Community Detection Algorithms: Implementation and Performance Evaluation in Python

Wang Shiqin
1023040807
Nanjing University of Post and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—With the advent of the big data era, community detection algorithms play a crucial role in network analysis and social network mining. This study aims to implement and compare several classic community detection algorithms, including KL, GN, FN, LPA, SLPA, COPAR, Louvain, and LFM algorithms, to delve into their performance on different datasets. This paper provides a detailed introduction to the principles and implementation details of each algorithm, followed by their application to two different social network datasets using the Python3 programming language. In the experiments, we focus on the efficiency, modularity, and adaptability of each algorithm in the context of big data. Through extensive performance evaluation and comparison, we observe variations in the performance of different algorithms on diverse datasets, indicating their adaptability to different network topologies and scales. Additionally, from the perspective of big data analysis, we explore the feasibility and efficiency of these algorithms in handling large-scale social networks, as well as their impact on the accuracy and stability of community structures. Our research not only provides a profound understanding of community partition algorithms but also offers valuable experiences and insights for social network analysis in the context of big data.**

keywords-Community Detection Algorithms, Network Topologies, datasets, Social Network Mining, Modularity

I. Introduction

Complex networks have emerged as a cross-disciplinary field in recent years, with an increasing number of problems being represented through the development of complex network theory. Complex networks have become a tool for problem-solving. Community partitioning originated from graph segmentation in computer science and hierarchical problems in sociology. Since Givan and Newman [1] introduced the concept of community structure in complex networks, community detection has become a hot topic in the field.

In 2002, Girvan and Newman proposed a community partitioning algorithm based on community splitting: the GN algorithm. By continuously removing edges with the highest edge betweenness, the GN algorithm can clearly display the hierarchical structure of a network. However, the GN algorithm has a flaw: it lacks an effective termination condition, resulting in a hierarchical clustering tree for the graph without knowing which layer is the optimal partition.

To address this issue, Newman and Girvan [2] introduced a function in 2004 to quantify the strength of community structure—the modularity function Q. They believed that if a network has a clear community structure, comparing it with the corresponding random network, the probability of connections within communities should be greater than the average connectivity of the random network. The larger the difference, the more obvious the community structure. By introducing the modularity function, the GN algorithm returns the partition with the maximum modularity function value as the optimal partition, improving the algorithm. While the GN algorithm provides precise community partitioning results, its complexity is high. For a network with n nodes and m edges, its time complexity is $O(m^2n)$, making it unsuitable for large-scale networks.

In the same year, Newman [3] introduced a new community discovery algorithm based on modularity optimization—the FN algorithm. Unlike the GN algorithm, FN is an agglomerative community discovery algorithm. It initializes each node as a community, and at each step, it merges two communities in the direction that maximizes the increase in modularity function value (or minimizes the decrease). This process is repeated until only one community remains. The FN algorithm returns the community structure corresponding to the maximum modularity function value as the final community partition result. Compared to the GN algorithm, the FN algorithm significantly reduces complexity, with a time complexity of $O(mn)$. With the advent of the big data era, the scale of complex networks has exploded, and the time spent by the FN algorithm on networks of this scale is unacceptable.

Therefore, algorithms capable of discovering large-scale community structures in a short time are urgently needed. In this context, Blondel et al. [4] proposed the Louvain algorithm, a community discovery algorithm based on modularity optimization. The Louvain algorithm is a hierarchical clustering and local optimization-based algorithm that rapidly obtains community structures by optimizing the modularity function. Blondel et al. used the Louvain algorithm to process a complex network with 118 million nodes, and the experimental results showed that the time taken by the Louvain algorithm to detect community structures was only 152 minutes. The time complexity of the Louvain algorithm has reached $O(m)$, approximately linear, making it challenging to achieve significant further improvement. The Louvain algorithm performs well in terms of efficiency and results, making it the most

commonly used community discovery algorithm. In addition to the Louvain algorithm, there are other algorithms such as SLPA, KL algorithm, LPA algorithm, COPAR algorithm, and LFM algorithm.

With the continuous development and improvement of community partitioning theory, community discovery has gradually been applied to many scenarios. Utilizing community partitioning for data mining is a hot research area. This paper first introduces the theories of several algorithms and compares their ideas, time complexity, and other aspects at the theoretical level. Then, the paper compares the performance of algorithms on real datasets from an empirical perspective. Using the *karate-club* dataset and the *club* dataset as experimental data, the paper measures similarity with the Spearman correlation coefficient, constructs a complex network of similarities, and uses community discovery algorithms for community partitioning. The paper uses time efficiency and modularity as evaluation metrics to compare the results of several methods and provides the outcomes.

