A Three-Phase Hierarchical Algorithm for Detecting Community Structure

Qiong Hu, Gongshen Liu

Department of Information Security, Shanghai Jiao Tong University, Shanghai, 20000, China
joan_hu@live.com, lgshen@sjtu.edu.cn

Abstract - Social networks, such as Facebook, Twitter, Renren, Sina Weibo, are rising rapidly in the past dozen years and are taking a more and more important role in our daily life. There's a common feature for social networks: they can be divided into subgroups, within which the network connections are dense, but between which they are sparse. The subgroup is known as community, also named group, cluster or module. Community structure detecting can help discover the potential of social networks.

This paper will propose a new algorithm based on the fast GN algorithm in pursuit of an improvement in quality. The algorithm consists of three phases. In the first phase, network will be divided based on the prior knowledge of the degree, during which time, the node with the highest degree in the network will group its neighbor according to a threshold; in the second phase, agglomeration will be done just like the fast GN algorithm; in the third phase, we try to achieve a better community structure by individual vertex moving. In the experiment, our algorithm will be compared with the fast improved GN algorithms and results show that our algorithm has a better in terms of the modularity and NMI, which are the index to measure the quality of community structure, and are proposed respectively by Newman and Lancichinetti.

Index Terms - social network, community structure, modularity, hierarchical algorithm, cluster.

1. Introduction

A community is formally defined as a sub-structure present into the network that represents connections among users, in which the density of relationships within the members of the community is much greater than the density of connections among communities.[1] It reveals a fact in reality: users belonging to the same community could either share tastes or interests in similar products, or be connected by a specific relationship in the real-world, just as the saying goes, birds of a feather flock together. The research on the detection of community structure can largely help commercial organizations to locate the target to put advertising, as well as other propagations.

In view of the fever of the social network, as well as the benefits above, the algorithm for discovering community structure is given more and more emphasis. The existing researches on community structure detection algorithms generally fall into two categories: graph-partitioning algorithms and hierarchical algorithms.

Graph-partitioning algorithms divide the social network into the number specified in advance. It usually partitions the network according to the objective function, such as k-means. Bisection spectral clustering algorithm is one of the representative graph-partitioning algorithms. However, social network division will be certainly beyond two groups. The derivative spectral clustering algorithms can be categorized into two major sets [2]: (1) Recursive Spectral, in which data are partitioned into two sets according to a single eigenvector and by recursion to generate the specified number of clusters, such as SM and KVV; (2) Multi-way Spectral, in which the information hidden in multiple eigenvectors was used in order to directly partition data into the specified number of clusters, such as NJW.

Hierarchical algorithms aim to construct a tree of which the leaves are the individual vertex in the network and the root represents the whole dataset. Hierarchical algorithm can be further divided into two classes: (1) Agglomerative algorithms, which start from the situation that every vertex is a community and groups the vertex pairs of the highest similarity. Most improved GN algorithms, such as fast GN algorithm, belong to this category. (2) Divisive algorithms, which start with a single community, the network as a whole, and attempt to find the least similar connected pairs of vertices, and then remove the edges between them. The GN algorithm is an acquainted divisive algorithm.

In this paper, Section 2 introduces the famous existing hierarchical algorithm, and then explains our design in detail, aiming to improve the existing algorithm. Section 3 will test our algorithms in terms of the modularity and NMI, which are the index to measure the quality of an algorithm. And finally section 4 makes a conclusion and looks forward to the future research.

2. Algorithms

A. New design based on fast GN algorithm

Just as what mentioned in the previous chapter, GN algorithm is a kind of divisive algorithm. The guidance principal of the divisive algorithm is to find the least similar connected pairs of vertices and then to remove the edges between them. While instead of looking for the most weakly connected vertex pairs, the GN-algorithm will look for the edges in the network that are most "between" other vertices. [3]

GN algorithm has two shortcomings:

- It makes high computational demands. It runs in time $O(n^3)$.
- It doesn't converge. That's to say, the algorithm lacks a termination condition.

In order to offset these defects, many improved GN algorithms have been put forward. These algorithms have two

common characteristics: most of them belong to agglomerative algorithm and emphasize on the efficiency improvement. Fast GN algorithm [4] is a typical algorithm among them.

At start of the fast GN algorithm, each node is viewed as an individual community. And then the two communities are merged in condition that they donate the largest Q increase. The algorithm ends when Q cannot be increased.

The fast GN algorithm firstly handles the convergence problem of the GN algorithm, for it offers the termination condition: Q cannot be increased. Moreover, it runs in worst-case time O(nlogn) on a sparse graph, greatly advanced the efficiency.

