International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Community Detection in Complex Network

Shikha Vishnoi

M. Tech., Computer Science & Engineering, Galgotias University, Greater Noida, UP, India

Abstract: ***A large number of networks in nature, society and technology are defined by a mesoscopic level of organization, in which groups of nodes form tightly connected units, called communities that are sparsely inter-linked to each other. Identifying this community structure is one of the most important problems in understanding of functions and structures of real world complex systems, which is still a challenging task. Various methods proposed so far are not efficient and accurate for large networks which comprise of millions of nodes because of their high computational cost. In this manuscript we will provide the computational analysis of BGLL algorithm and overlapping community detection algorithm (OCDA) for determining the structure of complex networks. BGLL is a variant of hierarchical agglomerative clustering approach and OCDA is based on the principle of edge betweenness.***

Keywords: Community structure, complex networks, BGLL, OCDA, edge betweenness.

1. Introduction

Representation of various real world systems as graphs can give deep insight to the understanding the structure and functions of complex networks. Identification of Community structure has gained a lot of attention among the researchers, along with the other properties of these networks such as small world effect, scale free, power-law degree distribution.

A network is said to possess community structure if the edge density within the community is sufficiently larger than between the communities. A community within a complex graph refers to a partition of nodes in cohesive sub-graphs. Those partitions may commonly allow to some nodes the possibility of belonging to more than one sub-group. This phenomenon is called overlapping communities.

The key problem is to identify communities within the network. Moreover, Community detection within a complex network is a very rich field of research, and considered as a graph partition problem. Discovering optimal partitions in complex networks is known by an exponential complexity since the number of possible community increases as the number of nodes increases. Additionally, the huge amount of data on complex networks and the large number of nodes make the issue of the efficiency of community detection algorithms a hard task, another major reason of this difficulty is due to overlapping community. Hence, a serious necessity is to find an efficient tool to detect accurate partition.

2. Literature Review

Newman and Girvan's work in 2001 suggested that complex systems posses the property of community structure. Identification and detection of community structure was originated as the extension of the problem of graph partitioning. Graph partitioning problem comprised of the division of graph nodes into predetermined number of partitions such that the number of edges between the groups is as few as possible. The number of edges between the groups is called as cut size. Kernighan Li approach is the most popular method for this problem. Spectral bisection is another approach for this problem, which is based on the Laplacian matrix. Another approach for graph partitioning is based on the min cut max flow method proposed by Ford and Fulkerson. Min cut and max flow approach is used in determining communities in the web networks by Flake et.al. Graph can be partitioned into groups by measuring the conductance which affine to the cut size of the graph.

Conductance is computed as:

$$\varphi(C)=(c(C,g/C))/(\min(k_C,k_(g/C)))$$

Where c(C,g\C) is the cut size of C,and k_(g\C) are the total degrees of C and of the rest of the graph g\C ,respectively.

Graph partitioning approaches are not good for community detection, as these methods require the prior information about the number of subparts into which the graph is to be divided. The traditional methods of community detection were hierarchical clustering. Hierarchical clustering (4) methods may be of two types either agglomerative or divisive.

Newman proposed a method of community detection which was based on the idea of betweenness measure. The betweenness was calculated for each pair of edges and edges with the maximum betweenness removed. This process was repeated until no edge remains. Clauset, Newman and Moore proposed the fast modularity approach for community detection which basically finds the best pair of communities to merge and what will be the criteria to stop merging the communities. In this method three data structures were maintained:

a) A sparse matrix Q containing the modularity increment for every pair of communities having a common edge.
b) A max heap consists of the maximum element of each row of Q.
c) An ordinary vector array a.

The running time of this method is O(mdlog n).The extension of CNM algorithm was proposed by Xu Liu in their seminal paper, where they used an additional parameter head-size with the heap data structure. The head size controlled the randomness in the search path and search strategies. Pons and Latapy used the idea of random walks across the network for

community detection. Instead of using modularity as similarity measure this method used the node similarity measure based on short walks. This method was also an instance of hierarchical aggregation. Walk trap method has the complexity of O(mn)2 which could be reduced to O(n4).

Markov clustering algorithm was implemented as a simulation of flow through the network. The underlying idea of this method is that the random walker will spend more time inside the same community. The Markov clustering algorithm has the complexity O(n3). Karsten and Nitesh V. Chawla proposed a method based on random walks. It assumed that a random walker with predetermined number of steps is likely walk inside a single community. The idea of this method was to compute many random short walks and keep tracks of the visited nodes in a single walk.

3. Community Detection Method

3.1 BGLL Algorithm

This community detection algorithm is mainly known as Louvain Method, named after its co-authors location. This method gained a lot of attention among the researchers because of its computational speed and accuracy of the communities detected. The idea of this method was given by Etienne Lefebvre in his Master's Thesis. This method was first improvised and studied by Blondel et al in their paper —"Fast Unfolding of Communities". They concluded in their study that this method was fast in determining the community structure for large networks. Its efficiency was also tested on a large data set of Belgian Phone network and on adhoc networks. It is a simple, efficient and fast method which has been tested on large networks consisting of millions of nodes. This method is considered as one of the most popular methods for community detection. Louvain method is considered as better method than other methods because it computes high modularity partitions and hierarchies of large networks in quick time. This method is more important because of its independence from the resolution limit problem, which was the major concern for all the other methods.