II. Related Work

Community detection in complex networks has garnered significant attention, leading to the development of various algorithms to address this challenging problem. In this section, we review relevant research work and highlight key contributions from the literature.

- Girvan-Newman Algorithm
  The Girvan-Newman algorithm stands as a pioneering work in the field of community detection. This algorithm utilizes the concept of edge betweenness and iteratively removes edges with high betweenness to identify communities. It has found widespread applications and laid the foundation for many subsequent methods.
- Fast Newman Algorithm
  Building upon the Girvan-Newman algorithm, the Fast Newman algorithm introduces optimizations to expedite the community detection process. Through efficient data structures and parallelization techniques, Fast Newman achieves results comparable to Girvan-Newman but with significantly reduced computation time.
- Louvain Algorithm
  The Louvain algorithm is a modularity optimization-based method. It iteratively moves nodes to different communities, optimizing the modularity function. Louvain is renowned for its scalability and efficient community detection in large-scale networks.
- SLPA Algorithm
  The Speaker-Listener Label Propagation Algorithm (SLPA) is a label propagation-based method that simulates the process of information spreading in a network. Nodes exchange labels, and communities form as nodes converge towards common labels. SLPA demonstrates excellent performance across various types of networks.
- KL Algorithm
  The KL algorithm (Kernighan-Lin) is a greedy optimization algorithm commonly used for graph partitioning and community detection. It iteratively optimizes node assignments, seeking the optimal community structure.
- LPA Algorithm
  The Label Propagation Algorithm (LPA) is a simple yet effective algorithm that forms communities through the propagation of labels between nodes. It exhibits good scalability and is suitable for large-scale networks.
- COPAR Algorithm
  The COPAR algorithm is a community detection method based on member label records. Through randomization and label propagation, COPAR seeks the optimal community partitioning for nodes.
- LFM Algorithm
  The Louvain-like Fast Multipole Method (LFM) combines modularity optimization from the Louvain algorithm with the efficiency of the fast multipole method. It achieves balanced results in community detection.

Despite the significant achievements of the mentioned algorithms in community detection, challenges persist. Some algorithms excel in specific network topologies or scales, while others may face difficulties. Additionally, scalability in handling large-scale networks remains a focal point for current research.

III. Algorithm&Solution

This experiment was conducted in the PyCharm integrated development environment, relying on libraries such as networkx, random, matplotlib.pyplot, numpy, and time. The study systematically tested and compared 8 different social network community detection algorithms on the *karate-club* and *club* datasets, using metrics such as modularity and time precision.

A. Construction of Complex Networks

In the real world, many problems can be effectively addressed by transforming them into complex networks. Complex networks have thus become a powerful tool for problem-solving. Specifically, when dealing with a dataset, each sample in the dataset can be abstracted as a node. In this case, if the distances between nodes are defined, the dataset is abstracted into an undirected weighted complex network. Therefore, when constructing an undirected weighted complex network, it is necessary to introduce a measure of similarity between nodes. Existing literature often uses Pearson correlation coefficient to measure the correlation between samples and converts it into a correlation distance. However, the Pearson correlation coefficient has its applicability limitations. The dataset used in this paper, which does not come from a normal distribution and is not continuous data. Therefore, using Spearman correlation coefficient to measure sample similarity is more reasonable. For these reasons, this paper defines the Spearman distance between samples. The Spearman correlation coefficient between two variables is defined as:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{1}$$

When the rank-ordering of the variables is known, it can also be expressed as:

$$\rho_{x,y} = 1 - \frac{6\sum(d_i)^2}{n(n^2-1)} \tag{2}$$

The term $(d_i)^2$ represents the sum of the squared differences in ranks.