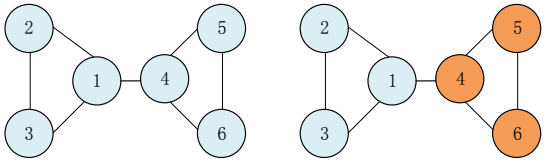
The efficiency improvement of the fast GN algorithm is at the cost of a little deterioration of the quality. Experiments show that fast GN algorithm achieve a little backward modularity compared to the original GN algorithm, the reason of which lies in that fast GN algorithm only resort to "community join", and however, the community joining is unlikely to find joinable communities in a nearly optimal partition.[6]To overcome this, we need to introduce "vertex moving". So, we propose an additional phase after the agglomeration phase, calling "refinement phase". It is employed to do vertex moving, so as to further improve the community structure. Besides, we also take account into the efficiency by creating division phase before agglomeration phase so as to abridge the agglomerative times.

In a word, the algorithm takes three phases, namely division phase, agglomeration phase and refinement phase. We will introduce them respectively in the following subchapters.

B. Division phase

As we can see, in social networks, there are huge amounts of nodes. Consequently, agglomerating from every single node is a very time-consuming operation. The division phase is designed to advance the velocity of agglomeration.

The division phase is first put forward by Haifeng DU et al.[7] In this phase, network will be divided according to the degrees of the nodes. The neighbors of the vertex with highest degree will be assigned together. At the end of the phase, an initial state is formed, with each vertex belongs to a community, either itself, or the community of a vertex with high degree. We will carry on agglomeration phase from this initial state. Apparently, the agglomerative times will be largely cut off. The further improved efficiency brings another problem in quality. The failing in quality takes form of the fact that it usually partitions the network into the clusters fewer than expected. For example, it will consider the network in Figure 1.a as a whole cluster. This problem can be solved by adding a connectivity threshold.



a. Partition by GN-PK algorithm

b. Partition by our algorithm

Fig.1 The threshold of the connectivity helps to find more communities

We define the connectivity between Node u and Community c as:

$$C(u,c) = \sum_{v \in c} a_{u,v} \Big/ k_u \tag{1}$$

Where k_u is the degree of node u, $a_{u,v}$ is the element of the adjacent matrix. The aim is to get how tightly the node is linked with the community.

In the divisive phase, a node is assigned only when C(u,c) is larger than the threshold value. Now problem come to that how to define the threshold value.

The threshold should be concerned individually, for nodes of different degree or community of different size will all influence the final partition. So the threshold cannot have a universal form, we denote it as C_u , where u represents a node in the network. We expect to enable the nodes with lower degree to be assigned to its neighbor, while the nodes with higher degree to remain unassigned. So the threshold should be inverse proportional to its degree, and then how about the numerator? We hope that the node to be assigned should have a close relationship with the nodes already in the community, as well as that most of its connection lead to the node in the community. So we give the definition of Cu as follows:

$$C_u = 0.5 * \min(|C|, k_u) / k_u \tag{2}$$

Where $|C|$ stands for the number of nodes in community C.

We take the network in Figure 1 as an example. From TABLE I, we can see that node 2 and 3 will be put into the same community of node 1, while node 4,5,6 will not as there connectivity is lower than threshold. Then we choose the node with highest degree in the remaining nodes, in this case, it is 4, and with the same method, node 5 and node 6 will be allocated to the community of 4. Thus, with the threshold, the cluster {1,2,3} and {4,5,6} will be successfully detected. The result is depicted in Figure 1.b

TABLE I Partition by Connectivity and Threshold

Node#	C(u,c)	Cu	Assigned to community 1?
2	1/2	1/4	Yes, for 1/2>1/4
3	1	1/2	Yes, for 1 > 1/2
4	1/3	1/2	No, for 1/3 < 1/2
5	0	1/2	No
6	0	1/2	No

We summarize steps in the division phase.

1): Calculate the degree for each node in the network;

2) : Find the node m with maximum degree;

3): calculate the threshold C_m

4): Calculate the C(u,c) for all its neighbouring nodes, if the value is higher than C_m , it should be classified to the community of node m;

5): Remove all connections of the node with maximum degree;

6): If there are no connections in the network, then an initial community structure is obtained, else go to step 1

C. Agglomeration phase

The agglomeration has no evident difference from that of fast GN algorithm.

