modularity that assigning vertex i to community c, $k_{i,in}$ is the number of edges that i connects to other vertices that are also in the same community, and $\Sigma_{tot}^{c} k_i$ is the sum of the count of edges that connects to any vertex that is in c.

3. If c is found and $\Delta M \geq 0$, then assign vertex i to community c.

4. Repeat step 2 and 3 until there is no significant change to the modularity.

The algorithm has proven its exceptional performance in many researches(8). However, since the modularity optimization (step 2) is an NP-complete problem and the nature of greediness of the algorithm(9), the computation cost can be extremely expensive. In this paper (8), there are several ways to reduce to time and memory cost of the algorithm by conducting parallel modularity computation and hash-based memory allocation. The improved version of the Louvain algorithm can be helpful for my study.

3.3 Approaches of Evaluation

One of the most intuitive approach of evaluation is the calculation of mean square error, the equation is as follows:

$$Q = \sum_{i}^{k} \sum_{j}^{n} C_{ij} * |v_j - \bar{v}_i|^2,$$

where k is the count of the communities in the network, $v_j \in V(G)$, $\bar{v}_i$ is the arithmetic mean of all the vertices in community i, and C is the community matrix, when vertex j is in community i then $C_{ij} = 1$, $C_{ij} = 0$ otherwise. The method (4) requires the numerical coordinates for every vertices in the network, it may not always be the case in every datasets, take friendship network for instance. However, it is a good initial evaluation to start nonetheless.

The second evaluation method is modularity which has high similarity to what we discussed in the Louvain algorithm section. However, the modularity optimization approach can sometimes lead to over-merging, merging small communities with larger communities to increase the change to modularity. Over-merging often leads to the wrong estimation with both the structure and the number of communities (4; 6). Note there is another variant of modularity quality function defined by Newman (2) as:

$$Q(C) = \sum_{c \in C} \left[\frac{E(c)}{m} - \left(\frac{Vol\,(c)}{2m} \right)^2 \right],$$

where E(c) is the count of edges in community c, Vol(c) is the sum of degree of vertices in community c. The method is able to evaluate if a group of vertices are formed into a community unexpectedly without drawing any assumption to the structure of the network. (7)

The third evaluation method is conductance quality function, the concept of the method is to compute the probability of a vertex exit its corresponding community with random walk.

$$\phi(C) = argmin_{c \in C} \frac{|(u, v) \in E, u \in c \ \wedge \ v \notin c|}{argmin(Vol(c), 2m - Vol(c))},$$

where Vol(c) is the sum of the degree of vertices in community c and m is the count of edges in the network. An ideal network in which every community is perfectly partitioned will lead to the conductance of every community inside the network being 0, implying that there is no change for any vertex inside one community to join another community. However, due to the nature of the evaluation function, the performance under the aggregation method can be misleading.(7) If there exist satellite vertices (vertices with no neighbors) in network , these vertices will turn into one-vertex communities, which is obviously not practical.

4 Timeline of Work Plan

The ultimate goal of the research is to find various approaches to evaluate community detection algorithms in different circumstances and thus be able to discover similarities between community detection algorithms despite the fact that they are not the same based on the structure of the algorithms. For datasets, I plan to apply the network collection from Stanford Network Analysis Project(SNAP) for two reasons:

1. The provided networks cover various properties. They can be (un)directed, (un)signed, and (un)weighted.

2. Some of the networks are with ground-truth communities, meaning that such networks are far easier to evaluate the effectiveness of the community detection algorithms compared to those without ground-truth communities.

In this section, I divide the research into 3 objectives to accomplish.

4.1 Objective 1: Community Detection Algorithms and Networks

For objective 1, I plan to spend approximately four weeks establishing the algorithms I want to discuss and applying them to networks from SNAP. The detailed timeline is as follows:

Task	Description	Duration(week)	Dependencies
1	Establish the community detection algorithms that will be on the research	$\frac{1}{2} \sim 1$	-
2	Implement the algorithms to networks in Python.	1	Completion of Task 1.
3	Determine which network from SNAP will participate the research	$\frac{1}{2} \sim 1$	-
4	Review the Papers and verify the suitability of algorithms and networks introduced in this objective.	1	Completion of Task 1 & 2 & 3.

4.2 Objective 2: Evaluation and Finding Similarities

For objective 2, I plan to spend $3 \sim 4$ weeks reviewing and implementing means of evaluating community detection algorithms on different networks. It is foreseeable that not all the networks selected in task 2 may take into consideration when evaluating the algorithms.

Task	Description	Duration(week)	Dependencies
5	Review papers about evaluating algorithms.	$\frac{1}{2}$	-
6	Evaluate the effectiveness of algorithms introduced in task 5.	$\frac{1}{2} \sim 1$	Completion of Task 5.
7	Review papers and determine the method to discover the similarities between algorithms	1	-
8	Integrate the result of task 6 and approaches learned from task 7 to discover the similarities. Further insights and analysis can be brought to the last objective.	1	Completion of Task 6 & 7.

4.3 Objective 3: Summary and Miscellaneous

For the last objective, I estimate the total duration will be four weeks. The objective will mainly focus on two key points:

1. Summarizing the result granted in the first two objectives.

2. Introducing an alternative option of data source, random networks, to provide readers a new perspective that networks can be artificially generated rather than can only be gathered from real-world data.

Task	Description	Duration(week)	Dependencies
9	Summarize the results from task 6 and 8. Discuss the alternative interpretation of the result.	1	Completion Task 6 & 8.
10	Review papers regarding to concepts of random networks.	$\frac{1}{2} \sim 1$	-
11	Evaluation and Application on random networks.	$\frac{1}{2} \sim 1$	Completion of Task 10.
12	Polish the writings done in previous tasks and adjust the structure of the research if necessary.	1	Completion of Task 1 to 11.

Note that the tasks are not arranged chronologically, except for those with specific dependencies. The total duration of the three objectives can be reduced to less than ten weeks if the schedule is deliberately arranged. The completion of all the tasks can result in achieving the goal of the research, that is, successfully evaluating and finding the similarities between community discovery algorithms under designed circumstances.

References

[1] *Graph Theory, 1736-1936*, Biggs, N., Lloyd, E.K. and Wilson, R.J., 1986.

[2] *Finding and evaluating community structure in networks*, M. E. J. Newman and M. Girvan, 2004

[3] *Community detection in graphs*, Santo Fortunato, 2010

[4] *Evaluation of Community Detection Methods* Liu, X., Cheng, H.-M. and Zhang, Z.-Y, 2019

[5] *Community Structure: A Comparative Evaluation of Community Detection Methods* V. L. DAO, C. BOTHOREL, and P LENCA, 2019

[6] *Performance of modularity maximization in practical contexts* B. H. Good, Y.-A. de Montjoye, and A. Clauset, 2010

[7] *Finding compact communities in large graphs* J. Creusefond, T. Largillier, and S. Peyronnet, 2015

[8] *Scalable Community Detection with the Louvain Algorithm* X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, 2015

[9] *Maximizing modularity is hard* U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner, 2006

[10] *Discovering Communities of Community Discovery* M Coscia, 2019

[11] *Community detection algorithm evaluation with ground-truth data* M. Jebabli,c, H. Cherifi , C. Cherifi ,and A. Hamouda, 2018

[12] *Qualitative Comparison of Community Detection Algorithms* G. K. Orman, V. Labatut , and H. Cherifi

[13] *Parallel Protein Community Detection in Large-scale PPI Networks Based on Multi-source Learning* J. Chen, K. Li, K. Bilal, A. A. Metwally, K. Li, and P. S. Yu, 2018