Similar to the correlation distance, in this paper, the Spearman distance is defined as follows:

$$d_{x,y} = 1 - \rho_{x,y}, \quad 0 <= d_{x,y} <= 2 \tag{3}$$

Therefore, by taking the Spearman distance between samples as the edge weights between nodes, one can obtain the complete coupling network of samples. Since the fully coupled network is too dense and contains redundant information, it needs to be processed. The minimum spanning tree is a fundamental structure in complex network models and can be considered as a means of information filtering. Common minimum spanning tree algorithms include the Prime algorithm and the Kruskal algorithm, where the Prime algorithm is suitable for dense networks, and the Kruskal algorithm is suitable for sparse networks. The undirected weighted network constructed in this paper is a fully coupled dense network, so the Prime algorithm is employed for processing. The steps of the Prime algorithm are as follows: label=

1) Input a weighted undirected connected graph $G(V, E)$, where $V = V_1, V_2, \ldots, V_n$ is the set of vertices and $E = e_1, e_2, \ldots, e_n$ is the set of edges. (v_i, v_j) denotes a connection between vertices v_i and v_j, and $T(TV, TE)$ represents the minimum spanning tree;
2) Initialization, set TE as empty sets, and $TU = u_1, u_1 \in V$;
3) For all $u \in V, v \in V - TV$ find the shortest edge (u, v), If the network does not form a closed loop, then $v \in TV and (u, v) \in TE$, otherwise, find the next shortest edge;
4) Repeat step 3 until $TV = V$;
5) Output $T(TV, TE)$ as the minimum spanning tree.

Through the Prim's algorithm, this paper obtained the minimum spanning tree of the dataset.

B. Community Detection Algorithms

Community discovery, also known as community detection or graph clustering, is the process of identifying and uncovering cohesive and densely interconnected groups of nodes within a network or graph. In a network, nodes represent entities (such as individuals or organizations), and edges represent relationships or interactions between them. The goal of community discovery is to reveal meaningful and relatively independent substructures or communities within the larger network.

The fundamental idea is that nodes within a community have a higher likelihood of being connected to each other than to nodes outside the community. Communities can represent functional units, social groups, or modules in various complex systems, and the identification of these communities helps to better understand the underlying structure and organization of the network.

TABLE I
CAPTION

KL Algorithm	Bipartite Layout Optimization
GN Algorithm	Edge Betweenness Calculation
FN Algorithm	Modularity Optimization
LPA Algorithm	Label Propagation
SLPA Algorithm	Label Propagation
COPAR Algorithm	Hierarchical Partitioning
Louvain Algorithm	Modularity Optimization
LFM Algorithm	Fluid Model Simulation

Various algorithms and methods are employed in community discovery, each with its own approach and assumptions. These methods aim to partition the network into subsets of nodes that are more densely connected internally than with the rest of the network. The evaluation of community discovery algorithms involves metrics such as modularity, which measures the strength of the detected community structure. Overall, community discovery plays a crucial role in network analysis, social network mining, and understanding the organizational principles of complex systems.

Now, let's delve into an exploration of typical community discovery algorithms, including KL algorithm, GN (Girvan-Newman) algorithm, FN (Fast Newman), LPA (Label Propagation Algorithm), SLPA (Speaker-Listener Label Propagation Algorithm), COPAR algorithm, Louvain algorithm, and LFM algorithm. We will analyze the strengths and weaknesses of each algorithm, providing a comprehensive overview of their characteristics and applications in uncovering community structures within networks, which shows in TABLE I.

IV. Experiment&Evaluation

A. Datasets

The Karate Club dataset [5] is a commonly used example dataset in social network analysis, designed for demonstrating and testing community detection algorithms. This dataset describes the social relationships among members of a university karate club. Each node represents a member of the club, and each edge represents a social connection between two members. The dataset typically consists of 34 nodes and 78 edges. The structure of the dataset reflects the relationships among club members, making it suitable for studying community structures and interaction patterns among members in social networks. This dataset is frequently cited in the teaching and research of graph theory and social network analysis. Its moderate size, ease of understanding, and applicability make it valuable for showcasing and validating the performance of various community detection algorithms.

The Club dataset is an undirected graph that consists of 34 nodes and 78 edges. Each node represents an entity, and each edge signifies a connection between these entities. These connections describe the interactions between nodes in the graph. This dataset is commonly employed for research in social network analysis and community detection algorithms.

