A Review of Community Detection Algorithms Based on Modularity Optimization

Xiao Zhang¹, Zhixin Ma¹, Ze Zhang¹, Qijuan Sun¹ and Jun Yan¹

¹School of Information Science and Engineering, Lanzhou University, Lanzhou, China
E-mail: 1411688299@qq.com

Abstract. Community structure is considered to be one of the most important features of the real network. Community detection helps to understand the real construction of the network and can better analyze various complex systems. In this paper, five algorithms based on module degree optimization (GN, FN, CNM, Louvain, SML) are introduced. The ideas and design principles of this algorithms are introduced in detail, and the characteristics and advantages and disadvantages of each method are summarized. Finally, the prospect of this kind of algorithm is summarized.

1. Introduction

In recent years, with the rapid development of Internet and computer technology, complex networks, such as social network, biological information network and semantic Web network, have springing up rapidly. The related research of complex networks has also received extensive attention from academic and industrial circles. Community structure in complex networks is considered to be one of the most important features of the real network [1], which is defined as a set of nodes which are closely connected to the other parts of the network and are relatively sparse with the other parts of the network. Community detection has been widely applied in many fields, such as sociology, biology and computer science.

For different types and scale of complex networks, researchers have proposed a lot of community detection algorithms. These algorithms can be roughly divided into modularity optimization algorithm, clique percolation algorithm, label propagation algorithm and hierarchical partitioning algorithm. In this paper, the ideas, design principles, characteristics and advantages and disadvantages of five algorithms based on modularity optimization (GN, FN, CNM, Louvain, SML) in community detection are introduced in detail.

2. Community Detection Algorithms Based on Modularity Optimization

2.1. The GN Algorithm

In 2004, Girvan and Newman put forward the GN algorithm [2] based on the split hierarchical clustering idea. The main idea of this method is to iteratively delete the edges with the largest betweenness from the network, and finally find the optimal partition in the splitting process. The betweenness is defined as the number of shortest paths through all nodes in a network. In this algorithm, the modularity $Q$ is introduced as the measurement standard of the partition quality, defines the $e_{ij}$ as the proportion of the number of edges between the $i$ community and the $j$ community, and the modularity can be expressed as the formula (1)(2).
\[ Q = \sum_i (e_{ii} - a_i^2) = \text{Tre} - \| e^2 \| \] (1)

\[ a_i = \sum_j e_{ij} \] (2)

\[ \| e^2 \| \] represents the sum of all elements in matrix \( e^2 \), and \( \text{Tre} \) is the trace of matrix.

The GN algorithm steps are described as follows:

Step 1. Computing the betweenness of all edges in a network.

Step 2. Removing the edge with the greatest betweenness from the network and calculate the modularity \( Q \) of the network at this time.

Step 3. Repeat steps 1 and 2 until all edges in the network have been removed.

Step 4. Comparing the value of each \( Q \) in the iteration, taking the partition that makes \( Q \) the largest, as the final network structure.

The GN algorithm is simple and easy to implement. The algorithm can be well divided on the Zachary’s karate club network, Collaboration network and other networks. However, there may be a lot of the shortest path in the calculation of the boundary number, which leads to the high time complexity of the algorithm.

2.2. Community Fast Detection Algorithm

In a network with \( n \) nodes and \( m \) edges, the time complexity of the GN algorithm is \( O(m^2n) \); therefore, the network size that can be detected is restricted to a network with a maximum of thousands of points. However, the large-scale network in the actual study occupies a high proportion, for example, the citation and collaboration networks contains millions of nodes [3], and the World-Wide-Web contains billions of [4] nodes. GN algorithm has not met the requirement. In order to efficiently divide the community structure of large-scale network, Newman put forward the FN algorithm [5] in 2003.

In the FN algorithm, we use modularity \( Q \) [2] to measure the quality of each partition. \( e_{ij} \) is defined as the ratio of the number of edges between the \( i \) community and the \( j \) community to the number of the entire network edge. The modularity \( Q \) of a network can be expressed by formula (3).

\[ Q = \sum_i (e_{ii} - a_i^2) \] (3)

The \( a_i^2 \) means that in the corresponding configuration models with the same degree sequence, the number of connected nodes between these nodes occupies the expectation of the ratio of the number of edges of the entire network.

The FN algorithm is an agglomerative hierarchical clustering method that uses a "greedy" optimization strategy [6]. The main process of the algorithm is described as follows:

Step 1. Initialization. Each point in the network as a separate community.

Step 2. Merge communities. Choose to merge the communities that increase the value of the modularity most (or minimize the value of the modularity) during the consolidation process.

Step 3. Iteratively executes the step 2 until all the communities are merged into a single community. Find the division when the value of the modularity value \( Q \) is maximized during the consolidation process as the final network structure.

Although the FN algorithm has low time complexity, it can only obtain the approximate optimal community structure under normal circumstances.

2.3. CNM Algorithm

In 2004, Clauset et al. proposed the CNM algorithm [7] by improving the optimization method and data structure of the FN algorithm. This method is also based on greedy modularity optimization strategy, but it performs better than FN algorithm on large scale network. For example, in a network with \( n \) nodes and \( m \) edges, the time complexity of algorithm runs to \( O(md\log n) \), where \( d \) represents the depth of the dendrogram of network structure.
In order to improve the efficiency of the algorithm in CNM algorithm, the following three data structures are defined. A sparse matrix made up of $\Delta Q$, a large heap composed of the largest elements of every row in $\Delta Q$ matrix; a vector array consisting of $a_i, \Delta Q$ and $a_i$ can be defined as follows formula (4) and formula (5):

$$
\Delta Q_{ij} = \begin{cases} 
1/2m - k_i k_j / (2m)^2 & \\
0 & 
\end{cases}
$$  \hspace{1cm} (4)

$$
a_i = \frac{k_i}{2m}
$$  \hspace{1cm} (5)

In formula (4), $m$ represents the total number of sides in the network. If the two communities have no edges connected, the value of $\Delta Q$ is 0. In formula (5), $k_i$ represents the degree of the $i$ community.