BGLL algorithm consists of two phases— In the first step each node belongs to its own cluster. The clusters are formed more specifically by finding the modularity change by moving a node into the group of neighborhood node. A node is placed into that group for which the modularity gain is positive and maximum. This process is applied for all nodes until there is no improvement possible in the value of modularity. In this modularity gain is computed by:

$$\Delta Q = \left[\frac{\sum_{\text{in}}+2k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right],$$

Where $\sum_{\text{in}}$ is the sum of weights of the links inside C

$\sum_{\text{tot}}$ is the sum of the weights of the links incident to nodes in C

k_i is the sum of the weights of the links incident to node i

$k_{i,\text{in}}$ is the sum of the weights of the links from node i to node in C

m is the sum of the weights of all the links in the networks.

The second phase comprise of building a new network by considering the communities found in first phase as nodes. The weight of the link between the two nodes is given by the sum of weights of the links in the communities. The two phases combined constitute a pass. After each the number of meta-communities decreases every time. The passes are repeated until no improvement can be achieved and the maximum modularity is achieved. The hierarchy of the structure is determined by the number of passes. The advantage of this method is that the calculation of modularity gain is simple in this method. The algorithm provides high modularity partition and hierarchical structure can be visualized at different resolution. Thus this method eliminates the resolution limit problem of earlier methods(1).

3.2 OCDA Algorithm

The proposed algorithm is an agglomerative approach and it builds on the notion of edge betweenness (3) introduced by Newman and Girvan and also present an improvement of the use of this concept. Overlapping Community Detection Algorithm (OCDA) allows nodes to belong to more than one Community.

Our method builds on the algorithm of Newman and Girvan relied on the concept of betweenness centrality. This method doesn't require providing the number of clusters in advance. However, GN algorithm has two major drawbacks which are: The GN algorithm is appropriate when we want discrete groups, not to the case of overlapping communities, and furthermore it is slow and complex to program, GN needs high computational requirements due the step of recalculating the betweenness. Many variations of GN have been proposed to improve it and reduce the complexity. In our method we need to calculate the betweenness just in the beginning of the algorithm. Actually the importance of edge betweenness lies in detecting edges that present a "bridge" between two communities maintaining shortest paths between any pairs of nodes. Additionally, our method differs from the GN algorithm since it allows nodes to belong to more than one community.

Algorithm: Overlapping Community Detection Algorithm OCDA
Input: Network G(V, E)
Output: set S of communities
(1) Begin
(2) While (V is not empty)
(3) Find initial community core (Coc)
(4) Compute the edge betweenness Bvi
(5) Repeat for each Coc
(6) Community Expanding phase
(7){
(8) If adjacent node (min Bvi)
(9) Add adjacent node

(10) }
(11) Repeat for (V-Coc)
(12) Add adjacent node
(13) Community optimization phase
(14) Return the final graph partition S
End

Initially, we do not know the number or the size of the communities. The main process consists on the following steps:

Step (1): Finding initial community core (Coc)

- Discover the key nodes in the network since those nodes are characterized by their influence to other nodes in the graph.
- To construct the initial partition, we must place each central node in a distinct community. This partition is composed of N communities with N is the number of central node.
- Compute the edge betweenness for each edge in the initial graph Bvi .

Step (2): Community expanding phase

- Update the initial community
- For each community core, if the adjacent node have a smaller edge betweenness
- Add iteratively the direct neighbor, i.e. adjacent node
- Add iteratively for each node his direct neighbor (2).
- If a node has a null betweenness with all central nodes, we put it in the community with which it has a maximum node in common.

Step (3): Community optimization phase

Two communities are merged if they are highly overlapped community, i.e. they share several nodes.

Two communities C1 and C2 are merged into C3 if they are highly overlapped community, i.e. they share several nodes.
C3 = C1 U C2

Table 1: OCDA Algorithm Complexity

Algorithm's phase	*Time Complexity*
Finding initial community core	O(n)
Community expanding phase	O(n)
Computing the edge betweenness	O(mn)
OCDA	O(mn)

The previous table presents the time complexity of the three main phase, so we can conclude the complexity of OCDA algorithm where :