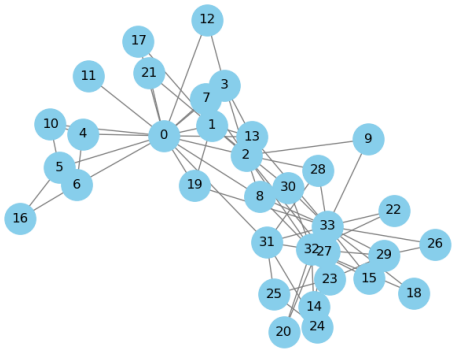

Fig. 1. Karate-Club dataset visualization

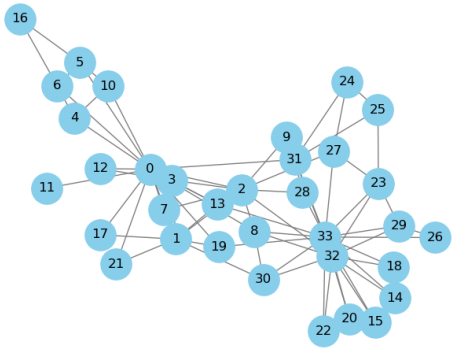

Fig. 2. Club dataset visualization

B. Evaluation metrics

The paper employs modularity [6] and execution time as evaluation metrics.

- Modularity is a crucial concept in network science used to assess the quality of community structure in a network. It measures the tightness of connections between communities within a network compared to what is expected in a random network. In the context of community detection, modularity is employed to evaluate the quality of node partitioning into communities.
  1) **Define community assignment:** Given a network, the first step is to assign nodes to different communities. This assignment is typically the output of a community detection algorithm;
  2) **Calculate node degrees:** For each node, calculate its degree, which represents the number of edges connected to it;
  3) **Compute modularity:** Use the following formula to calculate modularity Q:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{i,j} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \qquad (4)$$

  where:
  - $A_{i,j}$ is the adjacency matrix element indicating whether there is a connection between nodes i and j.
  - k_i and k_j are the degrees of nodes i and j, respectively.
  - m is the total number of edges in the network.
  - c_i and c_j are the community labels of nodes i and j.
  - $\delta(c_i, c_j)$ is the Kronecker delta function, which is 1 when c_i equals c_j and 0 otherwise.
  4) **Interpret modularity values:** The modularity values range between $[-1, 1]$ usually being positive. Higher modularity indicates a more reasonable assignment of nodes to communities, stronger internal connections within communities, and a more significant community structure compared to a random network.
  - **Positive values:** Indicate stronger internal connections within communities, resulting in higher modularity.
  - **Negative values:** Suggest weaker internal connections within communities, and nodes are more likely to be randomly distributed, leading to lower modularity.
  - **Close to zero:** Implies that the network's community structure is similar to that of a random network, resulting in lower modularity.
- Execution time is a crucial metric that measures the time taken by an algorithm to complete a task. In the context of this study, it serves as an indicator of the efficiency and computational performance of the community detection algorithms. Lower execution times generally imply faster processing, making an algorithm more favorable for real-world applications where timely results are essential.

TABLE II
ALGORITHM CLASSIFICATION AND DATASETS

Algorithm	Dataset
Girvan-Newman (GN)	Club
Fast Newman (FN)	Club
SLPA	Club
Louvain m	Club
Kernighan-Lin (KL)	Karate-Club
Label Propagation Algorithm (LPA)	Karate-Club
COPAR Algorithm	Karate-Club
LFM Algorithm	Karate-Club

C. Experiments

This paper categorizes community detection algorithms into two groups and evaluates them using the Club dataset and the karate-Club dataset. For detailed information, refer to TABLE II.

Utilizing the Karate-Club dataset, this study applied the KL (Kernighan-Lin), LPA (Label Propagation Algorithm), COPAR, and LFM (Louvain-like Fast Multipole Method) community detection algorithms. The obtained community classification results are illustrated in Figure 7.