1): Calculate the modularity according to the modularity function, which is expressed in the equation (3)

$$Q = \sum_{p=1}^{m} [e_{pp} - (\sum_{q=1}^{m} e_{pq})^2] \tag{3}$$

Where, $$e_{pq} = \sum_{i' \in V_p} \sum_{j' \in V_q} a_{i'j'} \Big/ \sum_{i \in V} \sum_{j \in V} a_{ij} \tag{4}$$

Where V represents the set of nodes of the whole network, V_p is the set of nodes of the community p, and a_{ij} is the element of the adjacent matrix. In fact, Q is the fraction of edges that fall within communities, minus the expected value of the same quantity if edges fall at random without regard for the community structure.

Modularity was designed to measure the strength of division of a network into modules.

2): Calculate the vibration of modularity, denoted as Φ_{pq} in equation 5, when joining each pair of communities and note down the largest Φ_{pq}.

$$\Phi_{pq} = e_{pq} - \sum_{j=1}^{m} e_{qj} \sum_{j=1}^{m} e_{pj} \tag{5}$$

3): If the largest Φ_{pq} is greater than zero, combine the two communities, p and q

4):Repeat from step 1until $\Phi pq \leqslant 0$

D. Refinement phase

Before introduce the refinement phase, it is necessary to distinguish the Community Joining with the Vertex Moving. Community Joining iteratively joins two communities, which computes a preliminary clustering starting from singleton communities; while Vertex Moving iteratively moves individual vertices, instead of moving an entire group of densely interconnected vertices, to different clusters.

The strategy for the fast GN algorithm is the iterated joining of community pair, while the community joining is unlikely to find joinable communities in a nearly optimal partition. That's why in our algorithm, we should introduce refinement phase.

The refinement phase in our algorithm will form a better community structure by vertex moving. The steps are concluded as follows:

1): For each node in the network, we move it to other communities and calculate the new modularity Q'.

2): If Q' is less than Q, we move the node back to its original; if not, Q is updated to Q'.

3): Repeat from Step1

E. Complexity

In the division phase, there are two iterations. The outer loop iterates the unassigned node with a high degree. These nodes will be considered as the center of the community. So the iteration times has a relationship with the number of initial communities. The inner loop iterates all nodes in the network.. So the total time compexity of the division phase is O(c*n), where n isthe total number of the nodes and c is denoted as the total number of initial communities in the network. As c is always smaller than n, so the time complexity is lower than $O(n^2)$

The process of the agglomeration phase incudes two "for" loops, meaning it takes time $O(n^2)$

Finally, the refinement phase firstly iterates all nodes in the network and then iterates through the groups. So similar to division phase, the total time is O(c*n).

To summarize, the time complexity as a whole is O(2*c+n)*n). As regarding large complex social network, when comparing to n, c is a number small enough to be neglected regarding the order of magnitude. Thus, the time complexity of the algorithm can be simply expressed as $O(n^2)$.

3. Applications

A. Data Source

Our discussion focuses primarily on networks with only a single type of vertex and a single type of undirected, unweighted edge, although generalizations to more complicated network types are certainly possible. It will be more convenient to calculate the adjacent matrix A_{ij}, which will normally be 0 or 1, although larger values are possible in networks where multiple edges are allowed.

We choose various kinds of data source, sorting by their scale, including karate (34 nodes, 78 edges) [12], political books about US (105 nodes, 441 edges) [13] and football association (115 nodes, 616 edges) [14].

B. Algorithms chosen

The experiments will be carried out on two algorithms: the fast GN algorithm, and the new algorithm proposed in chapter 2.

C. Evaluation methods

1) Modularity: Modularity (usually indicated by Q) is probably the most popular quantitative measure to evaluate the community structure in a network, and is proposed by Newman. It offers a numerical index for the community structure detecting algorithm, helping to quantitatively determine the quality of an algorithm. Networks with high modularity have dense connections between the vertices within modules but sparse connections between vertices in different modules, so the bigger the value of Q, the stronger is the community structure of the network. In practice, Q values for networks with strong community structure typically fall in the range from about 0.3 to 0.7.

2) NMI: Normalized mutual information[15], abbreviated as NMI, is a concept in the information theory. With the known community structure, NMI is a well index to measure the quality of a community structure detecting algorithm. If the result of the algorithm has a higher similarity with the reality, the NMI for this algorithm will also gain a higher value.

D. Results

1) Karate club: During the course of the study, the club

split into two groups as a result of a dispute within the organization, and the members of one group left to start their own club.