- m: the number of edges
- n: the number of nodes

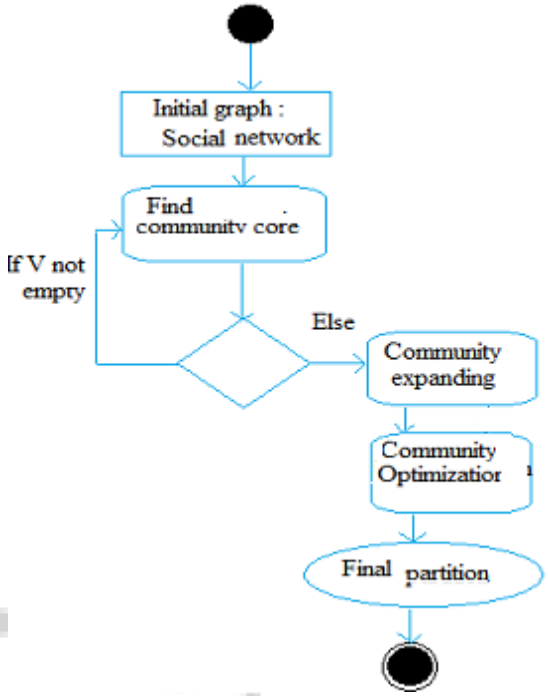

Figure 1: Graphic diagram of the community detection algorithm

This figure illustrates a schematic view of the proposed algorithm. After detecting the community cores, the phase of community expanding will be executed. The final phase lies in community optimization.
According to the author in (2), the degree central nodes are characterized by the following properties:

Definition 1
A node v is called central or also the core of a graph G if and only if the degree of the node v is greater than or equal to all its adjacent nodes' degree. Indeed, two nodes v1 and v2 are considered as central nodes if: deg(v1) ≠ deg (v2), in addition, v1 and v2 must be not adjacent. deg(vi) presents the degree of node vi

Definition 2
In (4) the betweenness of an edge measures the total of all the shortest paths between pairs of nodes running through it.
Bv3(v1,v2) : number of shortest paths from v1 to v2 that pass through v3

Definition 3
Intuitively, if the edge which connects two nodes has a lower betweenness, v1 and v2 are in the same community.

4. Conclusions

Studying and exploring structures in networks represent a helpful task to understand complicated interactions in the real world networks. A great effort has been devoted to find accurate community detection methods in large networks mainly social networks.

In BGLL algorithm modularity is always computed from the initial graph topology, operating on super graphs enables one to consider the variations of modularity for partitions of the

original graph after merging and/or splitting of groups of vertices. Therefore, at some iteration, modularity cannot increase any more, and the algorithm stops. The technique is more limited by storage demands than by computational time. The latter grows like O(m), so the algorithm is extremely fast and graphs with up to 109 edges can be analyzed in a reasonable time on current computational resources. The modularity maxima found by the method are better than those found with the greedy techniques by Clauset et al. and Wakita and Tsurumi.

One thing to be note about our method is that the outcome depends on the order in which the nodes are considered. Although the order doesn't affect the modularity obtained in each pass, but it may affect the computation time. So in order to enhance the computation time, some threshold of the modularity is to be considered as stopping criteria.

In this work, we came to know a new algorithm that seems to respond to the objective of identifying overlapping communities especially in social networks. This method works by first identifying all possible central nodes, then trying to expand communities and optimize this partition in the last phase. Future works include mainly improving the actual algorithm, handle with to several networks with larger size. In addition to that, any proposed method must meet the following requirements, firstly, several nodes may be belong to more than one community with various degree of attachment, and also take into consideration the problem of dynamic communities.

References

[1] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. P10008 (2008).
[2] Q. Chen, T.Wu, M. Fang. Detecting local community structures in complex networks based on local degree central nodes. Statistical Mechanics and its Applications. 2013
[3] M.Girvan, M. E. J.Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, volume 99, pp. 7821-7826, 2002.
[4] A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review . ACM Computing Surveys," Vol. 31, pp. 264-323, 1999.
[5] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167-256.University Press, Cambridge, UK, 2008.
[6] G.W. Flake, S. Lawrence, C. Lee Giles, F.M. Coetzee, Self-organization and identification of web communities, IEEE Computer 35 (2002) 6671.
[7] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[8] M. Latapy, P. Pons, Lect. Notes Comput. Sci. 3733 (2005) 284293.
[9] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[10] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (1970) 291307.
[11] M.E.J. Newman, A measure of betweenness centrality based on random walks, Soc. Netw. 27 (2005) 3954.
[12] Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.
[13] K. Wakita, T. Tsurumi, Finding community structure in mega-scale social networks, eprint arXiv:cs/0702048.
[14] Identifying and evaluating community structure in complex networks Karsten Steinhaeuser, Nitesh V. Chawla *University of Notre Dame, Department of Computer Science and Engineering, Interdisciplinary Center for Network Science and Applications (iCeNSA), Notre Dame, IN 46556, USAvan Dongen, S., 2000. Graph Clustering by Flow Simulation, Ph.D. Thesis, University of Utrecht, Netherlands.

Author Profile

Shikha Vishnoi received the B.Tech. degree in Information Technology from Ideal Institute of Technology, Ghaziabad(U.P) and I am pursuing M.Tech. degree in Computer Science and Engineering from Galgotias University respectively.