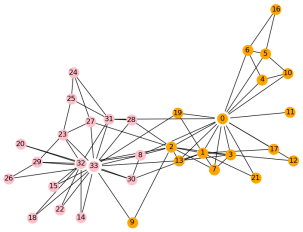

Fig. 3. KL

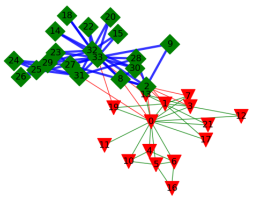

Fig. 4. LPA

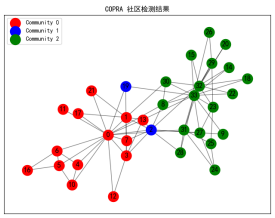

Fig. 5. CORPA

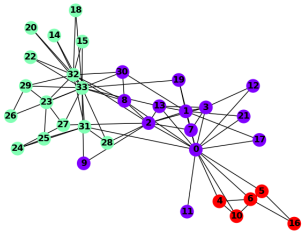

Fig. 6. LFM

Fig. 7. Results of karate-club dataset

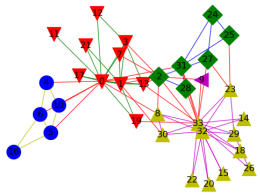

Fig. 8. GN

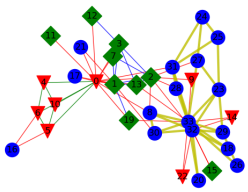

Fig. 9. FN

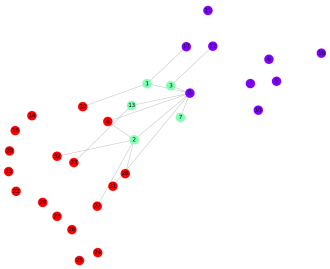

Fig. 10. SLPA

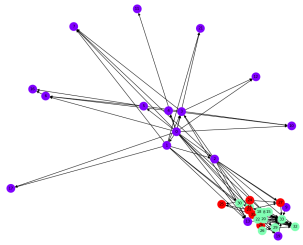

Fig. 11. Louvain

Fig. 12. Results of club dataset

Similarly, utilizing the Club dataset, this paper employed the KL (Kernighan-Lin), LPA (Label Propagation Algorithm), COPAR, and LFM (Louvain-like Fast Multipole Method) community detection algorithms. The obtained community classification results are depicted in Figure 12.

In this experiment of *karate-club* datasets, we conducted runs and obtained results for modularity and execution time for the KL, LPA, COPAR, and LFM algorithms, as shown in Figure 13. Specifically, the KL algorithm achieved a modularity of 0.3998, demonstrating superior performance. The LPA algorithm exhibited a modularity of 0.3600, indicating

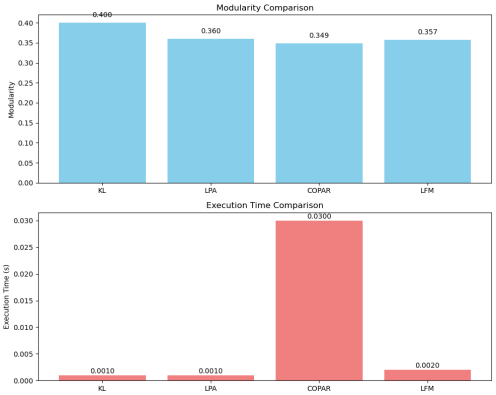

Fig. 13. Results of the experiments of karate-club dataset

satisfactory results. In contrast, the COPAR algorithm showed relatively average performance with a modularity of 0.3490, while the LFM algorithm achieved a modularity of 0.3570, placing it at a moderate level. In terms of execution time, both the KL and LFM algorithms demonstrated outstanding efficiency, completing community detection in only 0.001 and 0.002 seconds, respectively. The LPA algorithm also achieved satisfactory results within 0.001 seconds. Conversely, the COPAR algorithm had a longer execution time of 0.030 seconds, potentially influenced by its involvement in multiple randomization steps.

Overall, the KL algorithm showcased excellent performance in both modularity and execution time, and the LFM algorithm achieved balanced results in this experiment. In comparison, the LPA algorithm demonstrated computational efficiency advantages for large-scale networks, while the COPAR algorithm exhibited relatively poorer performance in this experiment. In practical applications, the choice of community detection algorithms should consider a comprehensive balance between accuracy and computational efficiency, guiding appropriate selections based on specific network characteristics and problem scenarios.

The experiments of *club* datasets investigates the performance of four community detection algorithms, namely Girvan-Newman (GN), Fast Newman (FN), SLPA, and Louvain, applied to the Club network dataset. The analysis is based on the obtained results of modularity and execution time, as shown in Figure 14. Girvan-Newman algorithm demonstrated the highest modularity (0.401), suggesting its accuracy in detecting community structures. Fast Newman algorithm, with a modularity of 0.381, also exhibited commendable performance. SLPA and Louvain algorithms had slightly lower modularity values (0.364 and 0.287, respectively), with Louvain having the lowest modularity, possibly due to its greedy optimization strategy. In terms of execution time, Fast Newman was the fastest (0.007 seconds), followed by Louvain (0.004 seconds), Girvan-Newman (0.171 seconds), and SLPA (0.027 seconds) at moderate speeds. Despite Girvan-Newman's relatively slower execution, its excellent