The CNM algorithm steps are described as follows:

Step 1. According to formula (4) and (5), calculating the initial $\Delta Q$ and $a_i$ in the network, and construct the large root heap $H$ by the largest element in every row of the matrix.

Step 2. Select the largest $\Delta Q$ from the large root heap $H$, merge its corresponding community, and update $\Delta Q$, $H$ and $a_i$.

Step 3. Repeat step 2 until all nodes in the network are merged into one community.

2.4. Lovain Algorithm

In 2008, Vincent D et al. proposed the Lovain algorithm based on modularity optimization [8]. The main goal of the algorithm is to optimize the modularity of the entire network by continuously dividing the community.

The Lovain algorithm is mainly divided into two steps. The first step is the Modularity Optimization process. For an arbitrary node $i$ in the network, it is divided into the communities $C_j$ where its neighboring nodes are located, and then the degree of change of modularity of the community $C_j$ is calculated according to formula (6). If $\Delta Q > 0$, accept this division, otherwise $i$ stays in the original community.

$$
\Delta Q = \left[ \sum_{m} \frac{k_{i,m}}{2m} - \left( \sum_{out} \frac{k_i}{2m} \right)^2 \right] - \left[ \sum_{m} \frac{k_{out}}{2m} - \left( \frac{k_i}{2m} \right)^2 \right]
$$  \hspace{1cm} (6)

$\sum_{m}$ is the sum of the weights of all edges whose ends are in community $C$. $\sum_{out}$ represents the sum of the weights of all the edges of an endpoint in the community $C$. The $k_i$ represents the sum of the weights of the edges that are connected to the node $i$. In the formula, $k_{i,m}$ represents the sum of the edge weights connected to node $i$ after adding it to community $C$, and $m$ represents the sum of all edge weights in the network.

The second step is the community aggregation process. Through this process, the communities generated in the first step are aggregated into one point, i.e., the network is reconstructed according to the community structure generated in the previous step.

The Lovain algorithm steps are described as follows:

Step 1. Initialization. In the network, the point $i$ will be randomly divided into community $C_i$.

Step 2. Each point in the community is attempted to be divided into the communities where its neighboring points are located, and the modularity of the community is calculated, then judging the degree of modularity difference $\Delta Q$. If it is positive, accept the division of this time. Otherwise, discard the division.

Step 3. Repeat the above process until the value of module degree is not increasing.

Step 4. Construct a new network structure diagram. Each point in the diagram represents each community identified in step 3, and continue to perform step 2 and step 3 until the structure of the community is not changed.

The Lovain algorithm execution process can be represented by the following figure 1:
2.5. Smart Local Moving Algorithm

Waltman et al. was inspired by the Louvain algorithm to propose SLM (Smart Local Moving Algorithm) algorithm [9] in 2013. The algorithm is mainly divided into three phases:

Step 1. Modularity Optimization: Iteratively divides each node into the community where the neighboring nodes with the largest degree of modularity change are located, until the degree of modularity is optimal.

Step 2. Repeat the first step in the sub-communities generated in the previous step.

Step 3. Rebuilding the network structure with each small community as a vertex, and iteratively performing step 1 and step 2 until the community structures no longer changes.

The difference between the SML and the Louvain algorithm is that after the algorithm completes the first stage modular optimization process, the modularity optimization process is performed again in each generated small community so that each divided community achieves a local optimum. The algorithm not only performs well on large-scale networks with millions of nodes, but also can clearly distinguish community structures in small networks and medium-sized networks.

3. Conclusion

This paper introduces five algorithms based on modularity optimization from several aspects including the idea, characteristics and time complexity of the algorithm. The basic flow of the algorithm and the scope of application of the algorithm are clarified. Because the five algorithms can not achieve the optimal efficiency and algorithm performance at the same time in large-scale networks, community-based detection algorithms based on modularity still have broad research prospects.

4. References

[1] Ma Yiqi, Liu Yong, Wan Jun, et al. Research on Community Detection Algorithm Based on Attribute Transfer in Complex Networks [C]// China Database Academic Conference. 2014.
[2] Newman M E, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69(2 Pt 2):026113.
[3] Newman M E J. The structure of scientific collaboration networks [J]. Proceedings of the National Academy of Sciences, 2001, 98 (2): 404 - 409.
[4] Kleinberg J, Lawrence S. The structure of the Web [J]. Science, 2001, 294 (5548): 1849 - 1850.
[5] Newman M E J. Fast algorithm for detecting community structure in networks [J]. Physical review E, 2004, 69(6): 066133.
[6] Scott J. Social network analysis [M]. Sage, 2017.
[7] Clauset A, Newman M E, Moore C. Finding community structure in very large networks. [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 70 (6 Pt 2): 066111.
[8] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks [J]. Journal of statistical mechanics: theory and experiment, 2008, 2008(10): P10008.
[9] Ludo Waltman, Nees Eck. A smart local moving algorithm for large-scale modularity-based community detection [J]. The European Physical Journal B - Condensed Matter and Complex Systems, 86(11):1–14, 2013.