From TABLE II, we get that though the fast GN algorithm achieves a higher modularity, our algorithm carries out a closer result to the reality, for it discover the exact number of groups in karate club, so naturally our algorithm can achieve a better NMI than fast GN algorithm.

TABLE II Modularity and NMI on the Dataset of Karate club

	Standard	Fast GN	Three-phase
Community number	2	3	2
Q	0.3715	0.3807	0.3718
NMI	-	0.9043	0.9186

2) Books about US politics: Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon. The books fall into three classes, which indicate three political attitudes: "liberal", "neutral", or "conservative".

TABLE III shows that our algorithm got the highest modularity as well as the accurate number of groups and higher NMI.

TABLE III Modularity and NMI on the Dataset of Books about US Politics

	Standard	Fast GN	Three-phase
Community number	3	4	3
Q	0.4132	0.5020	0. 5269
NMI	-	0.6535	0.8832

3) US Football Association: During the 2000 season, all of the 115 teams are divided into 12 conferences containing around 8 to 12 teams each. Games are more frequent between members of the same conference than between members of different conferences. Apparently, each conference can be considered as one community of the network.

From TABLE IV, we can see that our algorithm still got the highest modularity. What's more, it overwhelms fast GN algorithm in view of NMI .

TABLE IV The modularity of the three algorithms on the dataset of US football association

	Standard	Fast GN	Three-phase
Community number	12	6	6
Q	0.5436	0. 5508	0. 5658
NMI	-	0.5032	0.5856

4) Conclusion Firstly, in the aspect of modularity, our algorithm even gains a much higher value than the standard partition.

Secondly, our algorithm has a delicate advantage over fast GN algorithm, except the modularity of the karate club data.

Thirdly, it is acknowledged that the network's density will more or less influence the accuracy of the three algorithms.

We use E/N to define the density of a network, where E is the total edges of the network and N is total nodes of the network. The higher the value, the denser the network is.

TABLE V shows that the network with a high density will possibly gain a more ideal Q value.

TABLE V The modularity under the network with different density

	Karate Club	Books about US politics	US Football Association
E/N	2.2941	4.2	5.3565
Q (Fast GN)	0.3807	0.5020	0.5508
Q(Three-Phase)	0.3718	0. 5269	0.5916

4. Conclusion

This paper proposes a new hierarchical algorithm based on the GN algorithm to detect community structure, aiming to offset the disadvantages of the existing algorithms. The result of the experiment confirms the quality of our new algorithm, revealing that it takes an advantage over the existing improved GN algorithm like fast GN algorithm.

In reality, an individual may probably belong to two or more communities, so with regard to the future research, attention can be paid to the overlapping problems.

Acknowledgment

First and foremost, I would like show my deep gratitude to my supervisor Liu Gongshen, which gives me lots of help and guidance in the research. I shall also extend my thanks to all my comrades.

References

[1] Emilio Ferrara. Community structure discovery in Facebook. Int. J. Social Network Mining, Vol. 1, No. 1, 2012

[2] Shuzi Niu, Daling Wang, Shi Feng, Ge Yu. An Improved Spectral Clustering Algorithm for Community Discovery. Ninth International Conference on Hybrid Intelligent Systems,2009

[3] M. E. J. Newman, M. Girvan. Finding and evaluating community structure in networks. Physical review E, 2004

[4] M. E. J. Newman. Fast algorithm for detecting community structure in networks. Physical review E, 2004

[5] M. E. J. Newman. Modularity and community structure in networks. PNAS, vol103, 2006,6

[6] RANDOLF ROTTA, ANDREAS NOACK. Multilevel Local Search Algorithms for Modularity Clustering. Journal of Experimental Algorithmics (JEA), 2011

[7] Haifeng Du. An Algorithm for Detecting Community Structure of Social Networks Based on Prior Knowledge and Modularity. Journal of Xi'an Jiao Tong University, 2007, 6

[8] Gaoxia Wang, Yi Shen, Enjie Luan. A measure of centrality based on modularity matrix. Progress in Natural Science 18, 2008

[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment,2009

[10] Ilaria Bordino, Debora Donato, Aristides Gionis. Mining large networks with subgraph counting. Eighth IEEE International Conference on Data Mining, 2008

[11] Matthew J. Rattigan, Marc Maier, David Jensen. Graph Clustering with Network Structure Indices. Proceedings of the 24th international conference on Machine learning, Pages 783-790

[12] W. W. Zachary. An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452-473 (1977).

[13] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002).

[14] V. Krebs. unpublished. http://www.orgnet.com/.

[15] Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E, 2009, 80(1): 016118.