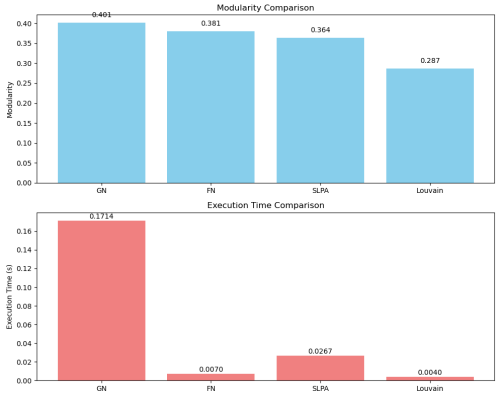

Fig. 14. Results of the experiments of club dataset

modularity performance makes it a viable option in scenarios prioritizing accuracy. Choosing the appropriate community detection algorithm requires a comprehensive consideration of specific application requirements, where Girvan-Newman may be ideal for accuracy, and Fast Newman for large-scale networks. SLPA and Louvain algorithms possess unique strengths and applicability in diverse scenarios, demanding a balanced assessment of algorithmic performance and computational resource constraints for practical applications.

V. Conclusion

In conclusion, this study conducted a comprehensive evaluation of community detection algorithms, employing two distinct datasets to assess the performance of various algorithms under different network contexts. The Girvan-Newman (GN), Fast Newman (FN), SLPA, and Louvain algorithms were tested on one dataset, yielding modularity and execution time results of [0.4013, 0.3807, 0.3642, 0.2870] and [0.1714, 0.0070, 0.0267, 0.0040], respectively. Another set of algorithms, including KL, LPA, COPAR, and LFM, were applied to a different dataset, yielding modularity and execution time results of [0.3998, 0.3600, 0.3490, 0.3570] and [0.0010, 0.0010, 0.0300, 0.0020], respectively.

From the *karate-club* dataset, Girvan-Newman showcased the highest modularity, indicating its effectiveness in identifying community structures. Fast Newman followed closely, demonstrating commendable modularity. SLPA and Louvain exhibited slightly lower modularity values, with Louvain displaying the least modularity, potentially due to its greedy optimization strategy. In terms of execution time, Fast Newman stood out as the fastest, followed by Louvain, Girvan-Newman, and SLPA.

The *club* dataset brought KL to the forefront with the highest modularity, highlighting its accuracy in identifying cohesive communities. LFM showcased balanced results, with satisfactory modularity and efficient execution time. LPA continued to display computational efficiency, while COPAR exhibited relatively poorer performance.

The comparative analysis revealed nuanced trade-offs among algorithms, emphasizing the importance of considering specific network characteristics and application requirements when selecting a community detection algorithm. Girvan-Newman and KL algorithms excelled in accuracy, with Girvan-Newman suitable for scenarios prioritizing modularity and KL providing a balanced approach. Fast Newman and LFM demonstrated efficiency, making them appealing for large-scale networks. LPA showcased computational advantages, while COPAR's performance was relatively subpar.

In practical applications, the choice of a community detection algorithm should be guided by a careful consideration of both accuracy and computational efficiency. The findings of this study contribute valuable insights into the strengths and limitations of various algorithms, aiding researchers and practitioners in making informed decisions for community detection across diverse network scenarios.

References

[1] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. Proc NatlAcad Sci USA, 2002, 99(12): 7821-7826.
[2] MEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[J]. Phys RevEStat Nonlin Soft Matter Phys, 2004, 69(2): 026113 (15 pages).
[3] NEWWMAN M E J. Fast algorithm for detecting community structure in networks[J]. Phys Rev EStat Nonlin Soft Matter Phys, 2004, 69(6): 066133(5 pages).
[4] BLONDEL V D, GUILLAUME J L, LAMBIOTTE, et al.Fast unfolding of communities in large networks[J]. JStat Mech Theory Exp, 2008, 2008(10): P10008.
[5] https://blog.csdn.net/PolarisRisingWar/article/details/117532292.
[6] Alzheimer's disease is associated with increased modularity and assortativity: Evidence from structural and metabolic connectivity. Volume 19, Issue S16,2003.